



Guide de l'utilisateur

AWS IoT SiteWise



AWS IoT SiteWise: Guide de l'utilisateur

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

Les marques commerciales et la présentation commerciale d'Amazon ne peuvent pas être utilisées en relation avec un produit ou un service extérieur à Amazon, d'une manière susceptible d'entraîner une confusion chez 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 SiteWise ?	1
Comment AWS IoT SiteWise fonctionne	2
Ingérez des données industrielles	2
Modélisez les actifs pour contextualiser les données collectées	3
Analysez à l'aide de requêtes, d'alarmes et de prédictions	4
Visualisez les opérations	4
Stockez les données	5
Intégrer à d'autres services	5
Cas d'utilisation pour AWS IoT SiteWise	5
Fabrication	6
Alimentation et boissons	6
Énergie et services publics	6
Travailler avec AWS SDKs	6
Concepts	7
Mise en route	15
Prérequis	15
Configurez un AWS compte	16
Inscrivez-vous pour un Compte AWS	16
Création d'un utilisateur doté d'un accès administratif	16
Utiliser la démo de démarrage rapide	19
Créez la AWS IoT SiteWise démo	19
Supprimer la AWS IoT SiteWise démo	21
Didacticiels	23
Calculer l'OEE	23
Prérequis	23
Calcul de l'OEE	24
Ingérer des données	26
Prérequis	28
Étape 1 : créer une AWS IoT politique	28
Étape 2 : créer un AWS IoT objet	31
Étape 3 : Création d'un modèle de ressource d'appareil	33
Étape 4 : Création d'un modèle de parc d'appareils	34
Étape 5 : Création et configuration d'un actif d'appareil	36
Étape 6 : Création et configuration d'un parc d'appareils	37

Étape 7 : créer une règle dans AWS IoT Core pour envoyer des données aux actifs de l'appareil	38
Étape 8 : Exécuter le script client de l'appareil	41
Étape 9 : Nettoyer les ressources après le didacticiel	49
Visualisez et partagez des données dans SiteWise Monitor	50
Prérequis	51
Étape 1 : créer un portail dans SiteWise Monitor	52
Étape 2 : Connectez-vous à un portail	56
Étape 3 : Création d'un projet de parc éolien	58
Étape 4 : Création d'un tableau de bord pour visualiser les données du parc éolien	62
Étape 5 : Explorez le portail	69
Étape 6 : Nettoyer les ressources après le didacticiel	70
Publier sur Amazon DynamoDB	73
Prérequis	73
Étape 1 : Configuration AWS IoT SiteWise pour publier les mises à jour de la valeur des propriétés	74
Étape 2 : créer une règle dans AWS IoT Core	76
Étape 3 : Création d'une table DynamoDB	79
Étape 4 : Configuration de l'action de la règle DynamoDB	81
Étape 5 : explorer les données dans DynamoDB	82
Étape 6 : Nettoyer les ressources après le didacticiel	83
Ingérez des données dans AWS IoT SiteWise	87
Gestion des flux de données	88
Configuration des autorisations et des paramètres	89
Associer un flux de données à une propriété d'actif	90
Dissocier un flux de données d'une propriété d'actif	91
Supprimer un flux de données	92
Mettre à jour un alias de propriété d'actif	93
Scénarios courants	94
Ingérez des données avec AWS IoT SiteWise APIs	97
BatchPutAssetPropertyValue API	97
CreateBulkImportJob API	100
AWS IoT Core Règles d'utilisation	108
Accorder l'accès requis	109
Configurer l'action de la règle	110
Réduisez les coûts avec Basic Ingest	119

AWS IoT Events Actions d'utilisation	120
Utiliser le gestionnaire de AWS IoT Greengrass flux	121
Utiliser les passerelles SiteWise Edge	123
Concepts clés du portail	123
Avantages de la mise en œuvre d' SiteWise Edge	124
Hébergez vous-même une passerelle	125
Prérequis	126
Création d'une passerelle	131
Installation du logiciel de passerelle	134
Passerelles V3 compatibles MQTT	137
Streams classiques, passerelles V2	195
Ajouter des sources de données	210
Composants pour SiteWise Edge	254
Filtrer les actifs	256
Support par proxy et magasins de confiance	257
Utiliser APIs	264
Hébergez une passerelle sur Siemens Industrial Edge	282
Sécurité	283
Siemens Secure Storage et l'application AWS IoT SiteWise Edge	284
Destinations pour les Siemens Industrial Edge appareils	284
Migrer depuis l'application de prévisualisation	286
Résolution des problèmes	287
AWS IoT SiteWise Journal des modifications de l'application Edge	287
Prérequis	288
Création d'une passerelle	289
Créer une Siemens Databus user	290
Accédez à l'application	292
Installez l'application	292
Mettre à jour la configuration d'une application installée	295
Destinations et filtres de parcours	295
Comprendre les destinations	296
Comprendre les filtres de trajectoire	300
Ajouter une destination en temps réel	303
Ajouter une destination mise en mémoire tampon à l'aide d'Amazon S3	312
Ajouter des filtres de chemin	320
Gérez les destinations	323

Gérez les passerelles	330
Gérez votre passerelle SiteWise Edge à l'aide de la AWS IoT SiteWise console	331
Gérez les passerelles SiteWise Edge à l'aide de AWS OpsHubAWS IoT SiteWise	331
Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification du système d'exploitation local	333
Gérer le certificat de passerelle SiteWise Edge	336
Modifier la version des packs de composants de la passerelle SiteWise Edge	337
Passerelles List SiteWise Edge	337
Décrire une passerelle SiteWise Edge	338
Création d'une passerelle SiteWise Edge	339
Mettre à jour une passerelle SiteWise Edge	340
Mettre à jour la configuration des fonctionnalités de passerelle	341
Tag : ressources de passerelle	342
Répertorier les balises d'une passerelle	343
Supprimer les tags d'une passerelle	344
Mettre à jour la version d'un AWS IoT SiteWise composant	345
Supprimer une passerelle SiteWise Edge	345
Passerelles de sauvegarde et de restauration	347
Sauvegardes quotidiennes des données métriques	347
Restaurer une passerelle SiteWise Edge	348
Restaurer AWS IoT SiteWise les données	349
Validez les sauvegardes et restaurations réussies	350
Passerelles traditionnelles (AWS IoT Greengrass Version 1)	351
Modéliser des actifs industriels	353
État des ressources et des modèles	355
Vérifier l'état d'un actif	356
Vérifier l'état d'un modèle d'actif ou de composant	357
Versions du modèle d'actifs	359
Récupérer la version active d'un modèle d'actif ou d'un modèle de composant (console)	361
Récupérez la version active d'un modèle d'actif ou d'un modèle de composant (AWS CLI) ..	361
Modèles composites personnalisés (composants)	362
Modèles composites personnalisés en ligne	363
Component-model-based modèles composites personnalisés	365
Utiliser des chemins pour référencer les propriétés personnalisées du modèle composite ...	366
Configurer l'objet IDs	368
Travailler avec un objet UUIDs	369

Utiliser un usage externe IDs	369
Création de modèles	371
Créez des modèles d'actifs dans AWS IoT SiteWise	372
Création de modèles de composants	388
Définir les propriétés des données	392
Création de modèles composites personnalisés (composants)	477
Créez des actifs	481
Création d'un actif (console)	481
Création d'un actif (AWS CLI)	482
Configuration d'un nouvel actif	484
Rechercher des actifs	484
Prérequis	484
Recherche avancée sur Console AWS IoT SiteWise	485
Mettre à jour les valeurs des attributs	488
Associer et dissocier des actifs	491
Associer et dissocier des actifs (console)	491
Associer et dissocier des actifs (AWS CLI)	493
Mettre à jour les actifs et les modèles	494
Mettre à jour les actifs dans AWS IoT SiteWise	495
Mettre à jour les modèles d'actifs et les modèles de composants	496
Mettre à jour des modèles composites personnalisés (composants)	502
Verrouillage optimiste pour les écrits sur le modèle d'actifs	506
Supprimer des actifs et des modèles dans AWS IoT SiteWise	510
Supprimer des actifs	510
Supprimer des modèles d'actifs	513
Opérations groupées avec actifs et modèles	515
Concepts clés et terminologie	515
Fonctionnalités prises en charge	516
Prérequis pour les opérations en masse	517
Exécuter une tâche d'importation groupée	520
Exécuter une tâche d'exportation groupée	522
Suivi de l'avancement des tâches et gestion des erreurs	525
Exemples d'importation de métadonnées	531
Exemples de métadonnées d'exportation	547
Schéma de tâche de transfert de métadonnées	549
Surveillez les données avec des alarmes	568

Types d'alarmes	568
États d'alarme	570
Propriétés de l'état de l'alarme	571
Définissez des alarmes sur les modèles d'actifs	574
Exigences relatives aux notifications d'alarme	577
Définir les AWS IoT Events alarmes	578
Définissez les alarmes externes	614
Configuration des alarmes sur les actifs	616
Configuration d'une valeur de seuil (console)	616
Configurer une valeur de seuil (AWS CLI)	617
Configuration des paramètres de notification	619
Répondre aux alarmes	621
Répondre à une alarme (console)	622
Répondre à une alarme (API)	626
Ingérer un état d'alarme externe	626
Cartographier les flux d'état d'alarme externes	627
Ingérer les données d'état des alarmes	629
AWS IoT SiteWise Assistante	631
Configuration de l' AWS IoT SiteWise assistant	631
Créer un jeu de données	633
Modifier un jeu de données	638
Supprimer un jeu de données	640
AWS IoT SiteWise Questions de l'assistant	641
Surveillez les données avec AWS IoT SiteWise Monitor	642
SiteWise Contrôler les rôles	643
Fédération SAML	645
SiteWise Concepts de surveillance	647
Commencez avec AWS IoT SiteWise Monitor (Classic)	648
Création d'un portail	649
Configurez votre portail	650
Inviter des administrateurs	654
Ajouter des utilisateurs de portail	657
Création de tableaux de bord (CLI)	661
Activez les alarmes pour vos portails	667
Activez votre portail à la périphérie	670
Administrez vos portails	670

Commencez avec AWS IoT SiteWise Monitor (compatible avec l'IA)	680
Création d'un portail	682
Configurez votre portail	682
Administrez vos portails	685
Supprimer un portail	689
Créez des tableaux de bord avec AWS CLI	690
Connexion au portail	695
Création d'un projet	696
Met à jour un projet.	697
Supprime un projet.	697
Création d'un tableau de bord	698
Mise à jour d'un tableau de bord	699
Suppression d'un tableau de bord	700
Configuration du tableau de bord	701
Interrogez les données de AWS IoT SiteWise	723
Rechercher les valeurs actuelles des actifs	724
Rechercher la valeur actuelle d'une propriété d'actif (console)	724
Rechercher la valeur actuelle d'une propriété d'actif (AWS CLI)	725
Rechercher les valeurs historiques des propriétés des actifs	726
Rechercher des agrégats de propriétés d'actifs	728
Agrégats pour une propriété d'actif (API)	728
Agrégats pour une propriété d'actif (AWS CLI)	730
AWS IoT SiteWise langage de requête	731
Prérequis	732
Référence du langage de requête	732
Interagissez avec d'autres services	741
Comprendre les propriétés des actifs dans les rubriques MQTT	742
Travailler avec les notifications	742
Activer les notifications relatives aux propriétés des actifs (console)	743
Activer les notifications relatives aux propriétés des actifs (AWS CLI)	743
Notifications de requêtes	745
Exporter des données vers Amazon S3	748
Intégrez Grafana	749
Intégrez avec AWS IoT TwinMaker	750
Activation de l'intégration	751
Intégration AWS IoT SiteWise et AWS IoT TwinMaker	751

Détecter les anomalies de l'équipement	752
Ajouter une définition de prédiction (console)	754
Entraînez une prédiction (console)	757
Démarrer ou arrêter l'inférence sur une prédiction (console)	758
Ajouter une définition de prédiction (CLI)	759
Entraînez une prédiction et démarrez l'inférence (CLI)	762
Entraînez une prédiction (CLI)	764
Démarrer ou arrêter l'inférence sur une prédiction (CLI)	766
Gérer le stockage des données	769
Configuration des paramètres de stockage	770
Impact sur la conservation des données	771
Configuration pour le niveau chaud (console)	771
Configurer pour Warm Tier (AWS CLI) (.....	773
Configuration pour le niveau froid (console)	776
Configurer pour le niveau froid (AWS CLI)	779
Résoudre les problèmes liés aux paramètres de stockage	784
Erreur : le compartiment n'existe pas	785
Erreur : accès refusé au chemin Amazon S3	785
Erreur : l'ARN du rôle ne peut pas être assumé	786
Erreur : Impossible d'accéder au compartiment Amazon S3 interrégional	786
Chemins de fichiers et schémas de données enregistrés dans le niveau froid	786
Données relatives à l'équipement (mesures)	786
Métriques, transformations et agrégats	791
Métadonnées des actifs	797
Métadonnées de hiérarchie des actifs	801
Fichiers d'index des données de stockage	803
Exemples de code	804
Principes de base	808
Bonjour AWS IoT SiteWise	809
Principes de base	813
Actions	876
Sécurité	953
Protection des données	954
Confidentialité du trafic inter-réseau	955
AWS IoT SiteWise Amélioration du service commercial adjoint	955
Chiffrement des données	955

Chiffrement au repos	956
Chiffrement en transit	959
Gestion des clés	961
Gestion des identités et des accès	962
Public ciblé	963
S'authentifier avec des identités	964
Comment AWS IoT SiteWise fonctionne avec IAM	967
Politiques gérées	988
Rôles liés à un service	992
Configurer les autorisations pour les alarmes	1010
Prévention interservices confuse des adjoints dans AWS IoT SiteWise	1016
Résoudre les problèmes d'identité et d'accès	1017
Validation de conformité	1019
Résilience	1021
Sécurité de l'infrastructure	1022
Analyse de la configuration et des vulnérabilités	1022
Points de terminaison d'un VPC	1023
Opérations d'API prises en charge	1024
Création d'un point de terminaison d'un VPC d'interface	1026
Accès AWS IoT SiteWise via un point de terminaison VPC d'interface	1027
Créer une politique de point de terminaison de VPC	1029
Bonnes pratiques de sécurité	1030
Utilisez les informations d'authentification sur vos serveurs OPC UA	1030
Utilisez des modes de communication cryptés pour vos serveurs OPC UA	1030
Maintenez vos composants à jour	1030
Chiffrez le système de fichiers de votre passerelle SiteWise Edge	1031
Accès sécurisé à votre configuration Edge	1031
Sécurisation des données sur Siemens Industrial Edge Management	1031
Accorder aux utilisateurs de SiteWise Monitor les autorisations minimales possibles	1031
Ne pas exposer d'informations sensibles	1032
Suivez les meilleures pratiques en matière de AWS IoT Greengrass sécurité	1032
Consultez aussi	1032
Journalisation et surveillance	1033
Surveiller les journaux de service	1034
Gérer la connexion AWS IoT SiteWise	1035
Exemple : entrées de fichier AWS IoT SiteWise journal	1037

Surveiller les journaux de la passerelle SiteWise Edge	1037
Utiliser Amazon CloudWatch Logs	1038
Utiliser les journaux de service	1040
Surveillez avec Amazon CloudWatch Metrics	1041
AWS IoT Greengrass Version 2 métriques de passerelle	1042
Enregistrez les appels d'API avec AWS CloudTrail	1051
AWS IoT SiteWise informations dans CloudTrail	1052
AWS IoT SiteWise événements de données dans CloudTrail	1053
AWS IoT SiteWise événements de gestion dans CloudTrail	1056
Exemple : entrées de fichier AWS IoT SiteWise journal	1056
Baliser vos ressources	1058
Utiliser des balises dans AWS IoT SiteWise	1058
Marquez avec le AWS Management Console	1058
Marquer avec l' AWS IoT SiteWise API	1059
Utiliser des balises avec des politiques IAM	1060
Résolution des problèmes	1062
Dépannage d'une passerelle	1062
Configuration et accès aux journaux de la passerelle SiteWise Edge	1062
Résolution des problèmes liés à la passerelle SiteWise Edge	1063
Résolution des problèmes liés à l'application AWS IoT SiteWise Edge sur Siemens Industrial Edge	1070
Résolution des problèmes liés aux intégrations open source à la périphérie	1071
AWS IoT Greengrass Problèmes de résolution des problèmes	1073
Résoudre les problèmes liés à un portail	1074
Les utilisateurs et les administrateurs ne peuvent pas accéder au AWS IoT SiteWise portail	1074
Résoudre les problèmes liés à une action de AWS IoT SiteWise règle	1075
Configuration AWS IoT Core des journaux	1076
Configurer une action d'erreur de republication	1077
Résoudre les problèmes liés aux règles	1079
Résoudre les problèmes liés à une règle ()AWS IoT SiteWise	1081
Résolution des problèmes liés à une règle (DynamoDB)	1082
Résoudre les problèmes d'importation et d'exportation en masse	1086
Points de terminaison et quotas	1088
Points de terminaison	1088
Quotas	1088

AWS IoT SiteWise Limites de limitation de l'API Assistant	1107
Quotas pour la détection des anomalies	1107
Historique de la documentation	1108
.....	mcxxxi

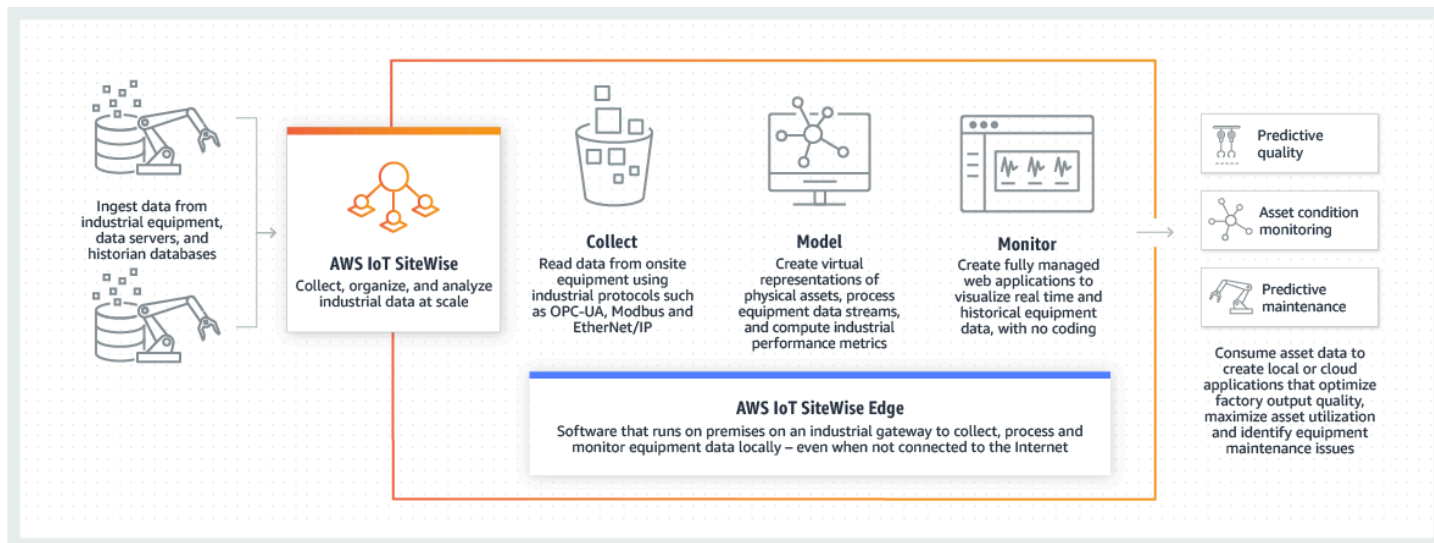
Qu'est-ce que c'est AWS IoT SiteWise ?

AWS IoT SiteWise est un service géré qui vous permet de collecter, de stocker, d'organiser et de surveiller les données provenant d'équipements industriels à grande échelle afin de vous aider à prendre de meilleures décisions basées sur les données. Vous pouvez l'utiliser AWS IoT SiteWise pour surveiller les opérations sur l'ensemble des installations, calculer rapidement les indicateurs de performance industrielle courants et créer des applications qui analysent les données des équipements industriels afin de prévenir les problèmes d'équipement coûteux et de réduire les écarts de production.

Vos utilisateurs opérationnels peuvent ainsi créer des applications Web pour visualiser et analyser vos données industrielles en temps réel. AWS IoT SiteWise Monitor Vous pouvez obtenir des informations sur vos opérations industrielles en configurant et en surveillant des mesures telles que le temps moyen entre les pannes et l'efficacité globale de l'équipement (OEE).

AWS IoT SiteWise Edge est un composant AWS IoT SiteWise qui permet de collecter, de stocker et de traiter des données sur des appareils locaux. Cela est utile si vous avez un accès limité à Internet ou si vous devez préserver la confidentialité de vos données.

Le schéma suivant montre l'architecture de base de AWS IoT SiteWise :



Rubriques

- [Comment AWS IoT SiteWise fonctionne](#)
- [Cas d'utilisation pour AWS IoT SiteWise](#)
- [Utilisation de ce service avec un AWS SDK](#)

- [AWS IoT SiteWise concepts](#)

Comment AWS IoT SiteWise fonctionne

AWS IoT SiteWise propose un cadre de modélisation des ressources que vous pouvez utiliser pour créer des représentations de vos appareils, processus et installations industriels. Les représentations de votre équipement et de vos processus sont appelées modèles d'actifs dans AWS IoT SiteWise. Avec les modèles d'actifs, vous définissez les données brutes à consommer et la manière de les transformer en indicateurs utiles. Créez et visualisez des actifs et des modèles pour vos opérations industrielles dans la [AWS IoT SiteWise console](#). Vous pouvez également configurer des modèles d'actifs pour collecter et traiter des données en périphérie ou dans le AWS cloud.

Rubriques

- [Ingérez des données industrielles](#)
- [Modélisez les actifs pour contextualiser les données collectées](#)
- [Analysez à l'aide de requêtes, d'alarmes et de prédictions](#)
- [Visualisez les opérations](#)
- [Stockez les données](#)
- [Intégrer à d'autres services](#)

Ingérez des données industrielles

Commencez à l'utiliser AWS IoT SiteWise en ingérant des données industrielles. L'ingestion de vos données s'effectue de différentes manières :

- Ingestion directe depuis les serveurs sur site : utilisez des protocoles tels que OPC UA pour lire les données directement à partir des appareils sur site. Déployez le logiciel de passerelle SiteWise Edge, compatible avec AWS IoT Greengrass V2, sur un large éventail de plateformes telles que les passerelles industrielles courantes ou les serveurs virtuels. Vous pouvez connecter jusqu'à 100 serveurs OPC UA à une seule AWS IoT SiteWise passerelle. Pour de plus amples informations, veuillez consulter [AWS IoT SiteWise Exigences relatives à la passerelle auto-hébergée Edge](#).

Notez que les protocoles tels que Modbus TCP et Ethernet/IP (EIP) sont pris en charge Domatoma dans le cadre de notre partenariat avec AWS IoT Greengrass V2

- Traitement des données Edge avec des packs : Améliorez votre passerelle SiteWise Edge en ajoutant des packs pour activer des fonctionnalités Edge complètes. Avec SiteWise Edge,

disponible sur AWS IoT Greengrass V2, le traitement des données est exécuté directement sur site avant d'être transmis de manière sécurisée vers le AWS Cloud via un AWS IoT Greengrass flux. Pour de plus amples informations, veuillez consulter [Configurer une source OPC UA dans Edge SiteWise](#).

- Ingestion adaptative via Amazon S3 avec opérations en masse : lorsque vous travaillez avec un grand nombre de ressources ou de modèles d'actifs, utilisez des opérations en masse pour importer et exporter des ressources en masse depuis des compartiments Amazon S3. Pour de plus amples informations, veuillez consulter [Opérations groupées avec actifs et modèles](#).
- Messages MQTT avec règles de AWS IoT base : pour les appareils connectés à AWS IoT Core qui envoient des messages MQTT, utilisez le moteur de règles AWS IoT Core pour diriger ces messages vers AWS IoT SiteWise. Si vous avez des appareils connectés à AWS IoT Core qui envoient des messages [MQTT](#), utilisez le moteur de règles AWS IoT Core pour acheminer ces messages vers. AWS IoT SiteWise Pour de plus amples informations, veuillez consulter [Ingérer des données pour AWS IoT SiteWise utiliser des règles AWS IoT Core](#).
- Ingestion de données déclenchée par un événement : utilisez AWS IoT Events des actions pour configurer l' SiteWise action IoT afin d' AWS IoT Events envoyer des données AWS IoT SiteWise lorsque des événements se produisent. Pour de plus amples informations, veuillez consulter [Ingérer des données vers AWS IoT SiteWiseAWS IoT Events](#).
- AWS IoT SiteWise API : vos applications en périphérie ou dans le cloud peuvent envoyer des données directement à AWS IoT SiteWise. Pour de plus amples informations, veuillez consulter [Ingérez des données avec AWS IoT SiteWise APIs](#).

Modélisez les actifs pour contextualiser les données collectées

Après avoir ingéré les données, vous pouvez les utiliser pour créer des représentations virtuelles de vos actifs, processus et installations en élaborant des modèles de vos opérations physiques. Un actif, représentant un appareil ou un processus, transmet des flux de données vers le AWS Cloud. Les actifs peuvent également signifier des groupements de périphériques logiques. Les hiérarchies sont formées en associant des actifs pour refléter des opérations complexes. Ces hiérarchies permettent aux actifs d'accéder aux données des actifs enfants associés. Les actifs sont créés à partir de modèles d'actifs. Les modèles d'actifs sont des structures déclaratives qui normalisent les formats des actifs. Réutilisez les composants des actifs pour l'organisation et la maintenabilité de vos modèles. Pour de plus amples informations, veuillez consulter [Modéliser des actifs industriels](#).

Avec AWS IoT SiteWise, vous pouvez configurer vos actifs pour transformer les données entrantes en mesures et transformations contextuelles.

- Transforme le travail lors de la réception des données de l'équipement.
- Les mesures sont calculées à des intervalles que vous définissez.

Les métriques et les transformations s'appliquent à la fois à des actifs individuels ou à plusieurs actifs. AWS IoT SiteWise calcule automatiquement les agrégats statistiques couramment utilisés tels que la moyenne, la somme et le nombre, sur différentes périodes pertinentes pour les données, les métriques et les transformations de votre équipement.

Les actifs peuvent être synchronisés à l'aide d'AWS IoT TwinMaker. Pour de plus amples informations, veuillez consulter [Intégration AWS IoT SiteWise et AWS IoT TwinMaker](#).

Analysez à l'aide de requêtes, d'alarmes et de prédictions

Analysez les données recueillies AWS IoT SiteWise en exécutant des requêtes et en configurant des alarmes. Vous pouvez également utiliser Amazon Lookout pour détecter automatiquement les anomalies dans les métriques et identifier leurs causes profondes.

- Définissez des alarmes spécifiques pour alerter votre équipe lorsque l'équipement ou les processus s'écartent des performances optimales, garantissant ainsi une identification et une résolution rapides des problèmes. Pour de plus amples informations, veuillez consulter [Surveillez les données avec des alarmes intégrées AWS IoT SiteWise](#).
- Utilisez les opérations de l'AWS IoT SiteWise API pour interroger les valeurs actuelles, les valeurs historiques et les agrégats des propriétés de vos actifs sur des intervalles de temps spécifiques. Pour de plus amples informations, veuillez consulter [Interrogez les données de AWS IoT SiteWise](#).
- Utilisez la détection des anomalies avec Amazon Lookout for Equipment pour identifier et visualiser les modifications de l'équipement ou des conditions de fonctionnement. Grâce à la détection des anomalies, vous pouvez déterminer les mesures de maintenance préventive pour vos opérations. Cette intégration permet aux clients de synchroniser les données entre Amazon Lookout for AWS IoT SiteWise Equipment et Amazon Lookout for Equipment. Pour de plus amples informations, veuillez consulter [Détectez les anomalies avec Lookout for Equipment](#).

Visualisez les opérations

Configurez SiteWise Monitor pour créer des applications Web pour vos employés opérationnels. Les applications Web aident les employés à visualiser vos opérations. Gérez différents niveaux d'accès pour vos employés à l'aide d'IAM Identity Center ou IAM. Configurez des connexions et des autorisations uniques pour chaque employé afin de consulter des sous-ensembles spécifiques de

l'ensemble d'une opération industrielle. AWS IoT SiteWise fournit un [guide d'application](#) permettant à ces employés d'apprendre à utiliser SiteWise Monitor.

Pour plus d'informations sur la visualisation de vos opérations, consultez [Surveillez les données avec AWS IoT SiteWise Monitor](#).

Stockez les données

Vous pouvez intégrer le stockage de séries chronologiques à votre lac de données industriel. AWS IoT SiteWise dispose de trois niveaux de stockage pour les données industrielles :

- Un niveau de stockage à chaud optimisé pour les applications en temps réel.
- Un niveau de stockage à chaud optimisé pour les charges de travail analytiques.
- Un niveau de stockage à froid géré par le client utilisant Amazon S3 pour les applications de données opérationnelles avec une tolérance de latence élevée.

AWS IoT SiteWise vous aide à gérer les coûts de stockage en conservant les données récentes dans le niveau de stockage à chaud. Vous définissez ensuite des politiques de conservation des données pour déplacer les données historiques vers un stockage à chaud ou à froid. Pour de plus amples informations, veuillez consulter [Gérez le stockage des données dans AWS IoT SiteWise](#).

Vous pouvez également importer et exporter les métadonnées des actifs. Pour de plus amples informations, veuillez consulter [Métadonnées des actifs](#).

Intégrer à d'autres services

AWS IoT SiteWise s'intègre à plusieurs AWS services pour développer une AWS IoT solution complète dans le AWS Cloud. Pour de plus amples informations, veuillez consulter [Interagissez avec d'autres AWS services](#).

Cas d'utilisation pour AWS IoT SiteWise

AWS IoT SiteWise est utilisé dans de nombreux secteurs pour de nombreuses applications de collecte et d'analyse de données industrielles.

Collectez des données de manière cohérente à partir de toutes vos sources afin de résoudre rapidement les problèmes. AWS IoT SiteWise propose une surveillance à distance pour collecter les données directement sur site ou à partir de sources multiples dans de nombreuses installations. AWS IoT SiteWise fournit la flexibilité nécessaire aux solutions de données IoT industrielles.

Fabrication

AWS IoT SiteWise peut simplifier le processus de collecte et d'utilisation des données de votre équipement afin d'identifier et de minimiser les inefficiences, améliorant ainsi les opérations industrielles. AWS IoT SiteWise vous aide à collecter des données sur les chaînes de fabrication et les équipements. Vous pouvez ainsi transférer les données vers le AWS cloud et créer des indicateurs de performance pour vos équipements et processus spécifiques. AWS IoT SiteWise Vous pouvez utiliser les indicateurs produits pour comprendre l'efficacité globale de vos opérations et identifier les opportunités d'innovation et d'amélioration. Vous pouvez également consulter votre processus de fabrication et identifier les défaillances des équipements et des processus, les écarts de production ou les défauts du produit.

Alimentation et boissons

Les installations du secteur de l'alimentation et des boissons gèrent une large variété d'opérations de transformation des aliments, notamment la mouture du grain en farine, le découpage et l'emballage de la viande, ainsi que l'assemblage, la préparation et la congélation de plats micro-ondables. Les usines de transformation des aliments s'étendent souvent sur plusieurs sites, les opérateurs d'usine et d'équipement étant centralisés pour surveiller les processus et les équipements. Par exemple, les unités de réfrigération évaluent la manipulation et l'expiration des ingrédients. Ils surveillent la création de déchets dans l'ensemble des installations afin de garantir l'efficacité opérationnelle. Vous pouvez ainsi regrouper les flux de données de capteurs provenant de plusieurs sites par ligne de production et par installation afin que vos ingénieurs de procédé puissent mieux comprendre et apporter des améliorations à l'ensemble des installations. AWS IoT SiteWise

Énergie et services publics

Avec AWS IoT SiteWise, vous pouvez résoudre les problèmes d'équipement plus facilement et plus efficacement. Vous pouvez surveiller les performances des actifs à distance et en temps réel. Accédez aux données historiques des équipements où que vous soyez pour identifier les problèmes potentiels, affecter des ressources précises et prévenir et résoudre les problèmes plus rapidement.

Utilisation de ce service avec un AWS SDK

AWS des kits de développement logiciel (SDKs) sont disponibles pour de nombreux langages de programmation courants. Chaque SDK fournit une API, des exemples de code et de la documentation qui facilitent la création d'applications par les développeurs dans leur langage préféré.

Documentation SDK	Exemples de code
AWS SDK pour C++	AWS SDK pour C++ exemples de code
AWS CLI	AWS CLI exemples de code
AWS SDK pour Go	AWS SDK pour Go exemples de code
AWS SDK pour Java	AWS SDK pour Java exemples de code
AWS SDK pour JavaScript	AWS SDK pour JavaScript exemples de code
AWS SDK pour Kotlin	AWS SDK pour Kotlin exemples de code
AWS SDK pour .NET	AWS SDK pour .NET exemples de code
AWS SDK pour PHP	AWS SDK pour PHP exemples de code
Outils AWS pour PowerShell	Outils AWS pour PowerShell exemples de code
AWS SDK pour Python (Boto3)	AWS SDK pour Python (Boto3) exemples de code
AWS SDK pour Ruby	AWS SDK pour Ruby exemples de code
Kit AWS SDK pour Rust	Kit AWS SDK pour Rust exemples de code
AWS SDK pour SAP ABAP	AWS SDK pour SAP ABAP exemples de code
Kit AWS SDK pour Swift	Kit AWS SDK pour Swift exemples de code

Exemple de disponibilité

Vous n'avez pas trouvé ce dont vous avez besoin ? Demandez un exemple de code en utilisant le lien Provide feedback (Fournir un commentaire) en bas de cette page.

AWS IoT SiteWise concepts

Les concepts de base suivants sont les AWS IoT SiteWise suivants :

Regrouper

Les agrégats sont des métriques fondamentales, ou mesures, qui sont calculées AWS IoT SiteWise automatiquement pour toutes les données de séries chronologiques. Pour de plus amples informations, veuillez consulter [Interrogez les agrégats de propriétés des actifs dans AWS IoT SiteWise](#).

Ressource

Lorsque vous saisissez ou ingérez des données AWS IoT SiteWise provenant de votre équipement industriel, vos appareils, équipements et processus sont tous présentés comme des actifs. Chaque actif est associé à des données. Par exemple, un équipement peut avoir un numéro de série, un emplacement, une marque et un modèle, ainsi qu'une date d'installation. Il peut également contenir des séries chronologiques pour la disponibilité, les performances, la qualité, la température, la pression, etc. Regroupez les actifs en hiérarchies pour permettre aux actifs d'accéder aux données stockées dans leurs actifs enfants. Pour de plus amples informations, veuillez consulter [Modéliser des actifs industriels](#).

Hiérarchie des ressources

Définissez des hiérarchies d'actifs pour créer des représentations logiques de vos opérations industrielles. Pour ce faire, définissez une hiérarchie dans un modèle d'actifs et associez les actifs créés à partir de ce modèle à la hiérarchie spécifiée. Les indicateurs des actifs parents peuvent combiner les données issues des propriétés des actifs enfants, ce qui vous permet de calculer des indicateurs qui fournissent des informations sur l'ensemble de vos opérations ou sur une partie spécifique de celles-ci. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).

Modèle de ressource

Chaque actif est créé à l'aide d'un modèle d'actif. Les modèles d'actifs sont des structures qui définissent et normalisent le format de vos actifs. Ils garantissent la cohérence des informations sur plusieurs actifs du même type, ce qui vous permet de gérer les données dans des actifs qui représentent des groupes d'appareils. Dans chaque modèle d'immobilisation, vous pouvez définir des [attributs](#), des entrées de séries chronologiques ([mesures](#)), des transformations de séries chronologiques ([transformations](#)), des agrégations de séries chronologiques ([métriques](#)) et [des hiérarchies de ressources](#). Pour de plus amples informations, veuillez consulter [Modéliser des actifs industriels](#).

Décidez où les propriétés de votre modèle d'actifs sont traitées en configurant votre modèle d'actif pour la périphérie. Utilisez cette fonctionnalité pour gérer et surveiller les données relatives aux actifs sur vos appareils locaux.

Propriété de la ressource

Les propriétés des actifs sont les structures de chaque actif qui contiennent des données industrielles. Chaque propriété possède un type de données et peut également avoir une unité. Une propriété peut être un [attribut](#), une [mesure](#), une [transformation](#), ou une [métrique](#). Pour de plus amples informations, veuillez consulter [Définir les propriétés des données](#).

Configurez les propriétés des actifs à calculer à la périphérie. Pour plus d'informations sur le traitement des données en périphérie, consultez [Configurer une source OPC UA dans Edge SiteWise](#).

Attribut

Les attributs sont des propriétés d'un actif qui restent généralement constantes, comme le fabricant ou l'emplacement de l'appareil. Les attributs peuvent avoir des valeurs prédéfinies. Chaque actif créé à partir d'un modèle d'actif inclut les valeurs par défaut des attributs définis dans ce modèle. Pour de plus amples informations, veuillez consulter [Définir des données statiques \(attributs\)](#).

Tableau de bord

Chaque projet contient un ensemble de tableaux de bord. Les tableaux de bord fournissent un ensemble de visualisations pour les valeurs d'un ensemble de ressources. Les propriétaires de projet créent les tableaux de bord et les visualisations qu'il contient. Lorsqu'un propriétaire de projet est prêt à partager l'ensemble de tableaux de bord, il peut inviter des utilisateurs dans le projet, ce qui leur donne accès à tous les tableaux de bord correspondants. Si vous souhaitez affecter différents groupes d'utilisateurs à différents tableaux de bord, vous devez diviser les tableaux de bord entre plusieurs projets. Lorsque les utilisateurs consultent les tableaux de bord, ils peuvent personnaliser la plage horaire pour examiner des données spécifiques.

Ensemble de données

Les ensembles de données sont des ensembles de données qui représentent des séries chronologiques, des données et des non-time-series données non liées à l'équipement, telles que les horaires de travail, les dossiers de maintenance et les bases de données des employés. Ils prennent en charge les données externes et utilisent des fonctionnalités AWS IoT SiteWise d'analyse. Il inclut les sources du jeu de données, le schéma du jeu de données et les paramètres

du jeu de données. L' AWS IoT SiteWise Assistant utilise des ensembles de données qui consomment les index Amazon Kendra.

Flux de données

Entrez ou ingérez des données industrielles avant AWS IoT SiteWise même de créer des modèles d'actifs et des actifs. AWS IoT SiteWise génère automatiquement des flux de données pour collecter des flux de données brutes provenant de votre équipement.

Alias de flux de données

Les alias de flux de données vous permettent d'identifier facilement un flux de données. Par exemple, l'alias `server1-windfarm/3/turbine/7/temperature` indique les valeurs de température provenant de la turbine #7 du parc éolien #3. Le terme `server1` est le nom de la source de données qui permet d'identifier le serveur OPC UA. `server1-` Il s'agit d'un préfixe attaché à tous les flux de données signalés par ce serveur OPC UA.

Association de flux de données

Après avoir créé des modèles d'actifs et des actifs, associez les flux de données aux propriétés des actifs définies dans vos actifs pour structurer vos données. AWS IoT SiteWise peut ensuite utiliser des modèles d'actifs et des actifs pour gérer les données entrantes provenant de vos flux de données. Vous pouvez également dissocier les flux de données des propriétés des actifs. Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).

Destinations

Les destinations dans SiteWise Edge représentent les points de terminaison auxquels vous souhaitez envoyer votre télémétrie ou vos données traitées. SiteWise Edge prend en charge le AWS IoT SiteWise hot tier, l'ingestion en mémoire tampon ou un compartiment Amazon S3 comme destinations. Vous pouvez configurer des destinations pour qu'elles s'abonnent à des sujets MQTT spécifiques à l'aide de filtres de chemin. Pour de plus amples informations, veuillez consulter [Comprendre les destinations AWS IoT SiteWise Edge](#).

Formule

Chaque propriété de [transformation](#) et de [métrique](#) est associée à une formule qui décrit la manière dont la propriété transforme ou agrège les données. Ces formules incluent les entrées de propriétés, les opérateurs et les fonctions proposés par AWS IoT SiteWise. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#).

Mesure

Les mesures sont des propriétés d'un actif qui représentent les flux de données chronologiques bruts d'un capteur provenant d'un appareil ou d'un équipement. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).

Métrique

Les métriques sont des propriétés d'un actif qui représentent des données de séries chronologiques agrégées. Chaque métrique est accompagnée d'une expression mathématique ([formule](#)) qui décrit comment agréger des points de données et un intervalle de temps pour le calcul de cette agrégation. Les métriques génèrent un point de données unique pour chaque intervalle de temps spécifié. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).

MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie léger pour les capteurs et les appareils.

Packages

SiteWise Les passerelles Edge utilisent des packs pour déterminer comment collecter, traiter et acheminer les données. Pour plus d'informations sur les packs disponibles pour votre passerelle SiteWise Edge, consultez [the section called "Utiliser des packs"](#).

Pack de collecte de données

Utilisez le pack de collecte de données afin que votre passerelle SiteWise Edge puisse collecter vos données industrielles et les acheminer vers la AWS destination de votre choix.

Pack de traitement de données

Utilisez le pack de traitement des données pour traiter, stocker et récupérer vos données en périphérie pendant 30 jours maximum. Échangez des données traitées en périphérie vers et depuis des applications locales via SiteWise Edge. APIs

OPC USA

OPC UA (Open Platform Communications Unified Architecture) est un protocole de communication pour l'automatisation industrielle.

Filtres de chemin

Utilisez des filtres de chemin au sein d'une passerelle pour vous abonner à des sujets MQTT et publier vers des destinations AWS IoT SiteWise prises en charge. Les sources, les pipelines de

traitement de données et les destinations basés sur MQTT échangent tous des données à l'aide de sujets MQTT sur une passerelle V3 auto-hébergée compatible MQTT. Vous pouvez définir des filtres thématiques pour spécifier les données que vous souhaitez ingérer ou acheminer vers différentes destinations.

Portail

Un AWS IoT SiteWise Monitor portail est une application Web que vous pouvez utiliser pour visualiser et partager vos AWS IoT SiteWise données. Un portail compte un ou plusieurs administrateurs et contient zéro ou plusieurs projets.

Administrateur du portail

Chaque portail SiteWise Monitor possède un ou plusieurs administrateurs de portail. Les administrateurs du portail utilisent ce dernier pour créer des projets contenant des collections de ressources et de tableaux de bord. Ils attribuent ensuite des ressources et des propriétaires à chaque projet. En contrôlant l'accès au projet, les administrateurs de portail spécifient les ressources que les propriétaires et les utilisateurs de projet peuvent voir.

Projet

Chaque portail SiteWise Monitor contient un ensemble de projets. Chaque projet est associé à un sous-ensemble de vos ressources AWS IoT SiteWise . Les propriétaires de projet créent un ou plusieurs tableaux de bord pour fournir un moyen cohérent de visualiser les données liées à ces ressources. Ils peuvent inviter des utilisateurs standard dans le projet pour leur permettre d'en consulter les ressources et les tableaux de bord. Le projet est l'unité de base du partage au sein de SiteWise Monitor. Les propriétaires de projets peuvent inviter les utilisateurs auxquels l' AWS administrateur a donné accès au portail. Tout utilisateur doit avoir accès à un portail avant qu'un projet de ce portail puisse être partagé avec lui.

Propriétaire du projet

Chaque projet SiteWise Monitor a des propriétaires. Ces propriétaires créent des visualisations sous la forme de tableaux de bord afin de représenter les données opérationnelles de manière cohérente. Lorsque les tableaux de bord sont prêts à être partagés, tout propriétaire du projet peut inviter des utilisateurs. Les propriétaires de projet peuvent également affecter d'autres propriétaires au projet. Les propriétaires de projets peuvent configurer des seuils et des paramètres de notification pour les alarmes.

Utilisateur de projet

Chaque projet SiteWise Monitor possède des spectateurs. Les utilisateurs de projet peuvent se connecter au portail pour consulter les tableaux de bord créés par les propriétaires de projet.

Dans chaque tableau de bord, les utilisateurs du projet peuvent ajuster la plage de temps pour mieux comprendre les données opérationnelles. Les utilisateurs de projet ne peuvent afficher que les tableaux de bord des projets auxquels ils ont accès. Les spectateurs du projet peuvent accuser réception des alarmes et les suspendre.

Alias de propriété

Vous avez la possibilité de créer des alias sur les propriétés des actifs, tels que le chemin de flux de données du serveur OPC UA (par exemple, /company/windfarm/3/turbine/7/temperature), afin de simplifier l'identification d'une propriété d'actif lors de l'ingestion ou de la récupération des données des actifs. Lorsque vous utilisez une [passerelle SiteWise Edge](#) pour ingérer des données provenant de serveurs, les alias de vos propriétés doivent correspondre aux chemins de vos flux de données brutes. Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).

Notification de propriété

Lorsque vous activez les notifications de propriété pour une propriété d'actif, AWS IoT SiteWise publie un message MQTT AWS IoT Core chaque fois que cette propriété reçoit une nouvelle valeur. La charge utile du message inclut des détails sur la mise à jour de cette valeur de propriété. Utilisez les notifications relatives à la valeur des propriétés pour créer des solutions qui connectent vos données industrielles AWS IoT SiteWise à d'autres AWS services. Pour de plus amples informations, veuillez consulter [Interagissez avec d'autres AWS services](#).

SiteWise Passerelle Edge

Une passerelle SiteWise Edge est installée dans les locaux du client pour collecter, traiter et diriger les données. Une passerelle SiteWise Edge se connecte à vos sources de données industrielles via différents protocoles pour collecter et traiter les données, puis les envoyer vers le AWS cloud. SiteWise Les passerelles Edge peuvent également se connecter aux sources de données partenaires. Pour de plus amples informations, veuillez consulter [Utiliser les passerelles AWS IoT SiteWise Edge](#).

Transformation

Les transformations sont des propriétés d'un actif qui représentent des données de séries chronologiques transformées. Chaque transformation est accompagnée d'une expression mathématique ([formule](#)) qui indique comment convertir les points de données d'un formulaire à un autre. Les points de données transformés entretiennent une one-to-one relation avec les points de données d'entrée. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).

Visualisation

Dans chaque tableau de bord, les propriétaires de projet décident comment afficher les propriétés et les alarmes des actifs associés au projet. La disponibilité peut être représentée sous forme de graphique linéaire, tandis que d'autres valeurs peuvent être affichées sous forme de diagrammes à barres ou d'indicateurs de performance clés (KPIs). Il est préférable d'afficher les alarmes sous forme de grilles d'état et de chronologies d'état. Les propriétaires de projet personnalisent chaque visualisation pour fournir une compréhension optimale des données de la ressource concernée.

Commencez avec AWS IoT SiteWise

Avec AWS IoT SiteWise, vous pouvez collecter, organiser, analyser et visualiser vos données.

AWS IoT SiteWise fournit une démonstration que vous pouvez utiliser pour explorer le service sans configurer une véritable source de données. Pour de plus amples informations, veuillez consulter [Utiliser la AWS IoT SiteWise démo](#).

Vous pouvez suivre les didacticiels suivants pour découvrir certaines fonctionnalités de AWS IoT SiteWise :

- [Ingérez des données dans AWS IoT SiteWise des objets AWS IoT](#)
- [Visualisez et partagez les données des parcs éoliens dans SiteWise Monitor](#)
- [Publier les mises à jour de la valeur des propriétés sur Amazon DynamoDB](#)

Consultez les rubriques suivantes pour en savoir plus sur AWS IoT SiteWise :

- [Ingérez des données dans AWS IoT SiteWise](#)
- [Modéliser des actifs industriels](#)
- [Configuration des fonctionnalités Edge sur AWS IoT SiteWise Edge](#)
- [Surveillez les données avec AWS IoT SiteWise Monitor](#)
- [Interrogez les données de AWS IoT SiteWise](#)
- [Interagissez avec d'autres AWS services](#)

Rubriques

- [Prérequis](#)
- [Configurez un AWS compte](#)

Prérequis

Vous devez avoir un AWS compte pour commencer AWS IoT SiteWise. Si vous n'en avez pas, consultez la section suivante pour savoir comment créer un compte.

Utilisez une région où elle AWS IoT SiteWise est disponible. Pour plus d'informations, consultez [Points de terminaison et quotas AWS IoT SiteWise](#). Vous pouvez utiliser le sélecteur de région dans le AWS Management Console pour passer à l'une de ces régions.

Configurez un AWS compte

Rubriques

- [Inscrivez-vous pour un Compte AWS](#)
- [Création d'un utilisateur doté d'un accès administratif](#)

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.

Une partie de la procédure d'inscription consiste à recevoir un appel téléphonique ou un message texte et à saisir un code de vérification sur le clavier 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 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, voir [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 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 .

Utiliser la AWS IoT SiteWise démo

Vous pouvez facilement explorer AWS IoT SiteWise en utilisant la AWS IoT SiteWise démo. AWS IoT SiteWise fournit la démo sous forme de AWS CloudFormation modèle que vous pouvez déployer pour créer des modèles d'actifs, des actifs et un portail de SiteWise surveillance, et générer des exemples de données pendant une semaine au maximum.

Important

Une fois que vous aurez créé la démo, les ressources créées et consommées par cette démo vous seront facturées.

Rubriques

- [Créez la AWS IoT SiteWise démo](#)
- [Supprimer la AWS IoT SiteWise démo](#)

Créez la AWS IoT SiteWise démo

Vous pouvez créer la AWS IoT SiteWise démo depuis la AWS IoT SiteWise console.

Note

La démo crée des fonctions Lambda, une règle d' CloudWatch événements et les rôles AWS Identity and Access Management (IAM) requis pour la démonstration. Vous pouvez voir ces ressources dans votre AWS compte. Nous vous recommandons de conserver ces ressources jusqu'à ce que vous ayez terminé la démo. Si vous supprimez les ressources, la démo risque de ne plus fonctionner correctement.

Pour créer la démo dans la AWS IoT SiteWise console

1. Accédez à la [AWS IoT SiteWise console](#) et trouvez la SiteWise démo dans le coin supérieur droit de la page.

2. (Facultatif) Sous SiteWise démo, modifiez le champ Jours pendant lesquels vous souhaitez conserver les ressources de démonstration pour spécifier le nombre de jours pendant lesquels vous devez conserver la démo avant de la supprimer.
3. (Facultatif) Pour créer un portail de SiteWise surveillance afin de surveiller des échantillons de données, procédez comme suit.

 Note

Les ressources du SiteWise moniteur créées et consommées par cette démo vous seront facturées. Pour plus d'informations, voir [SiteWise Surveiller](#) dans la section AWS IoT SiteWise Tarification.

- a. Choisissez Monitor Resources.
- b. Choisissez Autorisation.
- c. Choisissez un rôle IAM existant qui accorde à vos utilisateurs IAM fédérés l'accès au portail.

 Important

Votre rôle IAM doit disposer des autorisations suivantes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:Describe*",
        "iotsitewise:List*",
        "iotsitewise:Get*",
        "cloudformation:DescribeStacks",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedRolePolicies",
        "sso:DescribeRegisteredRegions",
        "organizations:DescribeOrganization"
      ],
      "Resource": "*"
    }
  ]
}
```

```
}
```

Pour plus d'informations sur l'utilisation de SiteWise Monitor, voir [Qu'est-ce que c'est AWS IoT SiteWise Monitor ?](#) dans le guide de AWS IoT SiteWise Monitor candidature.

4. Choisissez Créer une démonstration.

La démonstration prend environ 3 minutes à créer. Si la création de la démonstration échoue, votre compte peut ne pas disposer d'autorisations suffisantes. Basculez vers un compte disposant d'autorisations administratives ou suivez les étapes suivantes pour supprimer la démo et réessayez :

a. Choisissez Delete demo (Supprimer la démo).

La suppression de la démonstration prend environ 15 minutes.

b. Si la démo n'est pas supprimée, ouvrez la [AWS CloudFormation console](#), choisissez la pile nommée lo TSite WiseDemoAssets, puis choisissez Supprimer dans le coin supérieur droit.

c. Si la démo échoue à nouveau, suivez les étapes de la AWS CloudFormation console pour ignorer les ressources qui n'ont pas pu être supprimées, puis réessayez.

5. Une fois la démo créée avec succès, vous pouvez explorer les ressources et les données de démonstration dans la [AWS IoT SiteWise console](#).

Supprimer la AWS IoT SiteWise démo

La AWS IoT SiteWise démo se supprime d'elle-même au bout d'une semaine, ou après le nombre de jours que vous avez choisi si vous avez créé la pile de démonstration depuis la AWS CloudFormation console. Vous pouvez supprimer la démo avant si vous avez fini d'utiliser les ressources de démonstration. Vous pouvez également supprimer la démo si la démo ne parvient pas à créer. Suivez les étapes suivantes pour supprimer manuellement la démo.

Pour supprimer la AWS IoT SiteWise démo

1. Accédez à la [console AWS CloudFormation](#).
2. Choisissez IoTSiteWiseDemoAssets depuis la liste des Stacks.
3. Sélectionnez Delete (Supprimer).

Lorsque vous supprimez la pile, toutes les ressources créées pour la démonstration sont supprimées.

4. Dans la boîte de dialogue de confirmation, choisissez Delete (Supprimer).

La suppression de la pile prend environ 15 minutes. Si la suppression de la démonstration échoue, choisissez à nouveau Delete (Supprimer) dans le coin supérieur droit. Si la démo échoue à nouveau, suivez les étapes de la AWS CloudFormation console pour ignorer les ressources qui n'ont pas pu être supprimées, puis réessayez.

AWS IoT SiteWise tutoriels

Bienvenue sur la page des AWS IoT SiteWise tutoriels. Cette collection croissante de didacticiels vous permet d'acquérir les connaissances et les compétences nécessaires pour naviguer dans les subtilités de. AWS IoT SiteWise Ces didacticiels proposent un large éventail de sujets de base adaptés à vos besoins. En parcourant les didacticiels, découvrez des informations précieuses sur divers aspects de AWS IoT SiteWise.

Chaque tutoriel utilise un exemple d'équipement spécifique. Ces didacticiels sont destinés aux environnements de test et utilisent des noms de sociétés, des modèles, des actifs, des propriétés fictifs, etc. Leur objectif consiste à fournir des instructions générales. Les didacticiels ne sont pas destinés à être utilisés directement dans un environnement de production sans un examen minutieux et une adaptation pour répondre aux besoins uniques de votre organisation.

Rubriques

- [Calculez l'OEE en AWS IoT SiteWise](#)
- [Ingérez des données dans AWS IoT SiteWise des objets AWS IoT](#)
- [Visualisez et partagez les données des parcs éoliens dans SiteWise Monitor](#)
- [Publier les mises à jour de la valeur des propriétés sur Amazon DynamoDB](#)

Calculez l'OEE en AWS IoT SiteWise

Ce didacticiel fournit un exemple de la façon de calculer l'efficacité globale d'équipement (OEE) pour un processus de fabrication. Pour cette raison, vos calculs ou formules d'OEE peuvent être différents de ceux illustrés ici. En général, l'OEE est définie comme $\text{Availability} * \text{Quality} * \text{Performance}$. Pour en savoir plus sur le calcul de l'OEE, veuillez consulter [Efficacité globale de l'équipement](#) sur Wikipédia.

Prérequis

Pour terminer ce didacticiel, vous devez configurer l'ingestion de données pour un appareil qui possède les trois flux de données suivants :

- `Equipment_State`— Code numérique qui représente l'état de la machine, tel que le ralenti, le défaut, l'arrêt planifié ou le fonctionnement normal.

- **Good_Count**— Un flux de données dans lequel chaque point de données contient le nombre d'opérations réussies depuis le dernier point de données.
- **Bad_Count**— Un flux de données dans lequel chaque point de données contient le nombre d'opérations infructueuses depuis le dernier point de données.

Pour configurer l'ingestion de données, veuillez consulter [Ingérez des données dans AWS IoT SiteWise](#). Si vous n'avez pas d'opération industrielle disponible, vous pouvez écrire un script qui génère et télécharge des exemples de données via l'API AWS IoT SiteWise .

Calcul de l'OEE

Dans ce didacticiel, vous créez un modèle de ressource qui calcule l'OEE à partir de trois flux d'entrée de données : **Equipment_State**, **Good_Count** et **Bad_Count**. Dans cet exemple, considérez une machine d'emballage générique, telle qu'une machine utilisée pour l'emballage du sucre, des chips ou de la peinture. Dans la [AWS IoT SiteWise console](#), créez un modèle AWS IoT SiteWise d'actif avec les mesures, transformations et métriques suivantes. Ensuite, vous pouvez créer un actif pour représenter la machine d'emballage et observer comment l'OEE est AWS IoT SiteWise calculée.

Définissez les [mesures](#) suivantes pour représenter les flux de données brutes de la machine de conditionnement.

Mesures

- **Equipment_State**— Un flux de données (ou mesure) qui fournit l'état actuel de la machine d'emballage sous forme de codes numériques :
 - 1024— La machine est inactive.
 - 1020— Un défaut, tel qu'une erreur ou un retard.
 - 1000— Un arrêt planifié.
 - 1111— Un fonctionnement normal.
- **Good_Count**— Un flux de données dans lequel chaque point de données contient le nombre d'opérations réussies depuis le dernier point de données.
- **Bad_Count**— Un flux de données dans lequel chaque point de données contient le nombre d'opérations infructueuses depuis le dernier point de données.

À l'aide du flux de données de mesure `Equipment_State` et des codes qu'il contient, définissez les [transformations](#) (ou mesures dérivées) suivantes. Les transformations ont une one-to-one relation avec les mesures brutes.

Transformations

- `Idle = eq(Equipment_State, 1024)`— Un flux de données transformé qui contient l'état inactif de la machine.
- `Fault = eq(Equipment_State, 1020)`— Un flux de données transformé qui contient l'état de défaillance de la machine.
- `Stop = eq(Equipment_State, 1000)`— Un flux de données transformé qui contient l'état d'arrêt prévu de la machine.
- `Running = eq(Equipment_State, 1111)`— Un flux de données transformé qui contient l'état de fonctionnement normal de la machine.

À l'aide des mesures brutes et transformées, définissez les [métriques](#) suivantes qui regroupent les données machine sur des intervalles de temps spécifiés. Choisissez le même intervalle de temps pour chaque métrique lorsque vous définissez les métriques dans cette section.

Métriques

- `Successes = sum(Good_Count)`— Le nombre de colis remplis avec succès au cours de l'intervalle de temps spécifié.
- `Failures = sum(Bad_Count)`— Le nombre de colis remplis sans succès au cours de l'intervalle de temps spécifié.
- `Idle_Time = statetime(Idle)`— Durée totale d'inactivité de la machine (en secondes) par intervalle de temps spécifié.
- `Fault_Time = statetime(Fault)`— Durée totale de panne de la machine (en secondes) par intervalle de temps spécifié.
- `Stop_Time = statetime(Stop)`— Le temps d'arrêt total prévu de la machine (en secondes) par intervalle de temps spécifié.
- `Run_Time = statetime(Running)`— Durée totale (en secondes) de fonctionnement de la machine sans problème par intervalle de temps spécifié.
- `Down_Time = Idle_Time + Fault_Time + Stop_Time`— Le temps d'arrêt total de la machine (en secondes) sur l'intervalle de temps spécifié, calculé comme la somme des états de la machine autres que `Run_Time`.

- $Availability = Run\_Time / (Run\_Time + Down\_Time)$ — Temps de disponibilité de la machine ou pourcentage du temps prévu pendant lequel la machine est disponible pour fonctionner pendant l'intervalle de temps spécifié.
- $Quality = Successes / (Successes + Failures)$ — Le pourcentage de colis remplis avec succès par la machine sur les intervalles de temps spécifiés.
- $Performance = ((Successes + Failures) / Run\_Time) / Ideal\_Run\_Rate$ — Les performances de la machine sur l'intervalle de temps spécifié, en pourcentage par rapport à la cadence de fonctionnement idéale (en secondes) pour votre processus.

Par exemple, `Ideal_Run_Rate` peut être de 60 paquets par minute (1 paquet par seconde). Si votre `Ideal_Run_Rate` valeur est exprimée par minute ou par heure, vous devez la diviser par le facteur de conversion unitaire approprié, car `Run_Time` est exprimé en secondes.

- $OEE = Availability * Quality * Performance$ — L'efficacité globale de l'équipement de la machine sur l'intervalle de temps spécifié. Cette formule calcule l'OEE comme une fraction de 1.

Note

Si l'OEE est défini comme une transformation, les valeurs de sortie sont calculées pour chacune des valeurs d'entrée. Il est possible que des valeurs inattendues soient générées car l'évaluation de la transformation prend en compte les dernières valeurs disponibles pour toutes les propriétés contributives de la formule. Pour les mises à jour de propriétés avec le même horodatage, les valeurs de sortie peuvent être remplacées par des mises à jour provenant d'autres propriétés entrantes. Par exemple, lorsque la disponibilité, la qualité et les performances sont calculées, l'OEE est calculé avec les derniers points de données disponibles pour les deux autres propriétés. Ces valeurs contributives partagent des horodatages et entraînent des valeurs de sortie incorrectes de l'OEE. L'ordre n'est pas garanti pour le calcul des transformations.

Ingérez des données dans AWS IoT SiteWise des objets AWS IoT

Découvrez comment intégrer des données à AWS IoT SiteWise partir d'une multitude d' AWS IoT objets en utilisant les ombres des appareils dans ce didacticiel. Les ombres de périphérique sont des objets JSON qui stockent les informations d'état actuel d'un AWS IoT appareil. Pour plus d'informations, voir [Device Shadow Service](#) dans le Guide du AWS IoT développeur.

Après avoir terminé ce didacticiel, vous pouvez configurer une opération en AWS IoT SiteWise fonction AWS IoT des éléments. En utilisant AWS IoT des objets, vous pouvez intégrer vos opérations à d'autres fonctionnalités utiles de AWS IoT. Par exemple, vous pouvez configurer les AWS IoT fonctionnalités pour effectuer les tâches suivantes :

- Configurez des règles supplémentaires pour diffuser des données vers [AWS IoT Events](#) [Amazon DynamoDB](#) et d'autres services. AWS Pour plus d'informations, consultez la section [Règles](#) du guide du AWS IoT développeur.
- Indexez, recherchez et agrégez les données de vos appareils grâce au service d'indexation de AWS IoT flotte. Pour plus d'informations, consultez la section [Service d'indexation de flotte](#) dans le Guide du AWS IoT développeur.
- Auditez et sécurisez vos appareils avec AWS IoT Device Defender. Pour plus d'informations, consultez [AWS IoT Device Defender](#) dans le Guide du développeur AWS IoT .

Dans ce didacticiel, vous allez apprendre à ingérer des données depuis les zones d'ombre AWS IoT des appareils vers les actifs qu'ils contiennent. AWS IoT SiteWise Pour ce faire, vous créez un ou plusieurs AWS IoT éléments et exécutez un script qui met à jour l'ombre du périphérique de chaque élément avec les données d'utilisation du processeur et de la mémoire. Vous utiliserez les données d'utilisation de l'UC et de la mémoire dans ce didacticiel pour imiter des données de capteur réalistes. Ensuite, vous créez une règle avec une AWS IoT SiteWise action qui envoie ces données à un actif AWS IoT SiteWise chaque fois que le device shadow d'un objet est mis à jour. Pour de plus amples informations, veuillez consulter [Ingérer des données pour AWS IoT SiteWise utiliser des règles AWS IoT Core](#).

Rubriques

- [Prérequis](#)
- [Étape 1 : créer une AWS IoT politique](#)
- [Étape 2 : créer et configurer n'importe quel AWS IoT objet](#)
- [Étape 3 : Création d'un modèle de ressource d'appareil](#)
- [Étape 4 : Création d'un modèle de parc d'appareils](#)
- [Étape 5 : Création et configuration d'un actif d'appareil](#)
- [Étape 6 : Création et configuration d'un parc d'appareils](#)
- [Étape 7 : créer une règle dans AWS IoT Core pour envoyer des données aux actifs de l'appareil](#)
- [Étape 8 : Exécuter le script client de l'appareil](#)

- [Étape 9 : Nettoyer les ressources après le didacticiel](#)

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des éléments suivants :

- Un AWS compte. Si vous n'en avez pas, veuillez consulter [Configurez un AWS compte](#).
- Un ordinateur de développement exécutant Windows, macOS, Linux, ou Unix permettant d'accéder au AWS Management Console. Pour plus d'informations, consultez [Démarrer avec le AWS Management Console](#).
- Un utilisateur AWS Identity and Access Management (IAM) doté d'autorisations d'administrateur.
- Python3 installé sur votre ordinateur de développement ou installé sur l'appareil que vous souhaitez enregistrer en tant qu' AWS IoT objet.

Étape 1 : créer une AWS IoT politique

Dans cette procédure, créez une AWS IoT politique permettant à vos AWS IoT objets d'accéder aux ressources utilisées dans ce didacticiel.

Pour créer une AWS IoT politique

1. Connectez-vous à la [AWS Management Console](#).
2. Passez en revue les [AWS régions dans](#) lesquelles le AWS IoT SiteWise support est pris en charge. Basculez vers l'une de ces régions prises en charge, si nécessaire.
3. Accédez à la [console AWS IoT](#). Si un bouton Connect device apparaît, sélectionnez-le.
4. Dans le volet de navigation de gauche, choisissez Security, puis Policies.
5. Choisissez Créer.
6. Entrez le nom de la AWS IoT politique (par exemple, **SiteWiseTutorialDevicePolicy**).
7. Sous Document de stratégie, choisissez JSON pour saisir la politique suivante au format JSON. Remplacez *region* et *account-id* par votre région et votre numéro de compte, tels que **us-east-1** et **123456789012**.

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

```

    "Effect": "Allow",
    "Action": "iot:Connect",
    "Resource": "arn:aws:iot:region:account-id:client/SiteWiseTutorialDevice*"
  },
  {
    "Effect": "Allow",
    "Action": "iot:Publish",
    "Resource": [
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/update",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/delete",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/get"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iot:Receive",
    "Resource": [
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/update/rejected",
      "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/delete/rejected"
    ]
  },
  {
    "Effect": "Allow",
    "Action": "iot:Subscribe",
    "Resource": [
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/update/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/delete/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/get/accepted",
      "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/update/rejected",

```

```
    "arn:aws:iot:region:account-id:topicfilter/$aws/things/  
    ${iot:Connection.Thing.ThingName}/shadow/delete/rejected"  
  ],  
  {  
    "Effect": "Allow",  
    "Action": [  
      "iot:GetThingShadow",  
      "iot:UpdateThingShadow",  
      "iot>DeleteThingShadow"  
    ],  
    "Resource": "arn:aws:iot:region:account-id:thing/SiteWiseTutorialDevice*"  
  }  
]
```

Cette politique permet à vos AWS IoT appareils d'établir des connexions et de communiquer avec leurs ombres à l'aide de messages MQTT. Pour plus d'informations sur les messages MQTT, consultez [Qu'est-ce que MQTT ?](#). Pour interagir avec les ombres des appareils, vos AWS IoT objets publient et reçoivent des messages MQTT sur des sujets commençant \$aws/things/*thing-name*/shadow/ par. Cette politique intègre une variable de politique d'objets connue sous le nom de `${iot:Connection.Thing.ThingName}`. Cette variable remplace le nom de l'objet connecté dans chaque rubrique. L'`iot:Connect` instruction définit des limites quant aux appareils autorisés à établir des connexions, en veillant à ce que la variable `thing` policy ne puisse remplacer que les noms commençant par `SiteWiseTutorialDevice`.

Pour plus d'informations, consultez la section [Variables de politique des](#) objets dans le manuel du AWS IoT développeur.

Note

Cette stratégie s'applique aux objets dont le nom commence par `SiteWiseTutorialDevice`. Pour utiliser un nom différent pour vos objets, vous devez mettre à jour la stratégie en conséquence.

8. Choisissez Créer.

Étape 2 : créer et configurer n'importe quel AWS IoT objet

Dans cette procédure, vous créez et configurez n'importe quel AWS IoT objet. Vous pouvez désigner votre ordinateur de développement comme n'importe quel AWS IoT objet. Au fur et à mesure que vous progressez, n'oubliez pas que les principes que vous apprenez ici peuvent être appliqués à des projets réels. Vous avez la flexibilité de créer et de configurer des AWS IoT éléments sur n'importe quel appareil capable d'exécuter un AWS IoT SDK, y compris AWS IoT Greengrass FreeRTOS. Pour plus d'informations, consultez [AWS IoT SDKs](#) dans le Guide du développeur AWS IoT .

Pour créer et configurer n'importe quel AWS IoT objet

1. Ouvrez une ligne de commande et exécutez la commande suivante pour créer un répertoire pour ce didacticiel.

```
mkdir iot-sitewise-rule-tutorial
cd iot-sitewise-rule-tutorial
```

2. Exécutez la commande suivante pour créer un répertoire pour les certificats de l'objet.

```
mkdir device1
```

Si vous créez des objets supplémentaires, incrémentez le nombre en conséquence dans le nom du répertoire pour pouvoir déterminer quel certificat appartient à quel objet.

3. Accédez à la [console AWS IoT](#).
4. Dans le volet de navigation de gauche, choisissez Tous les appareils dans la section Gérer. Ensuite, choisissez Things (Objets).
5. Si une boîte de dialogue You don't have any things yet (Vous n'avez pas encore d'objet) s'affiche, choisissez Create a thing (Créer un objet). Sinon, choisissez Create things.
6. Sur la page Création d'objets, choisissez Créer un objet unique, puis cliquez sur Suivant.
7. Sur la page Spécifier les propriétés de l'objet, entrez le nom de votre AWS IoT objet (par exemple, **SiteWiseTutorialDevice1**) puis choisissez Suivant. Si vous créez des objets supplémentaires, incrémentez le nombre en conséquence dans le nom de l'objet.

 Important

Le nom de l'objet doit correspondre au nom utilisé dans la politique que vous avez créée à l'étape 1 : Création d'une AWS IoT politique. Dans le cas contraire, votre appareil ne pourra pas se connecter à AWS IoT.

8. Sur la page Configurer le certificat de l'appareil - facultatif, choisissez Générer automatiquement un nouveau certificat (recommandé), puis cliquez sur Suivant. Les certificats permettent AWS IoT d'identifier vos appareils en toute sécurité.
9. Sur la page facultative Attacher des politiques au certificat, sélectionnez la politique que vous avez créée à l'étape 1 : Création d'une AWS IoT politique, puis choisissez Créer un objet.
10. Dans la boîte de dialogue Télécharger les certificats et les clés, procédez comme suit :
 - a. Choisissez les liens Télécharger pour télécharger le certificat, la clé publique et la clé privée de votre objet. Enregistrez les trois fichiers dans le répertoire que vous avez créé pour les certificats de l'objet (par exemple, `iot-sitewise-rule-tutorial/device1`).

 Important

Il s'agit de l'unique fois où vous pouvez télécharger le certificat et les clés de l'objet, sans lesquels votre appareil ne pourra pas se connecter à AWS IoT.

- b. Cliquez sur le lien Télécharger pour télécharger un certificat CA racine. Enregistrez le certificat d'autorité de certification racine dans `iot-sitewise-rule-tutorial`. Nous vous recommandons de télécharger Amazon Root CA 1.
11. Sélectionnez Exécuté.

Vous avez maintenant enregistré AWS IoT quelque chose sur votre ordinateur. Procédez à l'une des étapes suivantes :

- Passez à l'étape 3 : Création d'un modèle de ressource d'appareil sans créer d' AWS IoT éléments supplémentaires. Vous pouvez réaliser ce didacticiel avec un seul objet.
- Répétez les étapes de cette section sur un autre ordinateur ou périphérique pour créer d'autres objets AWS IoT . Pour ce didacticiel, nous vous recommandons de suivre cette étape afin que vous puissiez ingérer des données d'utilisation d'UC et de mémoire uniques à partir de plusieurs appareils.

- Répétez les étapes de cette section sur le même périphérique (ordinateur) pour créer davantage d'objets AWS IoT . Chaque AWS IoT appareil reçoit des données d'utilisation du processeur et de la mémoire similaires de votre ordinateur. Utilisez cette approche pour démontrer l'ingestion de données non uniques provenant de plusieurs appareils.

Étape 3 : Création d'un modèle de ressource d'appareil

Dans cette procédure, vous créez un modèle d'actif AWS IoT SiteWise pour représenter vos appareils qui diffusent des données d'utilisation du processeur et de la mémoire. Pour traiter les données des actifs qui représentent des groupes d'appareils, les modèles d'actifs appliquent des informations cohérentes sur plusieurs actifs du même type. Pour de plus amples informations, veuillez consulter [Modéliser des actifs industriels](#).

Pour créer un modèle de ressource qui représente un périphérique

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Models (Modèles).
3. Sélectionnez Create model.
4. Sous Détails du modèle, entrez le nom de votre modèle. Par exemple, **SiteWise Tutorial Device Model**.
5. Sous Measurement definitions (Définitions de mesure), procédez comme suit :
 - a. Pour Name (Nom), entrez **CPU Usage**.
 - b. Pour Unit (Unité), saisissez %.
 - c. Conservez le type de données Double.

Les propriétés de mesure représentent les flux de données brutes d'un périphérique. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).

6. Choisissez Ajouter une nouvelle mesure pour ajouter une deuxième propriété de mesure.
7. Dans la deuxième ligne sous Measurement definitions (Définitions de mesure), procédez comme suit :
 - a. Pour Name (Nom), entrez **Memory Usage**.
 - b. Pour Unit (Unité), saisissez %.

- c. Conservez le type de données Double.
8. Sous Metric definitions (Définitions de métrique), procédez comme suit :
- a. Pour Name (Nom), entrez **Average CPU Usage**.
 - b. Pour Formula (Formule), saisissez **avg(CPU Usage)**. Choisissez CPU Usage dans la liste de saisie semi-automatique lorsqu'elle apparaît.
 - c. Dans Time interval (Intervalle de temps), entrez **5 minutes**.

Les propriétés de métrique définissent les calculs d'agrégation qui traitent tous les points de données en entrée sur un intervalle et qui produisent un point de données unique par intervalle. Cette propriété de métrique calcule l'utilisation moyenne de l'UC de chaque périphérique toutes les 5 minutes. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).

9. Choisissez Ajouter une nouvelle métrique pour ajouter une deuxième propriété de métrique.
10. Dans la deuxième ligne sous Metric definitions (Définitions de métrique), procédez comme suit :
- a. Pour Name (Nom), entrez **Average Memory Usage**.
 - b. Pour Formula (Formule), saisissez **avg(Memory Usage)**. Choisissez Memory Usage dans la liste de saisie semi-automatique lorsqu'elle apparaît.
 - c. Dans Time interval (Intervalle de temps), entrez **5 minutes**.

Cette propriété de métrique calcule l'utilisation moyenne de la mémoire de chaque périphérique toutes les 5 minutes.

11. (Facultatif) Ajoutez les autres métriques que vous souhaitez calculer pour chaque périphérique. Certaines fonctions intéressantes comprennent min et max. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#). À l'étape 4 : Création d'un modèle d'actifs de parc d'appareils, vous créez un actif parent capable de calculer des mesures à l'aide des données de l'ensemble de votre parc d'appareils.
12. Sélectionnez Create model.

Étape 4 : Création d'un modèle de parc d'appareils

Dans cette procédure, vous créez un modèle de ressource AWS IoT SiteWise pour symboliser votre collection d'appareils. Dans ce modèle d'actifs, vous établissez une structure qui vous permet de

relier de nombreux équipements à un actif de flotte global. Ensuite, vous définissez les mesures du modèle d'actifs du parc afin de consolider les données de tous les actifs des appareils connectés. Cette approche vous fournit des informations complètes sur les performances collectives de l'ensemble de votre flotte.

Pour créer un modèle de ressource qui représente un parc de périphériques

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Models (Modèles).
3. Sélectionnez Create model.
4. Sous Détails du modèle, entrez le nom de votre modèle. Par exemple, **SiteWise Tutorial Device Fleet Model**.
5. Sous Hierarchy definitions (Définitions de hiérarchie), procédez comme suit :
 - a. Dans Hierarchy name (Nom de la hiérarchie), entrez **Device**.
 - b. Dans Hierarchy model (Modèle de hiérarchie), choisissez votre modèle de ressource de périphérique (**SiteWise Tutorial Device Model**).

Une hiérarchie définit une relation entre un modèle de ressource parent (parc) et un modèle de ressource enfant (périphérique). Les ressources parents peuvent accéder aux données de propriété des ressources enfants. Lorsque vous créez des ressources par la suite, vous devrez associer des ressources enfants aux ressources parents selon une définition de hiérarchie dans le modèle de ressource parent. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).

6. Sous Metric definitions (Définitions de métrique), procédez comme suit :
 - a. Pour Name (Nom), entrez **Average CPU Usage**.
 - b. Pour Formula (Formule), saisissez **avg(Device | Average CPU Usage)**. Lorsque la liste de saisie semi-automatique apparaît, choisissez Device pour sélectionner une hiérarchie, puis choisissez Average CPU Usage pour sélectionner la métrique à partir de la ressource de périphérique que vous avez créée précédemment.
 - c. Dans Time interval (Intervalle de temps), entrez **5 minutes**.

Cette propriété de métrique calcule l'utilisation moyenne de l'UC de toutes les ressources de périphériques associées à une ressource de parc via la hiérarchie **Device**.

7. Choisissez Ajouter une nouvelle métrique pour ajouter une deuxième propriété de métrique.
8. Dans la deuxième ligne sous Metric definitions (Définitions de métrique), procédez comme suit :
 - a. Pour Name (Nom), entrez **Average Memory Usage**.
 - b. Pour Formula (Formule), saisissez **avg(Device | Average Memory Usage)**. Lorsque la liste de saisie semi-automatique apparaît, choisissez Device pour sélectionner une hiérarchie, puis choisissez Average Memory Usage pour sélectionner la métrique à partir de la ressource de périphérique que vous avez créée précédemment.
 - c. Dans Time interval (Intervalle de temps), entrez **5 minutes**.

Cette propriété de métrique calcule l'utilisation moyenne de la mémoire de toutes les ressources de périphériques associées à une ressource de parc via la hiérarchie **Device**.

9. (Facultatif) Ajoutez les autres métriques que vous souhaitez calculer pour l'ensemble de votre parc de périphériques.
10. Sélectionnez Create model.

Étape 5 : Création et configuration d'un actif d'appareil

Dans cette procédure, vous générez un actif d'appareil basé sur le modèle d'actif de votre appareil. Ensuite, vous définirez des alias de propriété pour chaque propriété de mesure. Un alias de propriété est une chaîne unique qui identifie la propriété d'un actif. Plus tard, vous pourrez identifier une propriété pour le téléchargement des données en utilisant les alias au lieu de l'ID de ressource et de l'ID de propriété. Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).

Pour créer une ressource de périphérique et définir des alias de propriété

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation de gauche, choisissez Assets (Ressources).
3. Choisissez Create asset (Créer une ressource).
4. Sous Informations sur le modèle, choisissez le modèle de ressource de votre appareil, **SiteWise Tutorial Device Model**.
5. Sous Informations sur l'actif, entrez le nom de votre actif. Par exemple, **SiteWise Tutorial Device 1**.
6. Choisissez Create asset (Créer une ressource).

7. Pour la nouvelle ressource de périphérique, choisissez Edit (Modifier).
8. Sous CPU Usage, entrez **/tutorial/device/SiteWiseTutorialDevice1/cpu** comme alias de propriété. Vous incluez le nom de l' AWS IoT objet dans l'alias de propriété, afin de pouvoir ingérer les données de tous vos appareils à l'aide d'une seule AWS IoT règle.
9. Sous Memory Usage, entrez **/tutorial/device/SiteWiseTutorialDevice1/memory** comme alias de propriété.
10. Choisissez Enregistrer.

Si vous avez créé plusieurs AWS IoT éléments précédemment, répétez les étapes 3 à 10 pour chaque appareil, puis incrémentez le numéro dans le nom de l'actif et les alias de propriété en conséquence. Par exemple, le nom de la deuxième ressource de périphérique doit être **SiteWise Tutorial Device 2**, et ses alias de propriété doivent être **/tutorial/device/SiteWiseTutorialDevice2/cpu** et **/tutorial/device/SiteWiseTutorialDevice2/memory**.

Étape 6 : Création et configuration d'un parc d'appareils

Dans cette procédure, vous créez un actif de parc d'appareils dérivé de votre modèle d'actifs de parc d'appareils. Ensuite, vous liez les actifs de vos appareils individuels à l'actif du parc. Cette association permet aux propriétés métriques de l'actif du parc de compiler et d'analyser les données provenant de plusieurs appareils. Ces données vous fournissent une vue consolidée des performances collectives de l'ensemble de la flotte.

Pour créer une ressource de parc de périphériques et associer des ressources de périphérique

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation de gauche, choisissez Assets (Ressources).
3. Choisissez Create asset (Créer une ressource).
4. Sous Informations sur le modèle, choisissez le modèle d'actif de votre parc d'appareils, **SiteWise Tutorial Device Fleet Model**.
5. Sous Informations sur l'actif, entrez le nom de votre actif. Par exemple, **SiteWise Tutorial Device Fleet 1**.
6. Choisissez Create asset (Créer une ressource).
7. Pour la nouvelle ressource de parc de périphériques, choisissez Edit (Modifier).

8. Sous Ressources associées à cette ressource, choisissez Ajouter une ressource associée et procédez comme suit :
 - a. Sous Hierarchy (Hiérarchie), choisissez Device. Cette hiérarchie identifie la relation hiérarchique entre les ressources de périphériques et de parc de périphériques. Vous avez défini cette hiérarchie dans le modèle de ressource de parc de périphériques précédemment dans ce didacticiel.
 - b. Sous Asset (Ressource), sélectionnez votre ressource de périphérique, SiteWise Tutorial Device 1.
9. (Facultatif) Si vous avez créé plusieurs actifs d'appareil précédemment, répétez les étapes 8 à 10 pour chaque actif d'appareil que vous avez créé.
10. Choisissez Enregistrer.

Vous devriez maintenant voir les ressources de votre périphérique organisées sous forme de hiérarchie.

Étape 7 : créer une règle dans AWS IoT Core pour envoyer des données aux actifs de l'appareil

Dans cette procédure, vous établissez une règle dans AWS IoT Core. La règle est conçue pour interpréter les messages de notification provenant des ombres de l'appareil et pour transmettre les données aux actifs de votre appareil en AWS IoT SiteWise. Chaque fois que le shadow de votre appareil est mis à jour, un AWS IoT message MQTT est envoyé. Vous pouvez créer une règle qui effectue des actions spécifiques lorsque les ombres de périphérique changent en fonction du message MQTT. Dans ce cas, l'objectif est de gérer le message de mise à jour, d'extraire les valeurs des propriétés et de les transmettre aux actifs de votre appareil dans AWS IoT SiteWise.

Pour créer une règle avec une AWS IoT SiteWise action

1. Accédez à la [console AWS IoT](#).
2. Dans le volet de navigation de gauche, choisissez Routage des messages, puis Règles.
3. Choisissez Créer une règle.
4. Entrez un nom et une description pour votre règle, puis choisissez Next.
5. Entrez l'instruction SQL suivante, puis choisissez Next.

```
SELECT
```

```
*  
FROM  
  '$aws/things/+/shadow/update/accepted'  
WHERE  
  startsWith(topic(3), "SiteWiseTutorialDevice")
```

Cette instruction de requête de règle fonctionne, car le service d'ombres de périphérique publie les mises à jour des ombres vers `$aws/things/thingName/shadow/update/accepted`. Pour plus d'informations sur le Device Shadow, voir [Device Shadow Service](#) dans le Guide du AWS IoT développeur.

Dans la clause WHERE, cette instruction de requête de règle utilise la fonction `topic(3)` pour obtenir le nom de l'objet à partir du troisième segment de la rubrique. Ensuite, l'instruction filtre les périphériques dont les noms ne correspondent pas à ceux des périphériques du didacticiel. Pour plus d'informations sur le AWS IoT SQL, consultez la [référence AWS IoT SQL](#) dans le Guide du AWS IoT développeur.

6. Sous Actions de règle, choisissez Envoyer les données des messages aux propriétés des actifs dans AWS IoT SiteWise et procédez comme suit :
 - a. Choisissez By property alias (Par alias de propriété).
 - b. Dans Property alias (Alias de propriété), entrez **`/tutorial/device/${topic(3)}/cpu`**.

La `${...}` syntaxe est un modèle de substitution. AWS IoT évalue le contenu à l'intérieur des bretelles. Ce modèle de substitution extrait le nom de l'objet à partir de la rubrique pour créer un alias spécifique à chaque objet. Pour plus d'informations, consultez la section [Modèles de substitution](#) dans le Guide du AWS IoT développeur.

 Note

Comme une expression dans un modèle de substitution est évaluée séparément de l'instruction SELECT, vous ne pouvez pas utiliser un modèle de substitution pour référencer un alias créé à l'aide d'une clause AS. Vous pouvez référencer uniquement les informations présentes dans la charge utile d'origine, en plus des fonctions et opérateurs pris en charge.

- c. Dans Numéro d'entrée - facultatif, entrez **`${concat(topic(3), "-cpu-", floor(state.reported.timestamp))}`**.

L'entrée IDs identifie de manière unique chaque tentative de saisie de valeur. Si une entrée renvoie une erreur, vous pouvez trouver son ID dans l'erreur générée pour résoudre le problème. Le modèle de substitution de cet ID d'entrée combine le nom de l'objet et l'horodatage signalé du périphérique. Par exemple, l'ID d'entrée généré peut ressembler à `SiteWiseTutorialDevice1-cpu-1579808494`.

- d. Dans Time in seconds (Délai en secondes), entrez **`${floor(state.reported.timestamp)}`**.

Ce modèle de substitution calcule le délai en secondes à compter de l'horodatage signalé du périphérique. Dans ce didacticiel, les périphériques signalent l'horodatage en secondes au format d'heure Unix epoch sous la forme d'un nombre à virgule flottante.

- e. Dans Décalage en nanos - facultatif, entrez **`${floor((state.reported.timestamp % 1) * 1E9)}`**.

Ce modèle de substitution calcule le décalage en nanoseconde à compter du délai en secondes en convertissant la partie décimale de l'horodatage signalé du périphérique.

Note

AWS IoT SiteWise nécessite que vos données aient un horodatage actuel en temps réel à l'époque Unix. Si vos périphériques n'indiquent pas l'heure avec précision, vous pouvez obtenir l'heure actuelle à partir du moteur de règles AWS IoT avec [timestamp\(\)](#). Cette fonction signale le délai en millisecondes. Vous devez donc remplacer les paramètres de délai de l'action de règle par les valeurs suivantes :

- Dans Time in seconds (Délai en secondes), entrez **`${floor(timestamp() / 1E3)}`**.
- Dans Offset in nanos (Décalage en nanosecondes), entrez **`${(timestamp() % 1E3) * 1E6}`**.

- f. Dans Data type (Type de données), choisissez Double.

Ce type de données doit correspondre au type de données de la propriété de ressource que vous avez définie dans le modèle de ressource.

- g. Dans Value (Valeur), entrez **`${state.reported.cpu}`**. Dans les modèles de substitution, vous utilisez l'opérateur `.` pour récupérer une valeur à partir d'une structure JSON.

- h. Choisissez Add entry (Ajouter une entrée) pour ajouter une entrée pour la propriété d'utilisation de la mémoire et suivez à nouveau les étapes ci-dessous pour cette propriété :
 - i. Choisissez By property alias (Par alias de propriété).
 - ii. Dans Property alias (Alias de propriété), entrez **`/tutorial/device/${topic(3)}/memory`**.
 - iii. Dans Numéro d'entrée - facultatif, entrez **`${concat(topic(3), "-memory-", floor(state.reported.timestamp))}`**.
 - iv. Dans Time in seconds (Délai en secondes), entrez **`${floor(state.reported.timestamp)}`**.
 - v. Dans Décalage en nanos - facultatif, entrez **`${floor((state.reported.timestamp % 1) * 1E9)}`**.
 - vi. Dans Data type (Type de données), choisissez Double.
 - vii. Dans Value (Valeur), entrez **`${state.reported.memory}`**.
 - i. Sous Rôle IAM, choisissez Créer un nouveau rôle pour créer un rôle IAM pour cette action de règle. Ce rôle permet AWS IoT de transférer les données vers les propriétés de votre parc d'appareils et de sa hiérarchie d'actifs.
 - j. Entrez un nom de rôle et choisissez Create.
7. (Facultatif) Configurez une action d'erreur que vous pouvez utiliser pour dépanner la règle. Pour de plus amples informations, veuillez consulter [Résoudre les problèmes liés à une règle \(\)AWS IoT SiteWise](#).
 8. Choisissez Suivant.
 9. Vérifiez les paramètres et choisissez Créer pour créer la règle.

Étape 8 : Exécuter le script client de l'appareil

Dans le cadre de ce didacticiel, vous n'utilisez pas un appareil réel pour communiquer des données. Au lieu de cela, vous exécutez un script pour mettre à jour l'ombre AWS IoT de l'appareil en fonction de l'utilisation du processeur et de la mémoire afin d'imiter les données réelles des capteurs. Pour exécuter le script, vous devez d'abord installer les Python packages requis. Dans cette procédure, vous installez les Python packages requis, puis vous exécutez le script client de l'appareil.

Pour configurer et exécuter le script client de périphérique

1. Accédez à la [console AWS IoT](#).

2. En bas du volet de navigation de gauche, choisissez Settings (Paramètres).
3. Enregistrez le point de terminaison personnalisé pour l'utiliser avec le script client de périphérique. Vous utiliserez ce point de terminaison pour interagir avec les ombres de votre objet. Ce point de terminaison est unique à votre compte dans la région actuelle.

Le point de terminaison personnalisé devrait ressembler à l'exemple suivant.

```
identifier.iot.region.amazonaws.com
```

4. Ouvrez une ligne de commande et exécutez la commande suivante pour accéder au répertoire du didacticiel que vous avez créé précédemment.

```
cd iot-sitewise-rule-tutorial
```

5. Exécutez la commande suivante pour installer le kit Kit SDK des appareils AWS IoT pour Python.

```
pip3 install AWSIoTPythonSDK
```

Pour plus d'informations, consultez [Kit SDK des appareils AWS IoT pour Python](#) le guide du AWS IoT développeur

6. Exécutez la commande suivante pour installer psutil, bibliothèque d'utilitaires système et de processus multiplateformes.

```
pip3 install psutil
```

Pour plus d'informations, consultez [psutil](#) dans l'index des packages Python.

7. Créez un fichier appelé `thing_performance.py` dans le répertoire `iot-sitewise-rule-tutorial`, puis copiez le code Python suivant dans le fichier.

```
import AWSIoTPythonSDK.MQTTLib as AWSIoTPyMQTT

import json
import psutil
import argparse
import logging
import time

# Configures the argument parser for this program.
```

```
def configureParser():
    parser = argparse.ArgumentParser()
    parser.add_argument(
        "-e",
        "--endpoint",
        action="store",
        required=True,
        dest="host",
        help="Your AWS IoT custom endpoint",
    )
    parser.add_argument(
        "-r",
        "--rootCA",
        action="store",
        required=True,
        dest="rootCAPath",
        help="Root CA file path",
    )
    parser.add_argument(
        "-c",
        "--cert",
        action="store",
        required=True,
        dest="certificatePath",
        help="Certificate file path",
    )
    parser.add_argument(
        "-k",
        "--key",
        action="store",
        required=True,
        dest="privateKeyPath",
        help="Private key file path",
    )
    parser.add_argument(
        "-p",
        "--port",
        action="store",
        dest="port",
        type=int,
        default=8883,
        help="Port number override",
    )
    parser.add_argument(
```

```

        "-n",
        "--thingName",
        action="store",
        required=True,
        dest="thingName",
        help="Targeted thing name",
    )
    parser.add_argument(
        "-d",
        "--requestDelay",
        action="store",
        dest="requestDelay",
        type=float,
        default=1,
        help="Time between requests (in seconds)",
    )
    parser.add_argument(
        "-v",
        "--enableLogging",
        action="store_true",
        dest="enableLogging",
        help="Enable logging for the AWS IoT Device SDK for Python",
    )
    return parser

# An MQTT shadow client that uploads device performance data to AWS IoT at a
# regular interval.
class PerformanceShadowClient:
    def __init__(
        self,
        thingName,
        host,
        port,
        rootCAPath,
        privateKeyPath,
        certificatePath,
        requestDelay,
    ):
        self.thingName = thingName
        self.host = host
        self.port = port
        self.rootCAPath = rootCAPath
        self.privateKeyPath = privateKeyPath

```

```

        self.certificatePath = certificatePath
        self.requestDelay = requestDelay

    # Updates this thing's shadow with system performance data at a regular
    interval.
    def run(self):
        print("Connecting MQTT client for {}".format(self.thingName))
        mqttClient = self.configureMQTTClient()
        mqttClient.connect()
        print("MQTT client for {} connected".format(self.thingName))
        deviceShadowHandler = mqttClient.createShadowHandlerWithName(
            self.thingName, True
        )

        print("Running performance shadow client for {}...
\n".format(self.thingName))
        while True:
            performance = self.readPerformance()
            print("{}".format(self.thingName))
            print("CPU:\t{}".format(performance["cpu"]))
            print("Memory:\t{}\n".format(performance["memory"]))
            payload = {"state": {"reported": performance}}
            deviceShadowHandler.shadowUpdate(
                json.dumps(payload), self.shadowUpdateCallback, 5
            )
            time.sleep(args.requestDelay)

    # Configures the MQTT shadow client for this thing.
    def configureMQTTClient(self):
        mqttClient = AWSIoTPyMQTT.AWSIoTMQTTShadowClient(self.thingName)
        mqttClient.configureEndpoint(self.host, self.port)
        mqttClient.configureCredentials(
            self.rootCAPath, self.privateKeyPath, self.certificatePath
        )
        mqttClient.configureAutoReconnectBackoffTime(1, 32, 20)
        mqttClient.configureConnectDisconnectTimeout(10)
        mqttClient.configureMQTTOperationTimeout(5)
        return mqttClient

    # Returns the local device's CPU usage, memory usage, and timestamp.
    def readPerformance(self):
        cpu = psutil.cpu_percent()
        memory = psutil.virtual_memory().percent
        timestamp = time.time()

```

```

        return {"cpu": cpu, "memory": memory, "timestamp": timestamp}

# Prints the result of a shadow update call.
def shadowUpdateCallback(self, payload, responseStatus, token):
    print("{} {}".format(self.thingName))
    print("Update request {} {} \n".format(token, responseStatus))

# Configures debug logging for the AWS IoT Device SDK for Python.
def configureLogging():
    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter(
        "%(asctime)s - %(name)s - %(levelname)s - %(message)s"
    )
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)

# Runs the performance shadow client with user arguments.
if __name__ == "__main__":
    parser = configureParser()
    args = parser.parse_args()
    if args.enableLogging:
        configureLogging()
    thingClient = PerformanceShadowClient(
        args.thingName,
        args.host,
        args.port,
        args.rootCAPath,
        args.privateKeyPath,
        args.certificatePath,
        args.requestDelay,
    )
    thingClient.run()

```

8. Exécutez `thing_performance.py` depuis la ligne de commande avec les paramètres suivants :

- `-n, --thingName` — Le nom de votre objet, tel que **SiteWiseTutorialDevice1**.
- `-e, --endpoint` — Votre point de AWS IoT terminaison personnalisé que vous avez enregistré plus tôt dans cette procédure.

- `-r, --rootCA` — Le chemin d'accès à votre certificat CA AWS IoT racine.
- `-c, --cert` — Le chemin d'accès à votre certificat d' AWS IoT objet.
- `-k, --key` — Le chemin d'accès à la clé privée de votre certificat d' AWS IoT objet.
- `-d, --requestDelay` — (Facultatif) Temps d'attente en secondes entre chaque mise à jour instantanée de l'appareil. Par défaut, il correspond à 1 seconde.
- `-v, --enableLogging` — (Facultatif) Si ce paramètre est présent, le script imprime des messages de débogage à partir du Kit SDK des appareils AWS IoT pour Python.

La commande doit être similaire à l'exemple suivant :

```
python3 thing_performance.py \  
  --thingName SiteWiseTutorialDevice1 \  
  --endpoint identifiant.iot.region.amazonaws.com \  
  --rootCA AmazonRootCA1.pem \  
  --cert device1/thing-id-certificate.pem.crt \  
  --key device1/thing-id-private.pem.key
```

Si vous exécutez le script pour d'autres AWS IoT éléments, mettez à jour le nom de l'objet et le répertoire des certificats en conséquence.

9. Essayez d'ouvrir et de fermer des programmes sur votre appareil pour voir comment évolue l'utilisation de l'UC et de la mémoire. Le script imprime chaque lecture de l'utilisation de l'UC et de la mémoire. Si le script charge des données vers le service d'ombres de périphérique avec succès, la sortie du script doit ressembler à l'exemple suivant.

```
[SiteWiseTutorialDevice1]  
CPU:    24.6%  
Memory: 85.2%  
  
[SiteWiseTutorialDevice1]  
Update request e6686e44-fca0-44db-aa48-3ca81726f3e3 accepted
```

10. Procédez comme suit pour vérifier que le script met à jour l'ombre du périphérique :
 - a. Accédez à la [console AWS IoT](#).
 - b. Dans le volet de navigation de gauche, choisissez Tous les appareils, puis Objets.
 - c. Choisissez votre truc, SiteWiseTutorialDevice.

- d. Choisissez l'onglet Device Shadows, choisissez Classic Shadow et vérifiez que l'état Shadow ressemble à l'exemple suivant.

```
{
  "reported": {
    "cpu": 24.6,
    "memory": 85.2,
    "timestamp": 1579567542.2835066
  }
}
```

Si l'état fantôme de votre objet est vide ou ne ressemble pas à celui de l'exemple précédent, vérifiez que le script est en cours d'exécution et que vous y êtes connecté correctement AWS IoT. Si le script continue à expirer lors de la connexion à AWS IoT, vérifiez que votre [politique d'objets](#) est configurée conformément à ce didacticiel.

11. Procédez comme suit pour vérifier que l'action de règle envoie des données à AWS IoT SiteWise :

- Accédez à la [console AWS IoT SiteWise](#).
- Dans le panneau de navigation de gauche, choisissez Assets (Ressources).
- Sélectionnez la flèche en regard de votre parc de périphériques (SiteWise Tutorial Device Fleet 1 1) pour développer sa hiérarchie de ressources, puis choisissez votre ressource de périphérique (SiteWise Tutorial Device 1).
- Choisissez Measurements (Mesures).
- Vérifiez que les cellules Latest value (Valeur la plus récente) ont des valeurs pour les propriétés CPU Usage et Memory Usage.

Measurements				
Name	Alias	Notification status	Notification topic	Latest value
CPU Usage	/tutorial/device/SiteWiseTutorialDevice1/cpu	⊖ Disabled	–	24.6
Memory Usage	/tutorial/device/SiteWiseTutorialDevice1/memory	⊖ Disabled	–	85.2

- Si les propriétés CPU Usage et Memory Usage n'ont pas les dernières valeurs, actualisez la page. Si aucune ligne n'apparaît après quelques minutes, consultez [Résoudre les problèmes liés à une règle \(AWS IoT SiteWise\)](#).

Vous avez terminé ce didacticiel. Si vous souhaitez explorer les visualisations en direct de vos données, vous pouvez configurer un portail dans AWS IoT SiteWise Monitor. Pour de plus amples informations, veuillez consulter [Surveillez les données avec AWS IoT SiteWise Monitor](#). Sinon, vous pouvez appuyer sur CTRL+C dans votre invite de commandes pour arrêter le script client du périphérique. Il est peu probable que le programme Python envoie suffisamment de messages pour générer des frais, mais il est recommandé d'arrêter le programme une fois que vous avez terminé.

Étape 9 : Nettoyer les ressources après le didacticiel

Après avoir terminé le didacticiel sur l'ingestion de données provenant d' AWS IoT objets, nettoyez vos ressources pour éviter d'encourir des frais supplémentaires.

Pour supprimer des actifs hiérarchiques dans AWS IoT SiteWise

1. Accédez à la [AWS IoT SiteWise console](#)
2. Dans le panneau de navigation de gauche, choisissez Assets (Ressources).
3. Lorsque vous supprimez des actifs AWS IoT SiteWise, vous devez d'abord les dissocier.

Procédez comme suit pour dissocier les ressources de périphérique de votre parc de périphériques :

- a. Choisissez l'actif de votre parc d'appareils (SiteWise Tutorial Device Fleet 1).
- b. Choisissez Modifier.
- c. Sous Assets associated to this asset (Ressources associées à cette ressource), choisissez Disassociate (Dissocier) pour chaque ressource de périphérique associée à cette ressource de parc de périphériques.
- d. Choisissez Enregistrer.

Désormais, les ressources de votre périphérique ne devraient plus être organisées sous forme de hiérarchie.

4. Choisissez votre ressource de périphérique (SiteWise Tutorial Device 1).
5. Sélectionnez Delete (Supprimer).
6. Dans le champ de confirmation, entrez, **Delete**, puis choisissez Delete (Supprimer).
7. Répétez les étapes 4 à 6 pour chaque actif de l'appareil et pour le parc d'appareils (SiteWise Tutorial Device Fleet 1).

Pour supprimer des modèles d'actifs hiérarchiques dans AWS IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).
2. Si vous ne l'avez pas déjà fait, supprimez vos périphériques et les ressources de votre parc de périphériques. Pour plus d'informations, consultez la [procédure précédente](#). Vous ne pouvez pas supprimer un modèle s'il existe des ressources créées à partir de ce modèle.
3. Dans le volet de navigation de gauche, choisissez Models (Modèles).
4. Choisissez votre modèle de ressource de parc de périphériques (SiteWise Tutorial Device Fleet Model).

Lorsque vous supprimez des modèles d'actifs hiérarchiques, commencez par supprimer d'abord le modèle d'actif parent.

5. Sélectionnez Delete (Supprimer).
6. Dans le champ de confirmation, entrez, **Delete**, puis choisissez Delete (Supprimer).
7. Répétez les étapes 4 à 6 pour le modèle de ressource de périphérique (SiteWise Tutorial Device Model Model).

Pour désactiver ou supprimer une règle dans AWS IoT Core

1. Accédez à la [console AWS IoT](#).
2. Dans le volet de navigation de gauche, choisissez Routage des messages, puis Règles.
3. Sélectionnez votre règle, puis cliquez sur Supprimer.
4. Dans la boîte de dialogue de confirmation, entrez le nom de la règle, puis choisissez Supprimer.

Visualisez et partagez les données des parcs éoliens dans SiteWise Monitor

Ce didacticiel explique comment visualiser et partager des données industrielles par le biais d'applications Web gérées, appelées portails. AWS IoT SiteWise Monitor Chaque portail englobe des projets, ce qui vous donne la possibilité de choisir les données accessibles au sein de chaque projet. Spécifiez ensuite les personnes de votre organisation qui peuvent accéder à chaque portail. Vos utilisateurs se connectent aux portails à l'aide de AWS IAM Identity Center comptes, afin que vous puissiez utiliser votre magasin d'identités existant ou un magasin géré par AWS.

Vous, ainsi que vos utilisateurs disposant d'autorisations suffisantes, pouvez créer des tableaux de bord dans chaque projet pour visualiser vos données industrielles de manière significative. Ensuite, vos utilisateurs peuvent consulter ces tableaux de bord pour obtenir rapidement des informations sur vos données et surveiller votre fonctionnement. Vous pouvez configurer des autorisations administratives ou en lecture seule pour chaque projet pour chaque utilisateur de votre entreprise. Pour de plus amples informations, veuillez consulter [Surveillez les données avec AWS IoT SiteWise Monitor](#).

Tout au long du didacticiel, vous améliorez la AWS IoT SiteWise démo en fournissant un exemple de jeu de données pour un parc éolien. Vous configurez un portail dans SiteWise Monitor, vous créez un projet et des tableaux de bord pour visualiser les données du parc éolien. Le didacticiel couvre également la création d'utilisateurs supplémentaires, ainsi que l'attribution d'autorisations pour posséder ou consulter le projet et ses tableaux de bord associés.

Note

Lorsque vous utilisez SiteWise Monitor, vous êtes facturé par utilisateur qui se connecte à un portail (par mois). Dans ce didacticiel, vous créez trois utilisateurs, mais vous ne devez vous connecter qu'avec un seul utilisateur. Après avoir terminé ce didacticiel, vous encourez des frais pour un utilisateur. Pour plus d'informations, consultez [AWS IoT SiteWise Pricing](#) (Tarification CTlong).

Rubriques

- [Prérequis](#)
- [Étape 1 : créer un portail dans SiteWise Monitor](#)
- [Étape 2 : Connectez-vous à un portail](#)
- [Étape 3 : Création d'un projet de parc éolien](#)
- [Étape 4 : Création d'un tableau de bord pour visualiser les données du parc éolien](#)
- [Étape 5 : Explorez le portail](#)
- [Étape 6 : Nettoyer les ressources après le didacticiel](#)

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des éléments suivants :

- Un Compte AWS. Si vous n'en avez pas, veuillez consulter [Configurez un AWS compte](#).
- Un ordinateur de développement en cours d'exécution Windows, macOS, Linux, ou Unix pour accéder au AWS Management Console. Pour plus d'informations, consultez [Qu'est-ce qu' AWS Management Console ?](#)
- Un utilisateur AWS Identity and Access Management (IAM) doté d'autorisations d'administrateur.
- Une démonstration de AWS IoT SiteWise parc éolien en cours d'exécution. Lorsque vous configurez la démo, elle définit les modèles et les actifs AWS IoT SiteWise et leur transmet des données pour représenter un parc éolien. Pour de plus amples informations, veuillez consulter [Utiliser la AWS IoT SiteWise démo](#).
- Si vous avez activé IAM Identity Center dans votre compte, connectez-vous à votre compte AWS Organizations de gestion. Pour plus d'informations, veuillez consulter la rubrique [Terminologie et concepts AWS Organizations](#). Si vous n'avez pas activé IAM Identity Center, vous allez l'activer dans ce didacticiel et définir votre compte comme compte de gestion.

Si vous ne parvenez pas à vous connecter à votre compte de AWS Organizations gestion, vous pouvez suivre partiellement le didacticiel tant que votre organisation compte un utilisateur IAM Identity Center. Dans ce cas, vous pouvez créer le portail et les tableaux de bord, mais vous ne pouvez pas créer de nouveaux utilisateurs IAM Identity Center à affecter à des projets.

Étape 1 : créer un portail dans SiteWise Monitor

Dans cette procédure, vous créez un portail dans AWS IoT SiteWise Monitor. Chaque portail est une application Web gérée à laquelle vous et vos utilisateurs pouvez vous connecter à l'aide de AWS IAM Identity Center comptes. Avec IAM Identity Center, vous pouvez utiliser le magasin d'identités existant de votre entreprise ou en créer un géré par AWS. Les employés de votre entreprise peuvent se connecter sans créer de compte distinct Comptes AWS.

Pour créer un portail

1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Passez en revue les [AWS IoT SiteWise points de terminaison et les quotas](#) pris en charge et changez de région, si nécessaire. AWS IoT SiteWise Vous devez exécuter la AWS IoT SiteWise démo dans la même région.
3. Dans le volet de navigation de gauche, choisissez Portals (Portails).
4. Choisissez Créer un portail.

5. Si vous avez déjà activé IAM Identity Center, passez à l'étape 6. Dans le cas contraire, procédez comme suit pour activer IAM Identity Center :
 - a. Sur la page Activer AWS IAM Identity Center (SSO), entrez votre adresse e-mail, votre prénom et votre nom de famille pour créer un utilisateur IAM Identity Center en tant qu'administrateur du portail. Utilisez une adresse e-mail à laquelle vous pouvez accéder afin de recevoir un e-mail afin de définir un mot de passe pour votre nouvel utilisateur IAM Identity Center.

Dans un portail, l'administrateur du portail crée des projets et affecte des utilisateurs à des projets. Vous pouvez créer plus d'utilisateurs ultérieurement.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Enable SSO

Step 2
Portal configuration

Step 3
Invite administrators

Step 4
Assign users

Enable AWS Single Sign-On (SSO)

AWS IoT SiteWise Monitor requires SSO to create a portal and invite users. Create your first user below to enable AWS Single-Sign On. Later in this process, you'll have the opportunity to create other users by using the AWS SSO console. [Learn more](#)

Create a user

Email address

First name

Last name

Upon creation this application will enable AWS Organizations and Single Sign-On. [Learn more](#)

Cancel **Create user**

- b. Choisissez Create user (Créer un utilisateur).
6. Sur la page Portal configuration (Configuration du portail), procédez comme suit :
 - a. Entrez un nom pour votre portail, tel que **WindFarmPortal**.
 - b. (Facultatif) Saisissez une description pour votre portail. Si vous avez plusieurs portails, utilisez des descriptions significatives pour vous aider à suivre ce que contient chaque portail.
 - c. (Facultatif) Téléchargez une image à afficher sur le portail.
 - d. Entrez une adresse e-mail que les utilisateurs du portail peuvent contacter en cas de problème avec le portail et qu'ils ont besoin de l'aide de l' AWS administrateur de votre entreprise pour le résoudre.

- e. Choisissez Créer un portail.
7. Sur la page Inviter des administrateurs, vous pouvez affecter des utilisateurs d'IAM Identity Center au portail en tant qu'administrateurs. Les administrateurs du portail gèrent les autorisations et les projets au sein d'un portail. Sur cette page, procédez comme suit :
- a. Sélectionnez un utilisateur comme administrateur du portail. Si vous avez activé IAM Identity Center plus tôt dans ce didacticiel, sélectionnez l'utilisateur que vous avez créé.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Invite administrators

Select the users that you want to be portal administrators. When invited, portal administrators control users' access to the operational data of your Sitewise assets. [Learn more](#)

Send invite to selected users

Users (1) Create user

Find resources

	Display name	Email
☒	John Doe	john.doe@example.com

Selected users (1)

Cancel **Next**

- b. (Facultatif) Choisissez Envoyer l'invitation aux utilisateurs sélectionnés. Votre client de messagerie s'ouvre et une invitation apparaît dans le corps du message. Vous pouvez personnaliser l'e-mail avant de l'envoyer aux administrateurs de votre portail. Vous pouvez également envoyer l'e-mail aux administrateurs de votre portail ultérieurement. Si vous essayez SiteWise Monitor pour la première fois et que vous êtes l'administrateur du portail, vous n'avez pas besoin de vous envoyer un e-mail.
 - c. Choisissez Suivant.
8. Sur la page Attribuer des utilisateurs, vous pouvez attribuer des utilisateurs d'IAM Identity Center au portail. Les administrateurs du portail peuvent ultérieurement désigner ces utilisateurs en tant que propriétaires ou spectateurs du projet. Les propriétaires de projets peuvent créer des tableaux de bord dans les projets. Les visionneurs de projets ont un accès en lecture seule aux projets qui leur sont assignés. Sur cette page, vous pouvez créer des utilisateurs IAM Identity Center à ajouter au portail.

Note

Si vous n'êtes pas connecté à votre compte AWS Organizations de gestion, vous ne pouvez pas créer d'utilisateurs IAM Identity Center. Choisissez Affecter des utilisateurs pour créer le portail sans utilisateurs du portail, puis ignorez cette étape.

Sur cette page, procédez comme suit :

- a. Effectuez deux fois les étapes suivantes pour créer deux utilisateurs IAM Identity Center :
 - i. Choisissez Créer un utilisateur pour ouvrir une boîte de dialogue dans laquelle vous entrez les informations relatives au nouvel utilisateur.
 - ii. Entrez l'adresse e-mail, le prénom et le nom de famille du nouvel utilisateur. IAM Identity Center envoie un e-mail à l'utilisateur pour qu'il définisse son mot de passe. Si vous souhaitez vous connecter au portail en tant que ces utilisateurs, choisissez une adresse e-mail à laquelle vous pouvez accéder. Chaque adresse e-mail doit être unique. Vos utilisateurs se connectent au portail en utilisant leur adresse e-mail comme nom d'utilisateur.

Create user [X]

Create a new AWS user. You can assign this user access to AWS applications and services

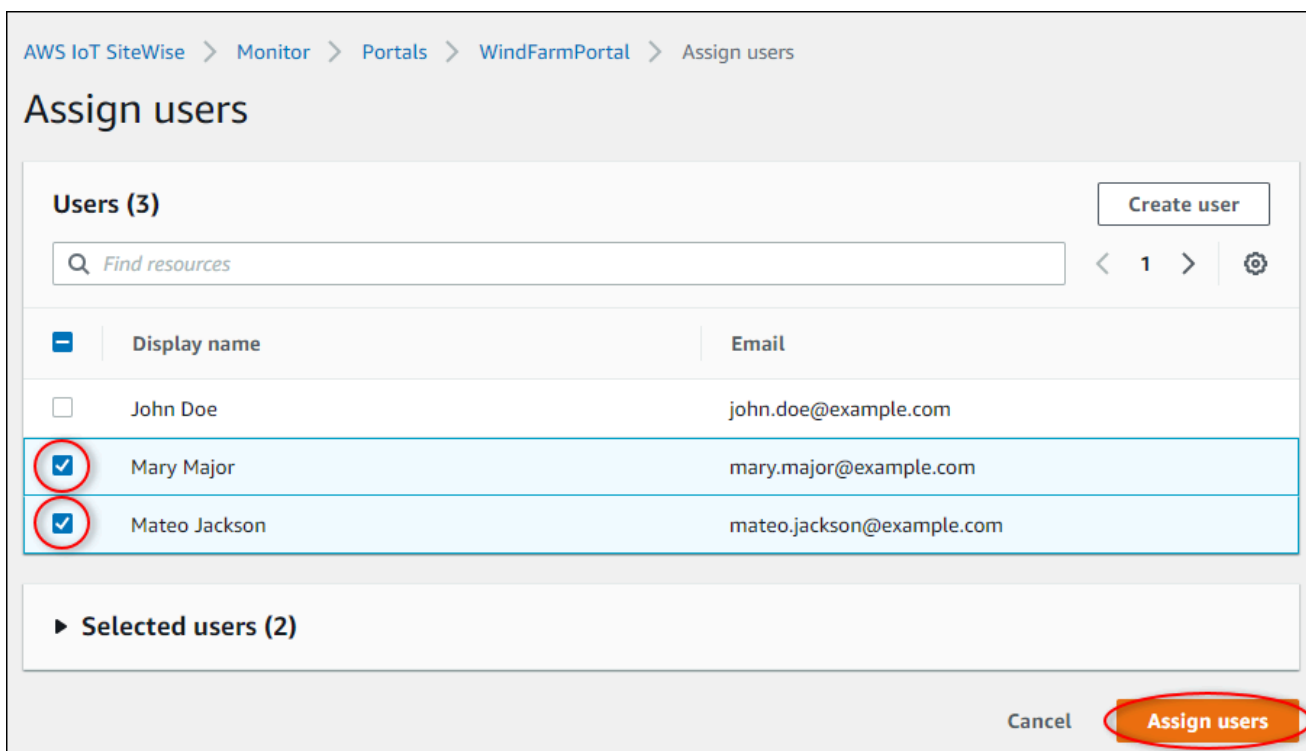
Email address
mary.major@example.com

First name
Mary

Last name
Major

Cancel Create user

- iii. Choisissez Create user (Créer un utilisateur).
- b. Sélectionnez les deux utilisateurs IAM Identity Center que vous avez créés à l'étape précédente.



- c. Choisissez Attribuer des utilisateurs pour ajouter ces utilisateurs au portail.

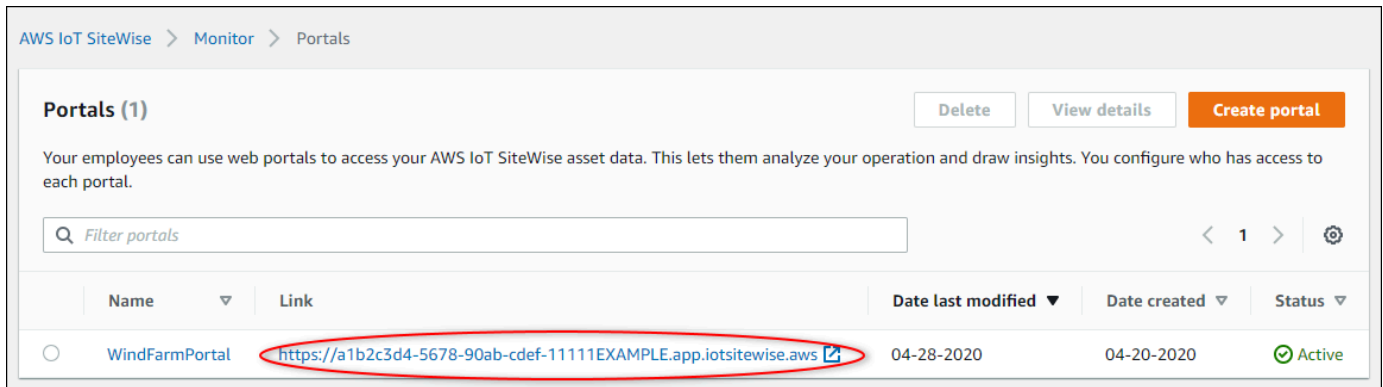
La page des portails s'ouvre avec votre nouveau portail répertorié.

Étape 2 : Connectez-vous à un portail

Dans cette procédure, vous vous connectez à votre nouveau portail en utilisant l' AWS IAM Identity Center utilisateur que vous avez ajouté au portail.

Pour vous connecter à un portail

1. Sur la page Portails choisissez le lien de votre nouveau portail pour ouvrir votre portail dans un nouvel onglet.



2. Si vous avez créé votre premier utilisateur IAM Identity Center plus tôt dans le didacticiel, suivez les étapes ci-dessous pour créer un mot de passe pour votre utilisateur :
 - a. Vérifiez la ligne d'objet de votre e-mail Invitation to join AWS IAM Identity Center.
 - b. Ouvrez cet e-mail d'invitation et choisissez Accept invitation.
 - c. Dans la nouvelle fenêtre, définissez un mot de passe pour votre utilisateur IAM Identity Center.

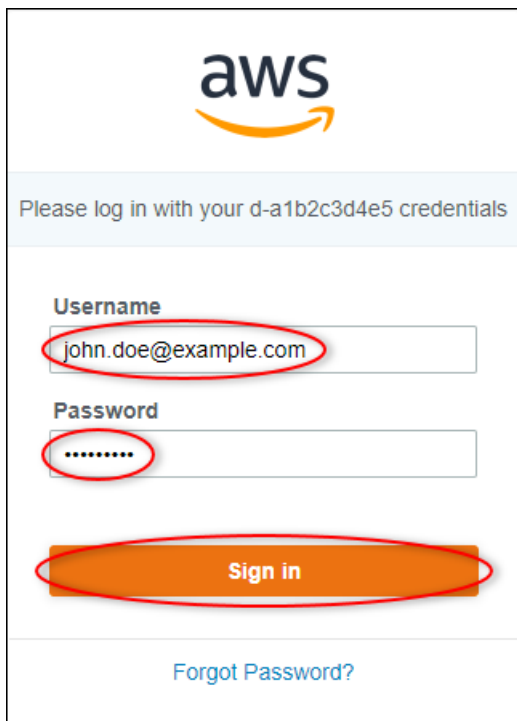
Si vous souhaitez vous connecter ultérieurement au portail en tant que deuxième et troisième utilisateurs d'IAM Identity Center que vous avez créés précédemment, vous pouvez également suivre ces étapes pour définir des mots de passe pour ces utilisateurs.

Note

Si vous n'avez pas reçu d'e-mail, vous pouvez générer un mot de passe pour votre utilisateur dans la console IAM Identity Center. Pour plus d'informations, voir [Réinitialiser le mot de passe utilisateur du centre d'identité IAM pour un utilisateur final dans le guide de l'AWS IAM Identity Center utilisateur](#).

3. Entrez dans votre centre d'identité IAM Username et Password. Si vous avez créé votre utilisateur IAM Identity Center plus tôt dans ce didacticiel, votre Username est l'adresse e-mail de l'administrateur du portail que vous avez créé.

Tous les utilisateurs du portail, y compris l'administrateur du portail, doivent se connecter à l'aide de leurs informations d'identification d'utilisateur IAM Identity Center. Ces informations d'identification ne sont généralement pas les mêmes que celles que vous utilisez pour vous connecter au AWS Management Console.

The image shows the AWS IoT SiteWise login interface. At the top is the AWS logo. Below it, a light blue banner contains the text "Please log in with your d-a1b2c3d4e5 credentials". The main form has two input fields: "Username" with the value "john.doe@example.com" and "Password" with masked characters. Both fields are circled in red. Below the password field is an orange "Sign in" button, also circled in red. At the bottom, there is a blue link labeled "Forgot Password?".

4. Choisissez Sign in.

Votre portail s'ouvre.

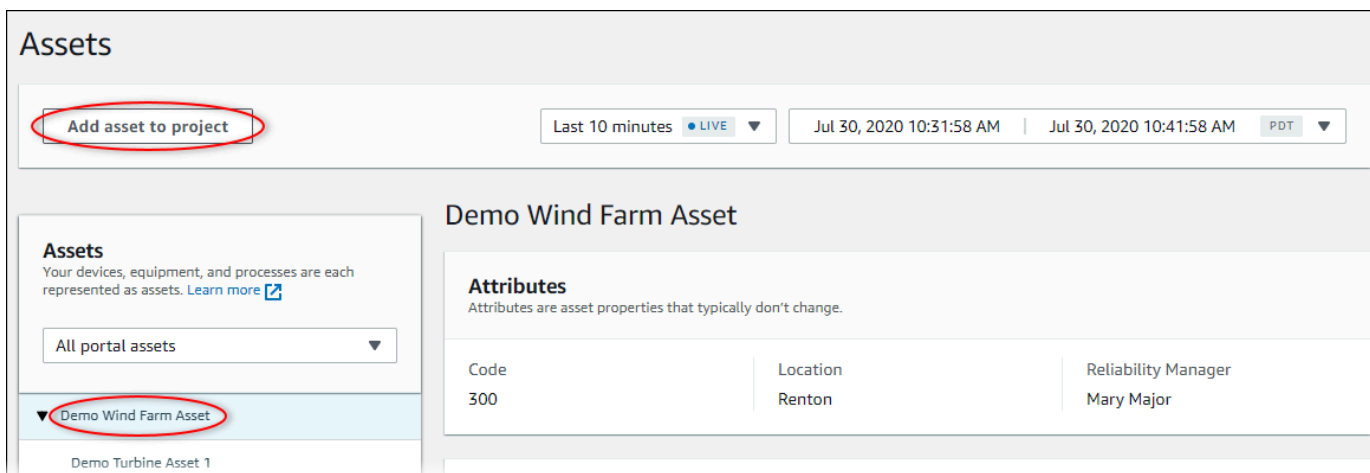
Étape 3 : Création d'un projet de parc éolien

Dans cette procédure, vous créez un projet dans votre portail. Les projets sont des ressources qui définissent un ensemble d'autorisations, d'actifs et de tableaux de bord, que vous pouvez configurer pour visualiser les données des actifs dans ce projet. Avec les projets, vous définissez les personnes ayant accès aux sous-ensembles de votre opération et comment les données de ces sous-ensembles sont visualisées. Vous pouvez attribuer aux utilisateurs du portail les propriétaires ou les visualisateurs de chaque projet. Les propriétaires de projet peuvent créer des tableaux de bord pour visualiser les données et partager le projet avec d'autres utilisateurs. Les utilisateurs du projet peuvent afficher les tableaux de bord, mais pas les modifier. Pour plus d'informations sur les rôles dans SiteWise Monitor, consultez [SiteWise Contrôler les rôles](#).

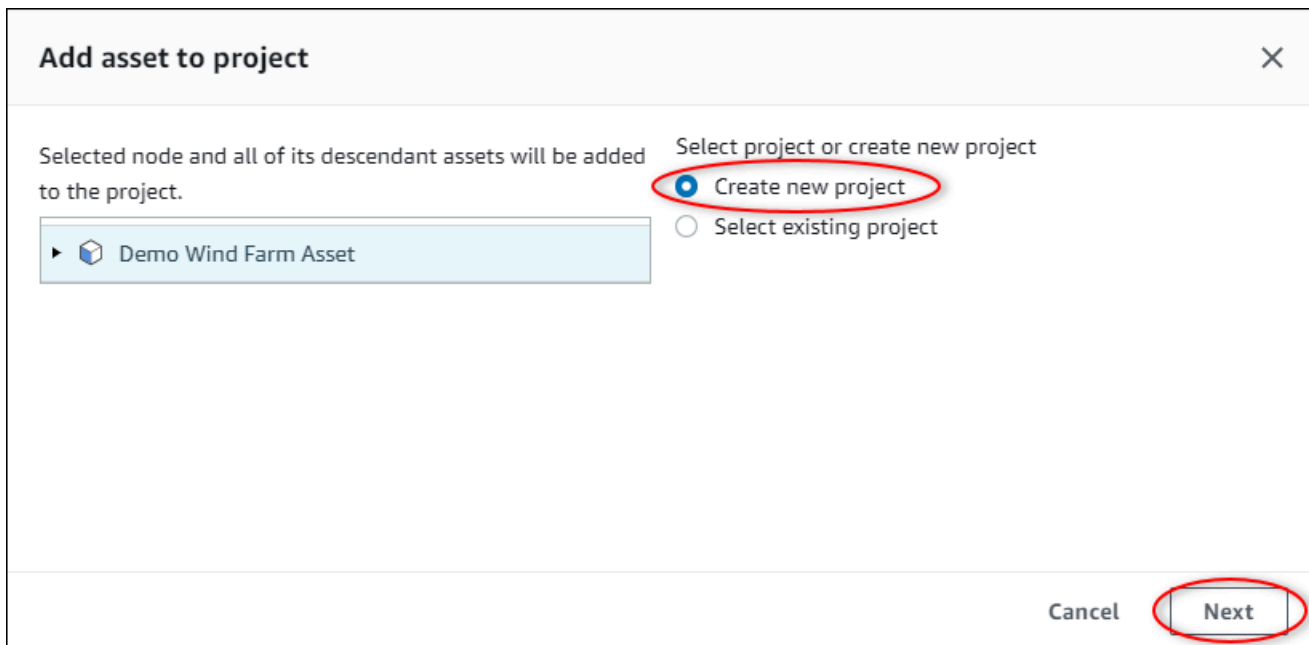
Pour créer un projet de parc éolien

1. Dans le volet de navigation de gauche de votre portail, choisissez l'onglet Ressources. Sur la page Ressources, vous pouvez explorer toutes les ressources disponibles sur le portail et ajouter des ressources aux projets.

2. Dans le navigateur de ressources, choisissez Demo Wind Farm Asset. Lorsque vous choisissez un actif, vous pouvez explorer les données en temps réel et historiques de cet actif. Vous pouvez également appuyer Shift pour sélectionner plusieurs actifs et comparer leurs données side-by-side.
3. Choisissez Ajouter un actif au projet en haut à gauche. Les projets contiennent des tableaux de bord que les utilisateurs de votre portail peuvent consulter pour explorer vos données. Chaque projet a accès à un sous-ensemble de vos actifs dans AWS IoT SiteWise. Lorsque vous ajoutez un actif à un projet, tous les utilisateurs ayant accès à ce projet peuvent également accéder aux données de cet actif et de ses enfants.



4. Dans la boîte de dialogue Ajouter un actif au projet, choisissez Créer un nouveau projet, puis cliquez sur Suivant.



5. Dans la boîte de dialogue Créer un nouveau projet, entrez le nom et la description du projet pour votre projet, puis choisissez Ajouter un actif au projet.

The screenshot shows a 'Create new project' dialog box with a close button (X) in the top right corner. It contains two text input fields: 'Project name' with the value 'Wind Farm 1' and a note 'The project name can have up to 256 characters.'; and 'Project description' with the value 'A project that contains dashboards for wind farm #1.' and a note 'The project description can have up to 2048 characters.' At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Add asset to project' (which is highlighted with an orange border).

La page de votre nouveau projet s'ouvre.

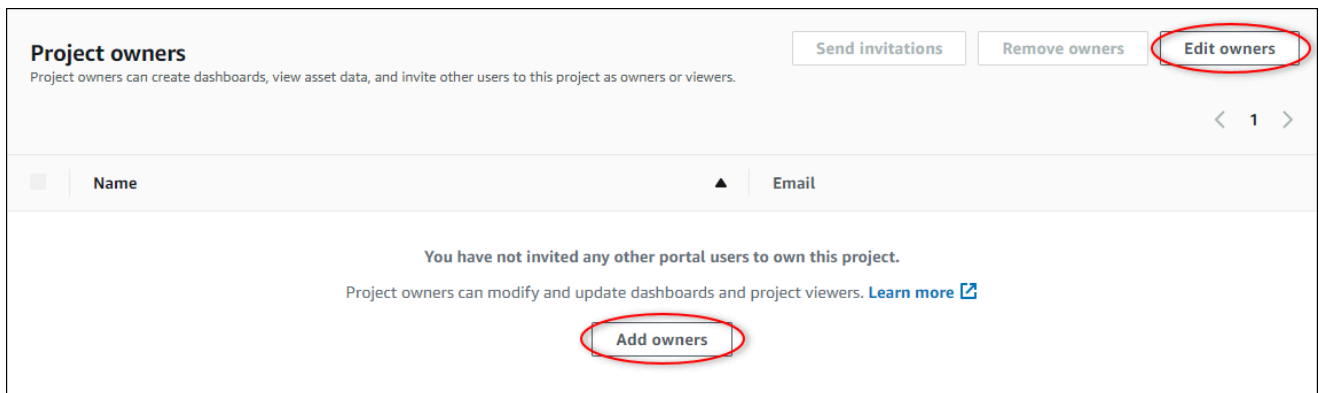
6. Sur la page du projet, vous pouvez ajouter des utilisateurs du portail en tant que propriétaires ou spectateurs de ce projet.

Note

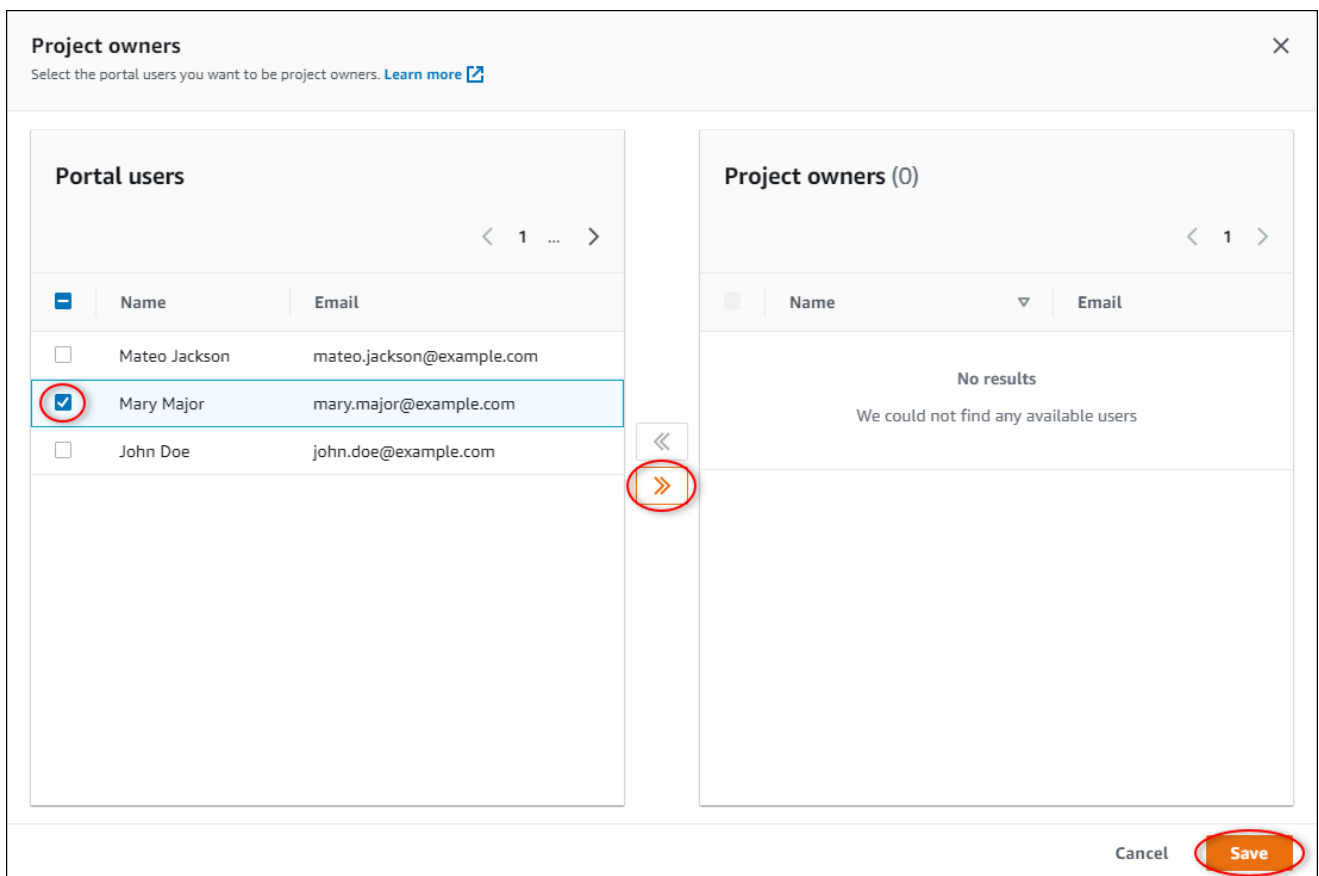
Si vous n'êtes pas connecté à votre compte de AWS Organizations gestion, il se peut que vous n'ayez pas d'utilisateurs du portail à affecter à ce projet. Vous pouvez donc ignorer cette étape.

Sur cette page, procédez comme suit :

- a. Sous Propriétaires du projet, choisissez Ajouter des propriétaires ou Modifier des utilisateurs.



- b. Choisissez l'utilisateur à ajouter en tant que propriétaire du projet (par exemple, Mary Major), puis choisissez l'icône >>.



- c. Choisissez Save (Enregistrer).

Votre utilisateur du IAM Identity Center Mary Major peut se connecter à ce portail pour modifier les tableaux de bord de ce projet et partager ce projet avec d'autres utilisateurs de ce portail.

- d. Sous Visionneuses de projet, choisissez Ajouter des visionneuses ou Modifier des utilisateurs.

- e. Choisissez l'utilisateur à ajouter en tant que visionneur de projet (par exemple, Mateo Jackson), puis choisissez l'icône >>.
- f. Choisissez Save (Enregistrer).

Votre utilisateur du IAM Identity Center Mateo Jackson peut se connecter à ce portail pour consulter, mais pas modifier, les tableaux de bord du projet de parc éolien.

Étape 4 : Création d'un tableau de bord pour visualiser les données du parc éolien

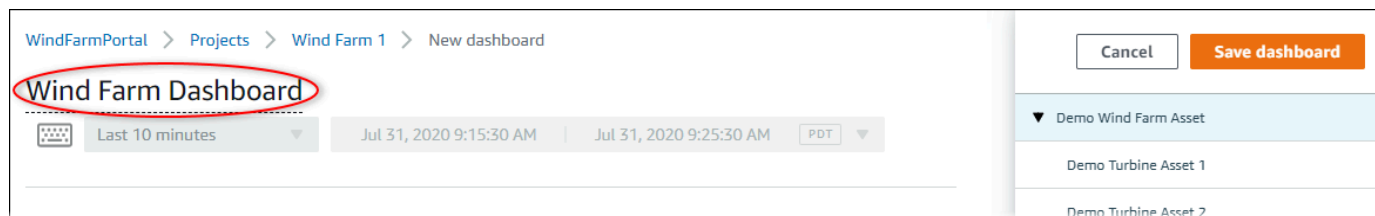
Dans cette procédure, vous créez des tableaux de bord pour visualiser les données du parc éolien de la démonstration. Les tableaux de bord contiennent les visualisations personnalisables des données de ressources de votre projet. Chaque visualisation peut avoir un type différent, tel qu'un graphique linéaire, un graphique à barres ou un affichage d'indicateurs de performance clés (KPI). Vous pouvez choisir le type de visualisation qui convient le mieux à vos données. Les propriétaires de projets peuvent modifier les tableaux de bord, tandis que les personnes qui consultent les projets ne peuvent consulter les tableaux de bord que pour obtenir des informations.

Pour créer un tableau de bord avec des visualisations

1. Sur la page de votre nouveau projet, choisissez Créer un tableau de bord pour créer un tableau de bord et ouvrir sa page de modification.

Dans la page de modification d'un tableau de bord, vous pouvez faire glisser les propriétés des ressources de la hiérarchie des ressources vers le tableau de bord pour créer des visualisations. Vous pouvez ensuite modifier le titre, les titres de légende, le type, la taille et l'emplacement de chaque visualisation dans le tableau de bord.

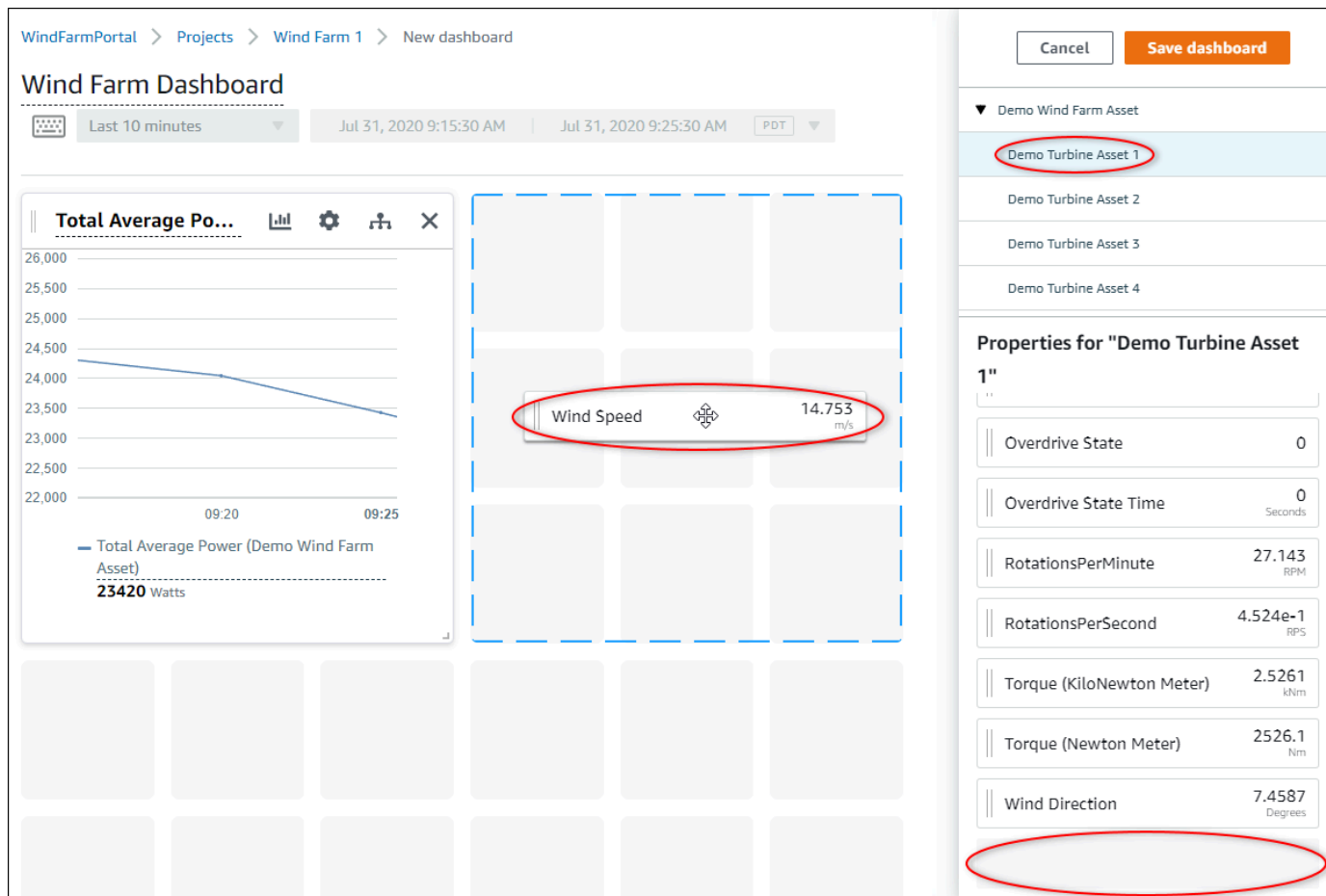
2. Donnez un nom à votre tableau de bord.



3. Faites glisser Total Average Power à partir des Demo Wind Farm Asset au tableau de bord pour créer une visualisation.

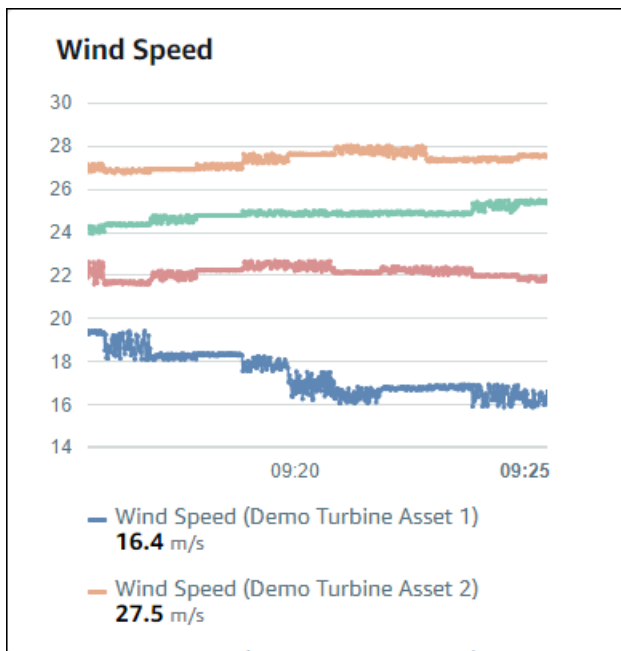
The screenshot displays the 'Wind Farm Dashboard' in the AWS IoT SiteWise console. The breadcrumb navigation at the top reads 'WindFarmPortal > Projects > Wind Farm 1 > New dashboard'. The dashboard title is 'Wind Farm Dashboard'. Below the title, there are filters for 'Last 10 minutes', a date range from 'Jul 31, 2020 9:15:30 AM' to 'Jul 31, 2020 9:25:30 AM', and a 'PDT' time zone selector. The main area is a grid of 18 placeholder widgets. One widget, 'Total Average Power', is highlighted with a red oval and shows a value of '24038 Watts'. On the right, a sidebar titled 'Demo Wind Farm Asset' lists four assets: 'Demo Turbine Asset 1', 'Demo Turbine Asset 2', 'Demo Turbine Asset 3', and 'Demo Turbine Asset 4'. Below this, the 'Properties for "Demo Wind Farm Asset"' section shows 'Code' as '300' and 'Total Overdrive State Time' as '0 seconds'. A red oval highlights the empty space below the 'Code' property.

4. Choisissez Demo Turbine Asset 1 pour afficher les propriétés de cet actif, puis faites glisser Wind Speed au tableau de bord pour créer une visualisation de la vitesse du vent.

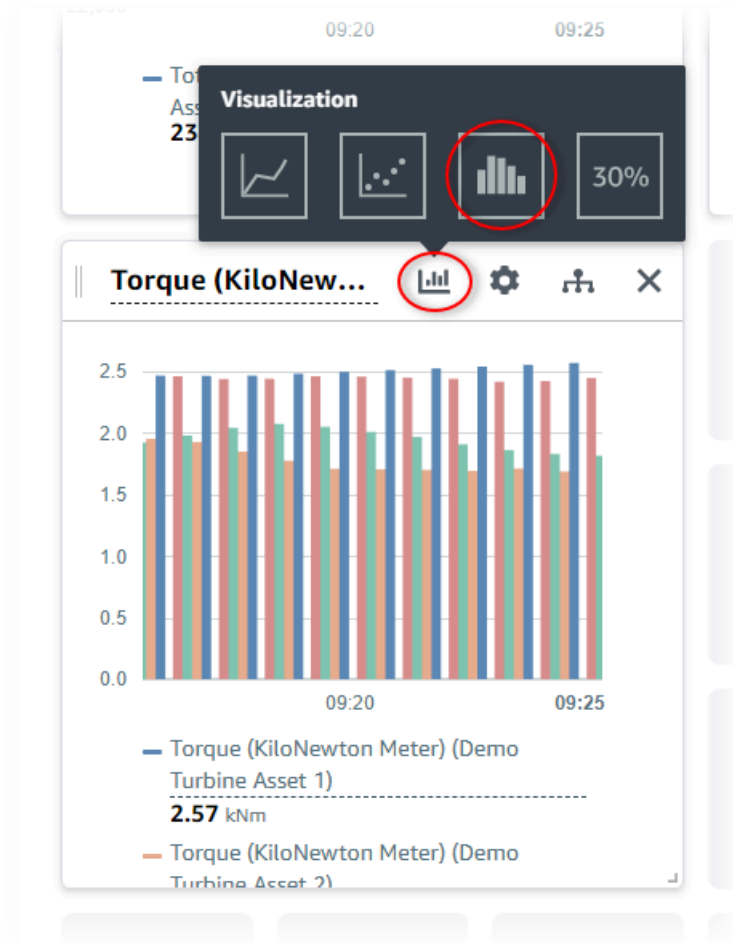


5. Addition Wind Speed à la nouvelle visualisation de la vitesse du vent pour chacun Demo Turbine Asset 2, 3, et 4 (dans cet ordre).

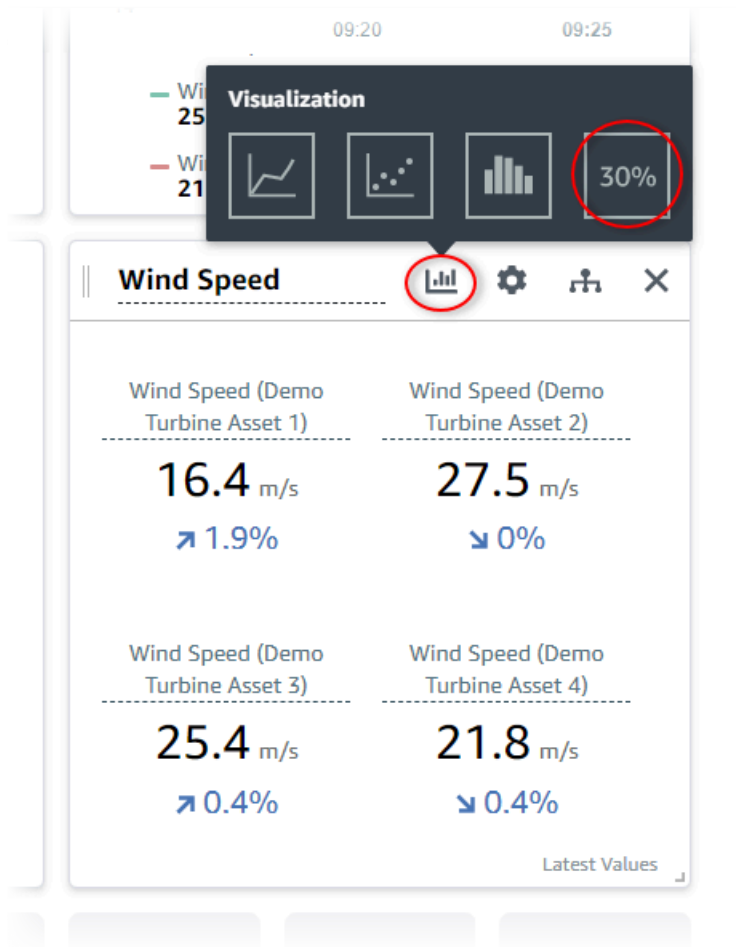
Votre Wind Speed la visualisation doit ressembler à la capture d'écran suivante.



6. Répétez les étapes 4 et 5 pour les éoliennes. » Torque (KiloNewton Meter) propriétés permettant de créer une visualisation du couple de l'éolienne.
7. Choisissez l'icône du type de visualisation pour le Torque (KiloNewton Meter) visualisation, puis choisissez l'icône du graphique à barres.

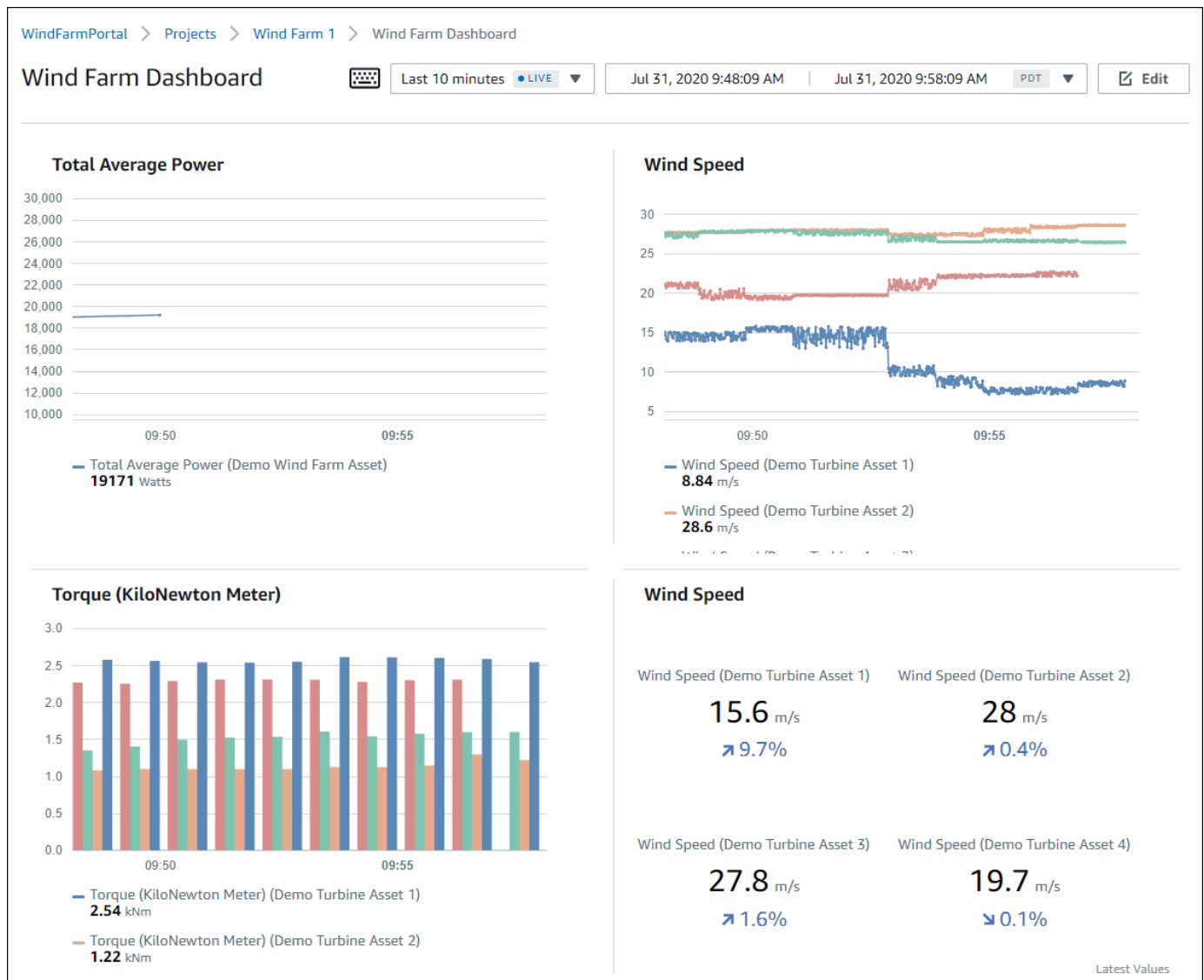


8. Répétez les étapes 4 et 5 pour les éoliennes. » Wind Direction propriétés pour créer une visualisation de la direction du vent.
9. Choisissez l'icône du type de visualisation pour le Wind Direction visualisation, puis choisissez l'icône du graphique KPI (30%).



10. (Facultatif) Apportez d'autres modifications au titre de la visualisation, aux titres de légende, au type, à la taille et à l'emplacement, si nécessaire.
11. Choisissez Enregistrer le tableau de bord en haut à droite pour enregistrer votre tableau de bord.

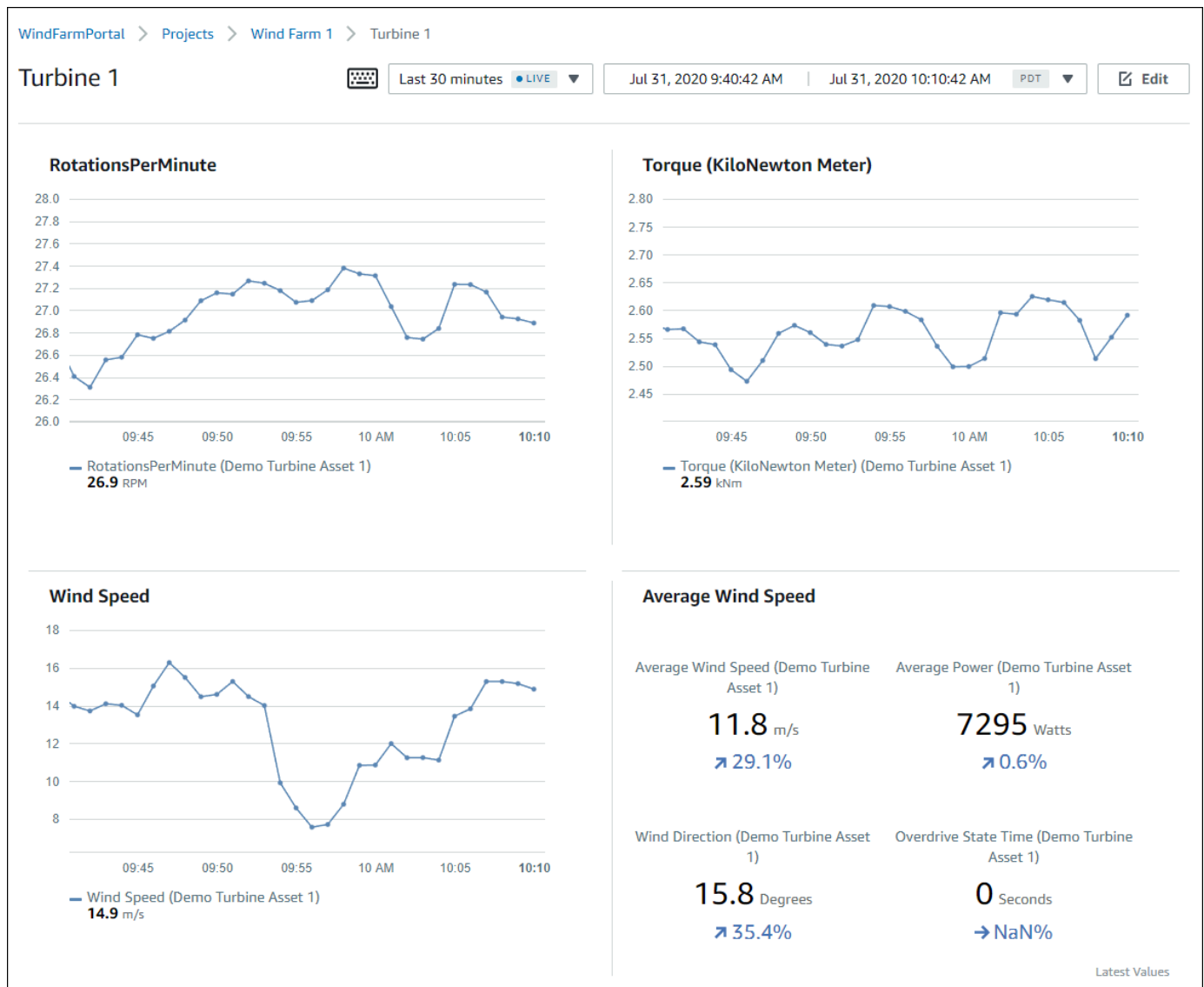
Votre tableau de bord devrait ressembler à la capture d'écran suivante.



12. (Facultatif) Créez un tableau de bord supplémentaire pour chaque ressource éolienne.

À titre de bonne pratique, nous vous recommandons de créer un tableau de bord pour chaque ressource afin que les utilisateurs de votre projet puissent examiner tous les problèmes liés à chaque ressource. Vous ne pouvez ajouter que 5 ressources à chaque visualisation. Vous devez donc créer plusieurs tableaux de bord pour vos ressources hiérarchiques dans de nombreux scénarios.

Un tableau de bord pour une éolienne de démonstration pourrait ressembler à la capture d'écran suivante.



13. (Facultatif) Modifiez la chronologie ou sélectionnez des points de données sur une visualisation pour explorer les données de votre tableau de bord. Pour plus d'informations, consultez la section [Affichage des tableaux de bord](#) dans le Guide de AWS IoT SiteWise Monitor l'application.

Étape 5 : Explorez le portail

Dans cette procédure, vous pouvez explorer le portail en tant qu'utilisateur disposant de moins d'autorisations qu'en tant qu'administrateur AWS IoT SiteWise du portail.

Pour explorer le portail et terminer le didacticiel

- (Facultatif) Si vous avez ajouté d'autres utilisateurs au projet en tant que propriétaires ou spectateurs, vous pouvez vous connecter au portail en tant que ces utilisateurs. Cela vous

permet d'explorer le portail en tant qu'utilisateur disposant de moins d'autorisations qu'en tant qu'administrateur du portail.

 Important

Vous êtes facturé pour chaque utilisateur qui se connecte à un portail. Pour plus d'informations, consultez [AWS IoT SiteWise Pricing](#) (Tarification CTlong).

Pour explorer le portail avec d'autres utilisateurs, procédez comme suit :

- a. Choisissez Déconnexion en bas à gauche du portail pour quitter l'application Web.
- b. Choisissez Se déconnecter dans le coin supérieur droit du portail de l'application IAM Identity Center pour vous déconnecter de votre utilisateur IAM Identity Center.
- c. Connectez-vous au portail en tant qu'utilisateur IAM Identity Center que vous avez désigné en tant que propriétaire ou visionneur de projet. Pour de plus amples informations, veuillez consulter [Étape 2 : Connectez-vous à un portail](#).

Vous avez terminé le didacticiel. Lorsque vous aurez fini d'explorer votre parc éolien de démonstration dans SiteWise Monitor, suivez la procédure suivante pour nettoyer vos ressources.

Étape 6 : Nettoyer les ressources après le didacticiel

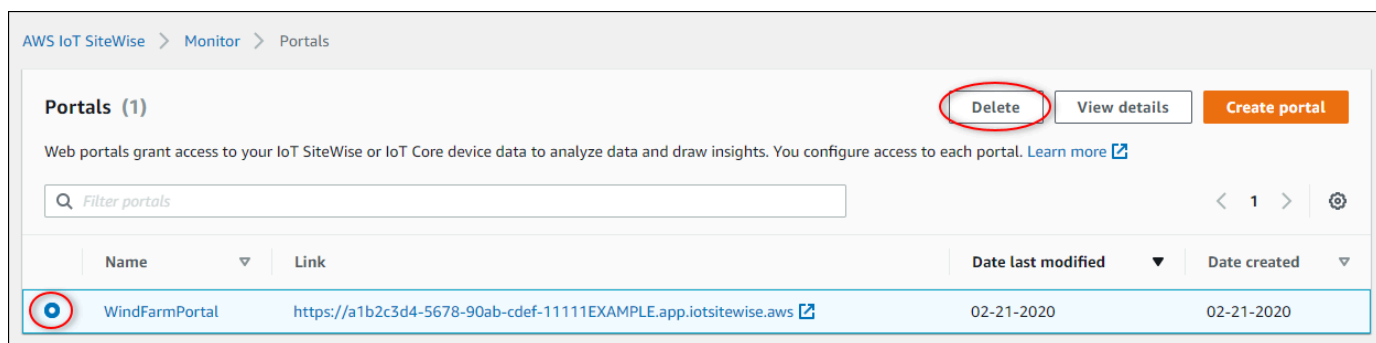
Après avoir terminé le didacticiel, vous pouvez nettoyer vos ressources. Vous n'êtes pas facturé pour AWS IoT SiteWise si les utilisateurs ne se connectent pas à votre portail, mais vous pouvez supprimer votre portail et les utilisateurs Répertoire AWS IAM Identity Center . Vos ressources de parc éolien de démonstration sont supprimées à la fin de la durée que vous avez choisie lors de la création de la démo, ou vous pouvez supprimer la démo manuellement. Pour de plus amples informations, veuillez consulter [Supprimer la AWS IoT SiteWise démo](#).

Suivez les procédures suivantes pour supprimer les utilisateurs de votre portail et de l'IAM Identity Center.

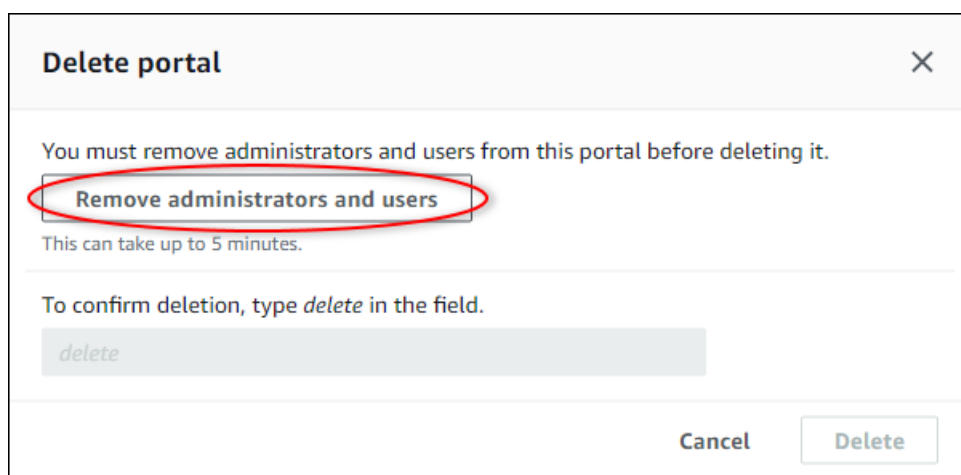
Pour supprimer un portail

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Portals (Portails).
3. Choisissez votre portail WindFarmPortal, puis sélectionnez Supprimer.

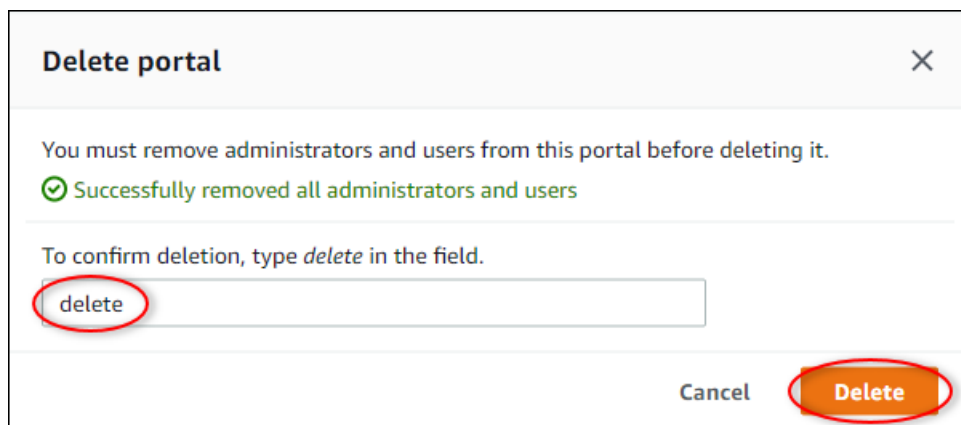
Lorsque vous supprimez un portail ou un projet, les ressources associées aux projets supprimés ne sont pas affectées.



4. Dans la boîte de dialogue Supprimer le portail, choisissez Supprimer les administrateurs et les utilisateurs.

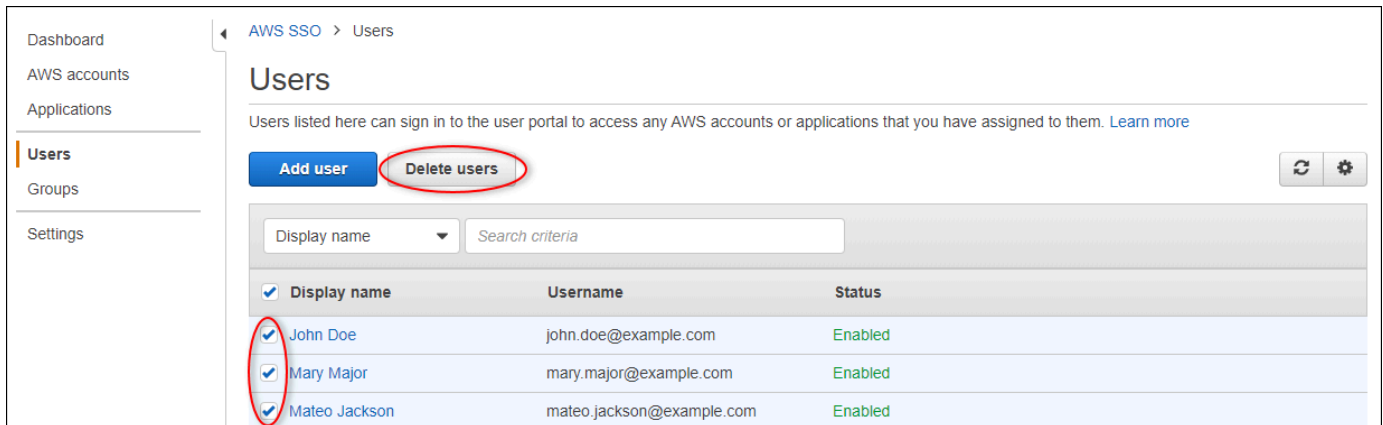


5. Entrez **delete** pour confirmer la suppression, puis choisissez Supprimer.

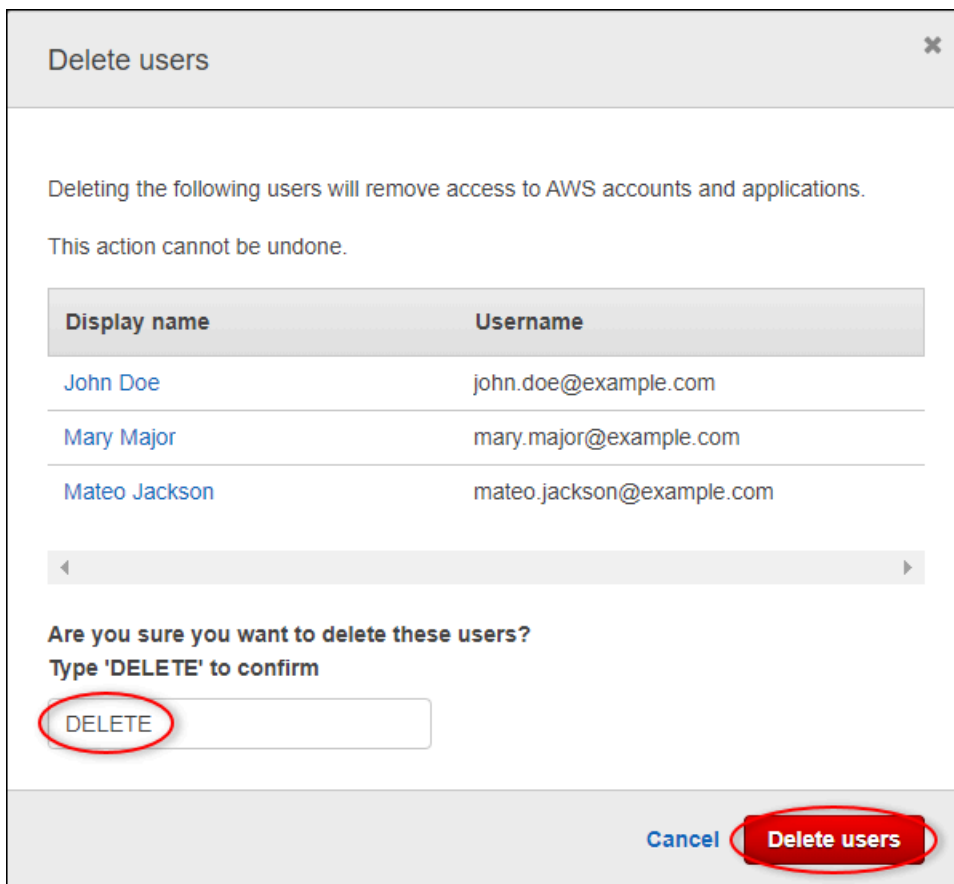


Pour supprimer des utilisateurs d'IAM Identity Center

1. Accédez à la [console IAM Identity Center](#).
2. Dans le volet de navigation de gauche, choisissez Utilisateurs.
3. Activez la case à cocher pour chaque utilisateur à supprimer, puis choisissez Delete users (Supprimer les utilisateurs).



4. Dans la boîte de dialogue Supprimer les utilisateurs, entrez **DELETE**, puis choisissez Supprimer les utilisateurs.



Publier les mises à jour de la valeur des propriétés sur Amazon DynamoDB

Ce didacticiel présente un moyen pratique de stocker vos données à l'aide d'[Amazon DynamoDB](#), afin de faciliter l'accès aux données historiques des actifs sans avoir à interroger l'API à plusieurs reprises. AWS IoT SiteWise Une fois ce didacticiel terminé, vous pouvez créer un logiciel personnalisé qui utilise les données de vos actifs, comme une carte en temps réel de la vitesse et de la direction du vent sur l'ensemble d'un parc éolien. Si vous souhaitez surveiller et visualiser vos données sans implémenter de solution logicielle personnalisée, consultez [Surveillez les données avec AWS IoT SiteWise Monitor](#).

Dans ce didacticiel, vous vous appuyerez sur la AWS IoT SiteWise démo qui fournit un exemple de jeu de données pour un parc éolien. Vous configurez les mises à jour de la valeur des propriétés à partir de la démonstration du parc éolien pour envoyer des données, via des règles de AWS IoT base, à une table DynamoDB que vous créez. Lorsque vous activez les mises à jour de la valeur des propriétés, AWS IoT SiteWise envoie vos données AWS IoT Core dans des messages MQTT. Définissez ensuite des règles de AWS IoT base qui exécutent des actions, telles que l'action DynamoDB, en fonction du contenu de ces messages. Pour de plus amples informations, veuillez consulter [Interagissez avec d'autres AWS services](#).

Rubriques

- [Prérequis](#)
- [Étape 1 : Configuration AWS IoT SiteWise pour publier les mises à jour de la valeur des propriétés](#)
- [Étape 2 : créer une règle dans AWS IoT Core](#)
- [Étape 3 : Création d'une table DynamoDB](#)
- [Étape 4 : Configuration de l'action de la règle DynamoDB](#)
- [Étape 5 : explorer les données dans DynamoDB](#)
- [Étape 6 : Nettoyer les ressources après le didacticiel](#)

Prérequis

Pour suivre ce didacticiel, vous aurez besoin des éléments suivants :

- Un AWS compte. Si vous n'en avez pas, veuillez consulter [Configurez un AWS compte](#).

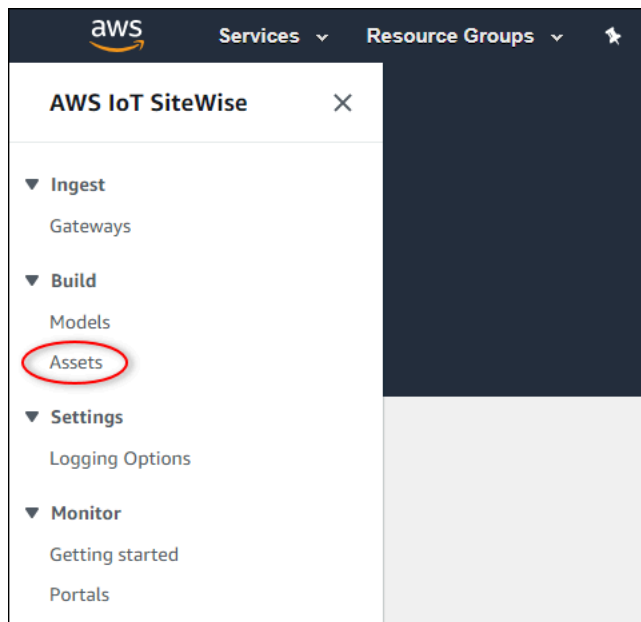
- Un ordinateur de développement exécutant Windows, macOS, Linux ou Unix pour accéder au AWS Management Console. Pour plus d'informations, voir [Qu'est-ce que le AWS Management Console ?](#)
- Utilisateur IAM possédant des autorisations de niveau administrateur.
- Une démonstration de AWS IoT SiteWise parc éolien en cours d'exécution. Lorsque vous configurez la démo, elle définit les modèles et les actifs AWS IoT SiteWise et leur transmet des données pour représenter un parc éolien. Pour de plus amples informations, veuillez consulter [Utiliser la AWS IoT SiteWise démo](#).

Étape 1 : Configuration AWS IoT SiteWise pour publier les mises à jour de la valeur des propriétés

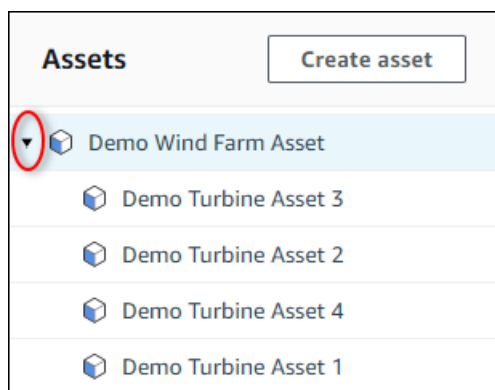
Dans cette procédure, vous activez les notifications relatives à la valeur des propriétés sur les actifs de vos turbines de démonstration. Wind Speed propriétés. Une fois que vous avez activé les notifications de valeur de propriété, AWS IoT SiteWise publie chaque mise à jour de valeur dans un message MQTT envoyé à AWS IoT Core.

Pour activer les notifications de mise à jour de valeur des propriétés de ressources

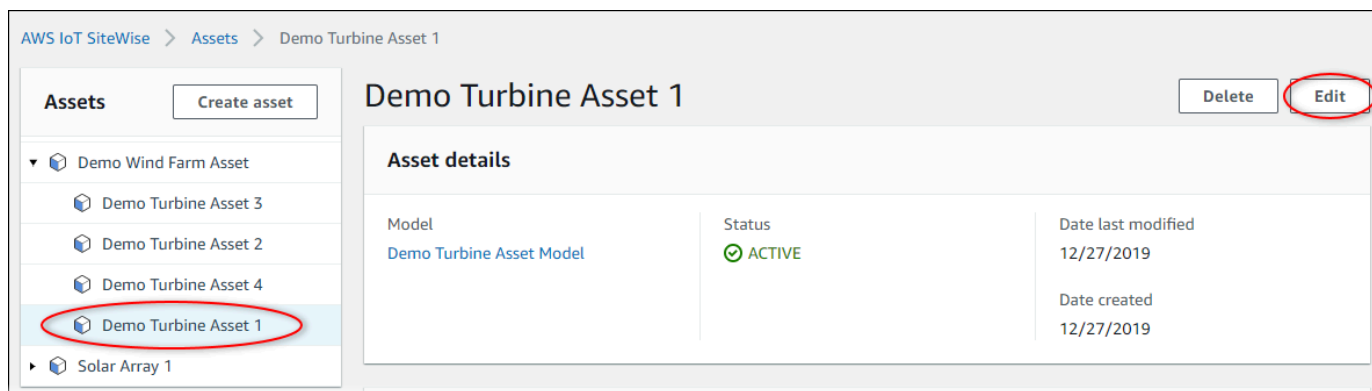
1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Passez en revue les [AWS IoT SiteWise points de terminaison et les quotas](#) pris en charge et changez de AWS région, si nécessaire. AWS IoT SiteWise Basculez vers la région dans laquelle vous exécutez la AWS IoT SiteWise démo.
3. Dans le panneau de navigation de gauche, choisissez Assets (Ressources).



4. Cliquez sur la flèche à côté de Demo Wind Farm Asset pour élargir la hiérarchie des actifs du parc éolien.



5. Choisissez une éolienne de démonstration et choisissez Edit (Modifier).



6. Mettre à jour le Wind SpeedLe statut de notification de la propriété est activé.

"Wind Speed"

Enter a property alias

Must be less than 2048 characters.

Notification status

ENABLED

Notification will be published to topic \$aws/sitewise/asset-models/d8f8f20a-4d3a-491c-a9c5-352736979bdb/assets/db36f80f-ed03-44d9-84ef-817eb30d5497/properties/ca5b9e21-f19c-4ea1-8472-0e9400fc12bf

7. En bas de la page, choisissez Save asset (Enregistrer la ressource).
8. Répétez les étapes 5 à 7 pour chaque ressource d'éolienne de démonstration.
9. Choisissez une turbine de démonstration (par exemple, Demo Turbine Asset 1).
10. Choisissez Measurements (Mesures).
11. Choisissez l'icône de copie située à côté du Wind Speed propriété permettant de copier le sujet de notification dans votre presse-papiers. Enregistrez la rubrique de notification qui sera utilisée ultérieurement dans ce didacticiel. Vous avez seulement besoin d'enregistrer la rubrique de notification d'une éolienne.

Torque (KiloNewton Meter)	-	⊖ Disabled	-	2.128123
Wind Speed	-	✔ Enabled	\$aws/sitewise/asset-models/d8f8f...	26.49812

La rubrique de notification doit ressembler à l'exemple suivant.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

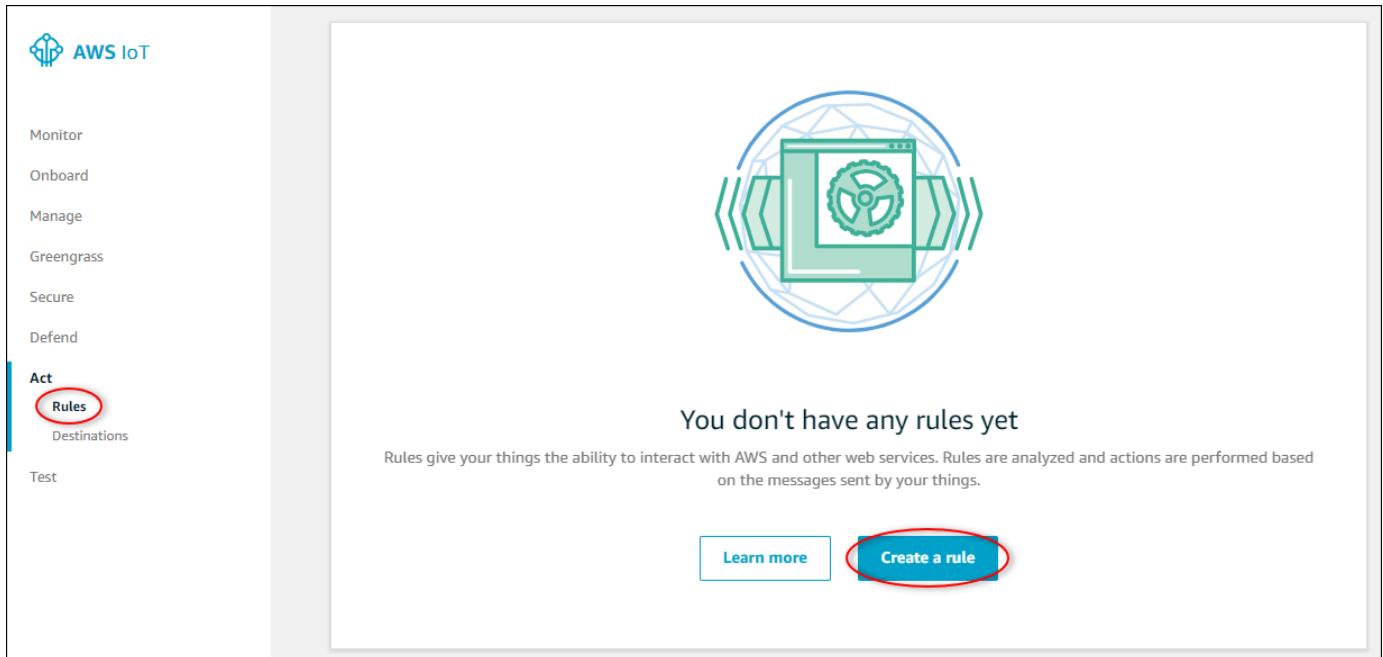
Étape 2 : créer une règle dans AWS IoT Core

Dans cette procédure, vous créez une règle dans AWS IoT Core qui analyse les messages de notification relatifs à la valeur de la propriété et insère des données dans une table Amazon DynamoDB. AWS IoT Les règles de base analysent les messages MQTT et exécutent des actions en fonction du contenu et du sujet de chaque message. Vous créez ensuite une règle avec une action DynamoDB pour insérer des données dans une table DynamoDB que vous créez dans le cadre de ce didacticiel.

Pour créer une règle avec une action DynamoDB

1. Accédez à la [console AWS IoT](#). Si un bouton Get started (Démarrer) apparaît, sélectionnez-le.

2. Dans le panneau de navigation de gauche, choisissez Act (Agir) puis Rules (Règles).



3. Si une boîte de dialogue Vous ne possédez pas encore de règle s'affiche, choisissez Créer une règle. Sinon, cliquez sur Create.
4. Saisissez un nom et une description pour la règle.

5. Recherchez la rubrique de notification que vous avez enregistrée précédemment dans ce didacticiel.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE
```

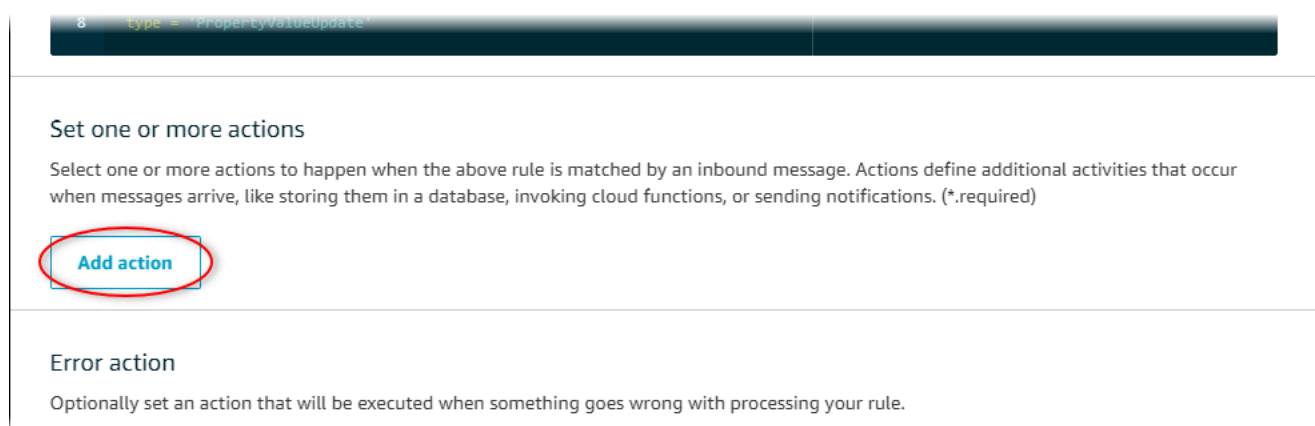
Remplacez l'ID de ressource (l'ID après `assets/`) dans la rubrique par un `+`. Cela permet de sélectionner la propriété de vitesse du vent pour tous les actifs d'éoliennes de démonstration. Le filtre de rubrique `+` accepte tous les nœuds d'un seul niveau dans une rubrique. Votre sujet doit ressembler à l'exemple suivant.

```
$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE
```

- Entrez l'instruction de requête de règle suivante. Remplacez la rubrique de la section FROM par votre rubrique de notification.

```
SELECT
  payload.assetId AS asset,
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed,
  timestamp() AS timestamp
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/'
  properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

- Sous Définissez une ou plusieurs actions, choisissez Ajouter une action.



8 `type = 'PropertyValueUpdate'`

Set one or more actions

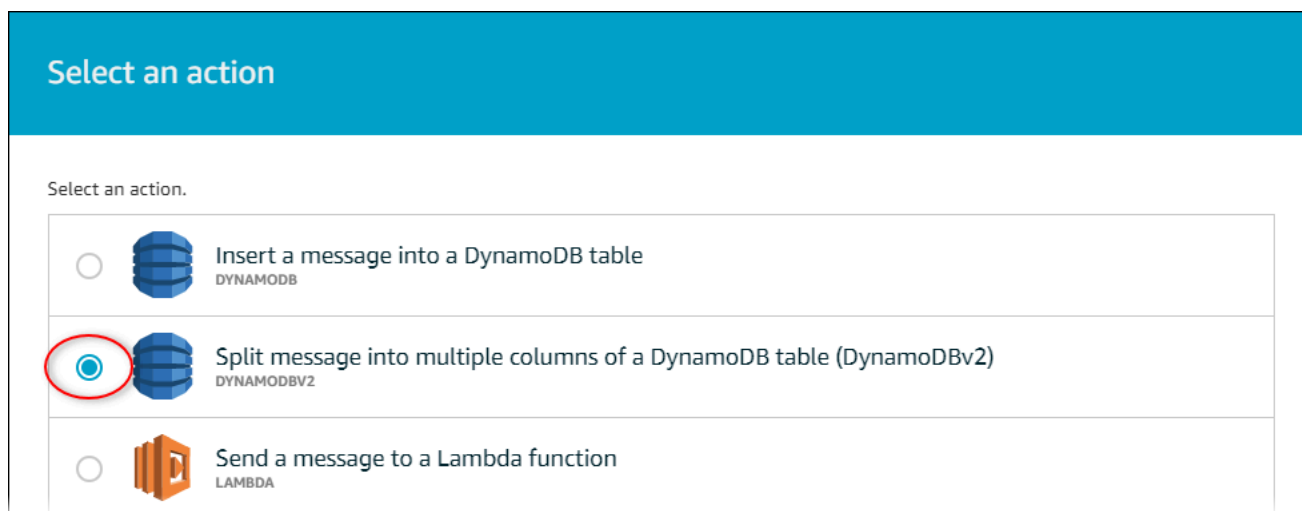
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (*.required)

Add action

Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

- Sur la page Sélectionner une action, choisissez Diviser le message en plusieurs colonnes d'une table DynamoDB (Dynamo). DBv2



9. En bas de la page, choisissez Configure action (Configurer l'action).
10. Sur la page Configure action, choisissez Create a new resource.

La console DynamoDB s'ouvre dans un nouvel onglet. Laissez l'onglet de l'action de la règle ouvert pendant que vous effectuez les procédures suivantes.

Étape 3 : Création d'une table DynamoDB

Dans cette procédure, vous créez une table Amazon DynamoDB pour recevoir les données de vitesse du vent issues de l'action de la règle.

Pour créer une table DynamoDB

1. Dans le tableau de bord de la console DynamoDB, choisissez Create table.
2. Saisissez un nom pour votre table.

Create DynamoDB table

Tutorial ?

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

Table name* ⓘ

Primary key* Partition key

ⓘ

☒ Add sort key

ⓘ

Table settings

Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.

☒ Use default settings

- No secondary indexes.
- Provisioned capacity set to 5 reads and 5 writes.
- Basic alarms with 80% upper threshold using SNS topic "dynamodb".
- Encryption at Rest with DEFAULT encryption type.

ⓘ You do not have the required role to enable Auto Scaling by default.
Please refer to [documentation](#).

+ Add tags **NEW!**

Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.

Cancel **Create**

3. Pour Clé primaire, procédez comme suit :
 - a. Pour la clé de partition, saisissez **timestamp**.
 - b. Choisissez le type Numéro .
 - c. Cochez la case Ajouter une clé de tri.
 - d. Pour la clé de tri, saisissez **asset** et laissez le type de clé de tri par défaut Chaîne.
4. Sélectionnez Create (Créer).

Lorsque le message La table est en cours de création. disparaît, votre table est prête.

5. Retournez à l'onglet avec la page Configure action (Configurer l'action). Gardez l'onglet DynamoDB ouvert pendant que vous effectuez les procédures suivantes.

Étape 4 : Configuration de l'action de la règle DynamoDB

Dans cette procédure, vous configurez l'action de règle Amazon DynamoDB pour insérer les données issues des mises à jour des valeurs de propriété dans votre nouvelle table DynamoDB.

Pour configurer l'action de règle DynamoDB

1. Sur la page Configurer l'action, actualisez la liste des noms de table et choisissez votre nouvelle table DynamoDB.

2. Choisissez Create role pour créer un rôle IAM qui accorde un accès AWS IoT Core pour exécuter l'action de règle.
3. Saisissez un nom de rôle, puis choisissez Create role (Créer un rôle).

4. Choisissez Add action.
5. Choisissez Create rule (Créer une règle) au bas de la page pour terminer la création de la règle.

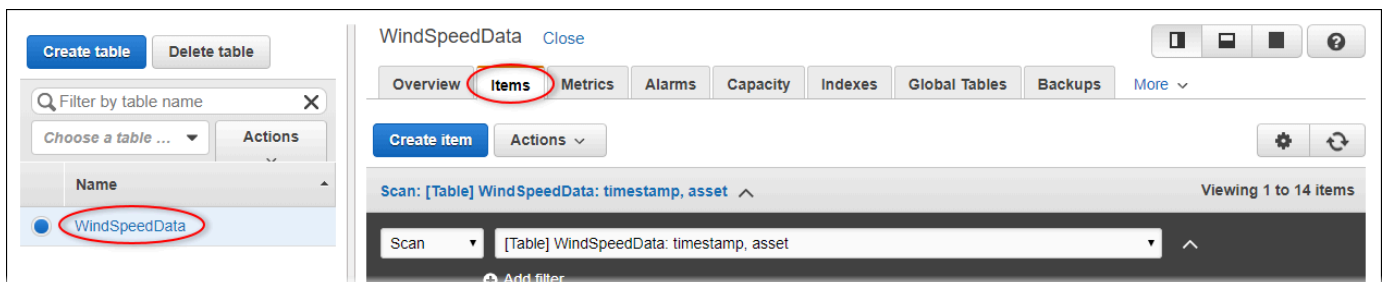
Les données de vos actifs de démonstration devraient commencer à apparaître dans votre table DynamoDB.

Étape 5 : explorer les données dans DynamoDB

Dans cette procédure, vous allez explorer les données de vitesse du vent des actifs de démonstration dans votre nouvelle table Amazon DynamoDB.

Pour explorer les données relatives aux actifs dans DynamoDB

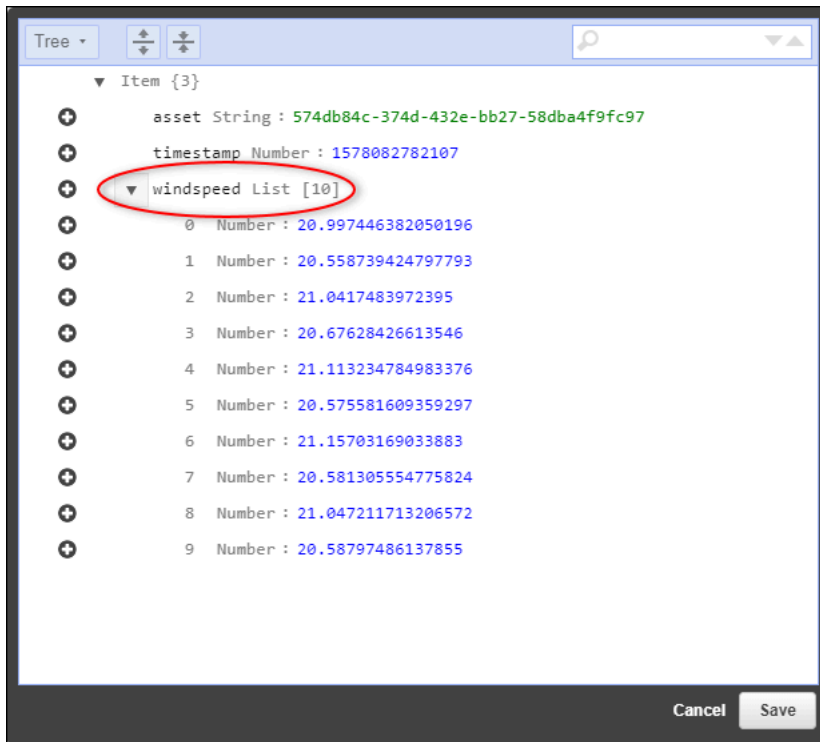
1. Retournez à l'onglet avec la table DynamoDB ouverte.
2. Dans la table que vous avez créée précédemment, choisissez l'onglet Éléments pour afficher les données de la table. Actualisez la page si la table ne comporte aucune ligne. Si aucune ligne n'apparaît après quelques minutes, consultez [Résolution des problèmes liés à une règle \(DynamoDB\)](#).



3. Dans une ligne du tableau, choisissez l'icône de modification pour développer les données.

	timestamp ⓘ ▲	asset ▼	windspeed ▼
☐	1578093637414	db36f80f-ed03-44d9-84ef-817eb30d5497	[{ "N": "40.18707553698584" }, { "N": "40.20834808480326" }, { "N": "40.21820505495924" }]
☐	1578093637422	db36f80f-ed03-44d9-84ef-817eb30d5497	[{ "N": "40.21081344172715" }, { "N": "40.21820888809424" }, { "N": "40.21820505495924" }]
☐	1578093637451	db36f80f-ed03-44d9-84ef-817eb30d5497	[{ "N": "40.218912043562895" }, { "N": "40.22691091326525" }, { "N": "40.21820505495924" }]
☐	1578093637453	db36f80f-ed03-44d9-84ef-817eb30d5497	[{ "N": "40.22876939941959" }, { "N": "40.21820505495924" }, { "N": "40.21820505495924" }]

4. Cliquez sur la flèche à côté du windspeedstructure pour étendre la liste des points de données sur la vitesse du vent. Chaque liste reflète un lot de points de données sur la vitesse du vent envoyés AWS IoT SiteWise par la démo du parc éolien. Vous pouvez avoir besoin d'un format de données différent si vous configurez une action de règle adaptée à votre utilisation. Pour de plus amples informations, veuillez consulter [Interrogez les notifications relatives aux propriétés des actifs dans AWS IoT SiteWise](#).



Maintenant que vous avez terminé le didacticiel, désactivez ou supprimez la règle et supprimez votre table DynamoDB pour éviter des frais supplémentaires. Pour nettoyer vos ressources, voir [Étape 6 : Nettoyer les ressources après le didacticiel](#).

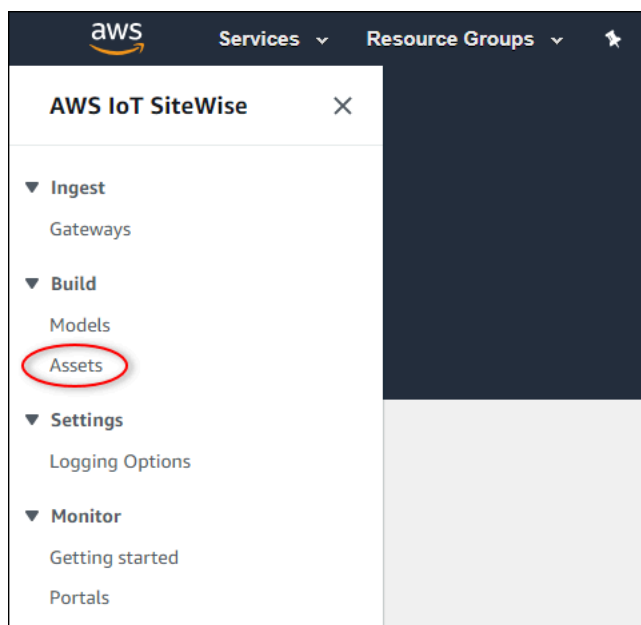
Étape 6 : Nettoyer les ressources après le didacticiel

Une fois que vous avez terminé le didacticiel, nettoyez les ressources pour éviter d'encourir des frais supplémentaires. Les actifs de votre parc éolien de démonstration sont supprimés à la fin de la durée que vous avez choisie lors de la création de la démo. Vous pouvez également supprimer la démo manuellement. Pour de plus amples informations, veuillez consulter [Supprimer la AWS IoT SiteWise démo](#).

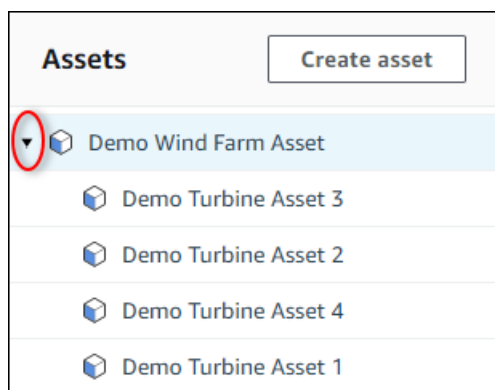
Utilisez les procédures suivantes pour désactiver les notifications de mise à jour de la valeur des propriétés (si vous n'avez pas supprimé la démo), désactiver ou supprimer votre AWS IoT règle et supprimer votre table DynamoDB.

Pour désactiver les notifications de mise à jour de valeur des propriétés de ressources

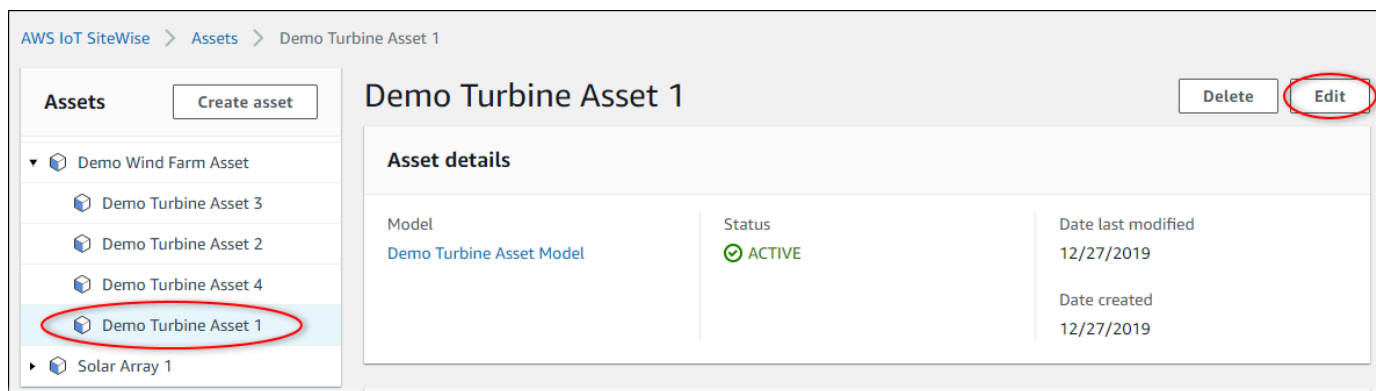
1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation de gauche, choisissez Assets (Ressources).



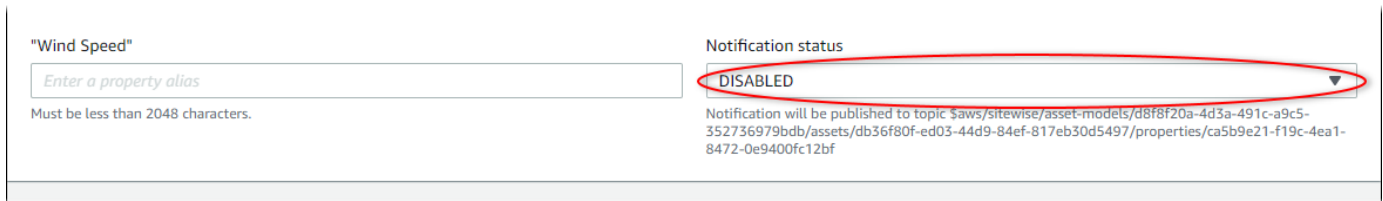
3. Cliquez sur la flèche à côté de Demo Wind Farm Asset pour élargir la hiérarchie des actifs du parc éolien.



4. Choisissez une éolienne de démonstration et choisissez Edit (Modifier).



5. Mettre à jour le Wind SpeedLe statut de notification de la propriété est défini sur DÉSACTIVÉ.



"Wind Speed"

Enter a property alias

Must be less than 2048 characters.

Notification status

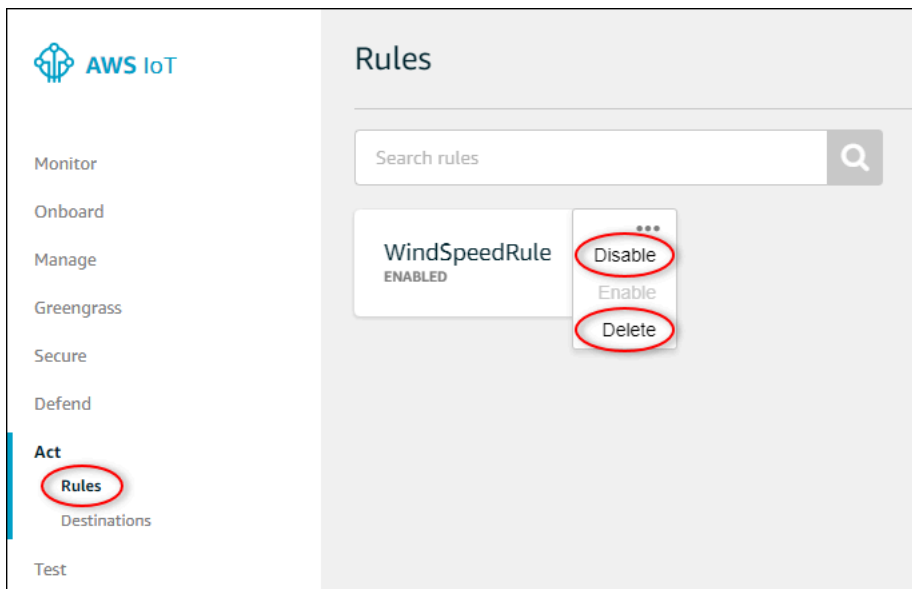
DISABLED

Notification will be published to topic \$aws/sitewise/asset-models/d8f8f20a-4d3a-491c-a9c5-352736979bdb/assets/db36f80f-ed03-44d9-84ef-817eb30d5497/properties/ca5b9e21-f19c-4ea1-8472-0e9400fc12bf

6. En bas de la page, choisissez Save asset (Enregistrer la ressource).
7. Répétez les étapes 4 à 6 pour chaque ressource d'éolienne de démonstration.

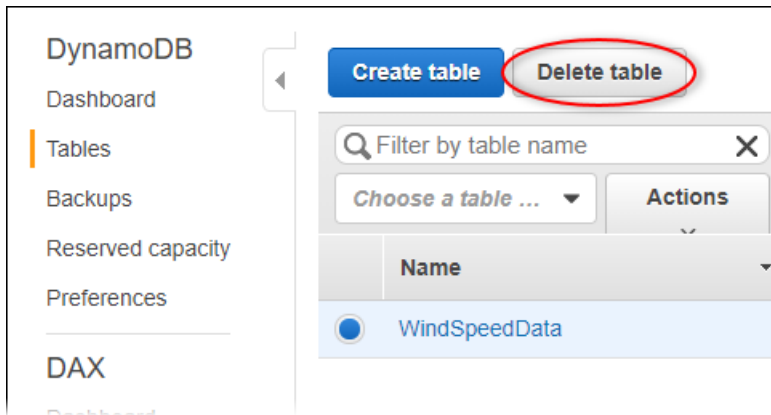
Pour désactiver ou supprimer une règle dans AWS IoT Core

1. Accédez à la [console AWS IoT](#).
2. Dans le panneau de navigation de gauche, choisissez Act (Agir) puis Rules (Règles).
3. Choisissez le menu de votre règle et choisissez Désactiver ou Supprimer.

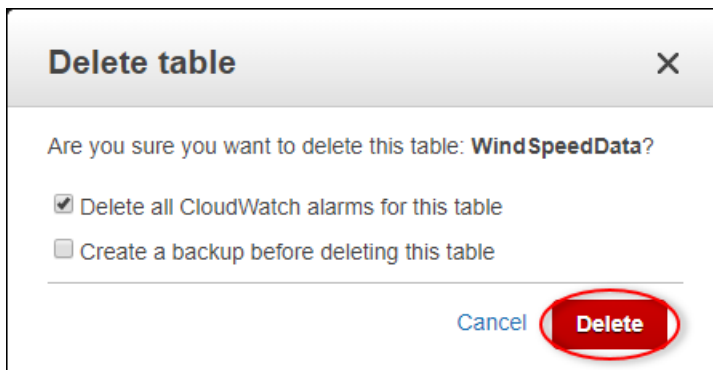


Pour supprimer une table DynamoDB

1. Accédez à la console [DynamoDB](#).
2. Dans le volet de navigation de gauche, choisissez Tables.
3. Choisissez le tableau que vous avez créé précédemment, WindSpeedData.
4. Choisissez Supprimer la table.



5. Dans la boîte de dialogue Delete table (Supprimer la table), choisissez Delete (Supprimer).



Ingérez des données dans AWS IoT SiteWise

AWS IoT SiteWise est conçu pour collecter et corréler efficacement les données industrielles avec les actifs correspondants, représentant divers aspects des opérations industrielles. Cette documentation se concentre sur les aspects pratiques de l'ingestion de données AWS IoT SiteWise, en proposant de multiples méthodes adaptées à divers cas d'utilisation industrielle. Pour obtenir des instructions sur la création de votre opération industrielle virtuelle, veuillez consulter [Modéliser des actifs industriels](#).

Vous pouvez envoyer des données industrielles à AWS IoT SiteWise l'aide de l'une des options suivantes :

- AWS IoT SiteWise Edge —Utilisez la [passerelle SiteWise Edge](#) comme intermédiaire entre AWS IoT SiteWise et vos serveurs de données. AWS IoT SiteWise fournit des AWS IoT Greengrass composants que vous pouvez déployer sur n'importe quelle plate-forme pouvant être exécutée AWS IoT Greengrass pour configurer une passerelle SiteWise Edge. Cette option prend en charge la liaison avec le protocole de serveur [OPC UA](#).
- AWS IoT SiteWise API —Utilisez l'[AWS IoT SiteWise API](#) pour télécharger des données depuis n'importe quelle autre source. Utilisez notre [BatchPutAssetPropertyValue](#)API de streaming pour une ingestion en quelques secondes, ou l'[CreateBulkImportJob](#)API orientée par lots pour faciliter une ingestion rentable en lots plus importants.
- AWS IoT Règles de base : utilisez les [règles de AWS IoT base](#) pour télécharger des données à partir de messages MQTT publiés par un AWS IoT objet ou un autre AWS service.
- AWS IoT Events actions —Utilisez des [AWS IoT Events actions](#) déclenchées par des événements spécifiques dans AWS IoT Events. Cette méthode convient aux scénarios dans lesquels le téléchargement de données est lié à des événements.
- AWS IoT Greengrass gestionnaire de flux : utilisez le [gestionnaire de AWS IoT Greengrass flux](#) pour télécharger des données à partir de sources de données locales à l'aide d'un périphérique périphérique. Cette option convient aux situations dans lesquelles les données proviennent de sites locaux ou périphériques.

Ces méthodes offrent une gamme de solutions pour gérer les données provenant de différentes sources. Examinez les détails de chaque option pour acquérir une compréhension complète des fonctionnalités d'ingestion de données AWS IoT SiteWise .

Gérez les flux de données pour AWS IoT SiteWise

Un flux de données est la ressource qui contient des séries chronologiques historiques. Chaque flux de données est identifié par un alias unique, ce qui permet de suivre plus facilement l'origine de chaque donnée. Les flux de données sont automatiquement créés AWS IoT SiteWise lorsque les premières données de séries chronologiques sont reçues. Si les premières données de série chronologique sont identifiées par un alias, AWS IoT SiteWise crée un nouveau flux de données avec cet alias, à condition qu'aucune propriété d'actif ne soit déjà attribuée à cet alias. Sinon, si les premières données de série chronologique sont identifiées par un ID d'actif et un ID de propriété, un nouveau flux de données est AWS IoT SiteWise créé et associe ce flux de données à la propriété de l'actif.

Il existe deux manières d'attribuer un alias à une propriété d'actif. La méthode utilisée dépend du fait que les données sont envoyées au AWS IoT SiteWise premier ou qu'un actif est créé en premier.

- Si les données sont envoyées au AWS IoT SiteWise premier, cela crée automatiquement un flux de données avec l'alias attribué. Lorsque la ressource est créée ultérieurement, utilisez l' [AssociateTimeSeriesToAssetProperty](#) API pour associer le flux de données et son alias à la propriété de la ressource.
- Si un actif est créé en premier, utilisez l' [UpdateAssetProperty](#) API pour attribuer un alias à une propriété d'actif. Lorsque les données sont ensuite envoyées à AWS IoT SiteWise, le flux de données est automatiquement créé et associé à la propriété de l'actif.

À l'heure actuelle, vous pouvez uniquement associer des flux de données à des mesures. Les mesures sont un type de propriété d'actif qui représente les flux de données brutes des capteurs des appareils, tels que les valeurs de température horodatées ou les valeurs de rotations par minute (RPM) horodatées.

Lorsque ces mesures définissent des métriques ou des transformations, les données entrantes déclenchent des calculs spécifiques. Il est important de noter qu'une propriété d'actif ne peut être liée qu'à un seul flux de données à la fois.

AWS IoT SiteWise utilise TimeSeries la ressource Amazon Resource Name (ARN) pour déterminer vos frais de stockage. Pour plus d'informations, consultez [AWS IoT SiteWise Pricing](#) (Tarification CTlong).

Les sections suivantes expliquent comment utiliser la AWS IoT SiteWise console ou l'API pour gérer les flux de données.

Rubriques

- [Configuration des autorisations et des paramètres](#)
- [Associer un flux de données à une propriété d'actif](#)
- [Dissocier un flux de données d'une propriété d'actif](#)
- [Supprimer un flux de données](#)
- [Mettre à jour un alias de propriété d'actif](#)
- [Scénarios courants](#)

Configuration des autorisations et des paramètres

Les flux de données sont automatiquement créés AWS IoT SiteWise lorsque les premières données de séries chronologiques sont reçues. Si les données ingérées ne sont pas associées à une propriété d'actif, AWS IoT SiteWise crée un nouveau flux de données dissocié qui est configurable pour être associé à une propriété d'actif. Configurez le contrôle d'accès de la passerelle à laquelle les données sont envoyées AWS IoT SiteWise, en utilisant les politiques IAM pour spécifier le type de données à ingérer.

La politique IAM suivante désactive l'ingestion de données dissociée depuis la passerelle, tout en autorisant l'ingestion de données vers les flux de données associés à une propriété d'actif :

Exemple Politique utilisateur IAM qui désactive l'ingestion de données dissociées depuis la passerelle

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAssetIdAndPropertyId",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*"
    },
    {
      "Sid": "AllowPutAssetPropertyValuesUsingAliasWithAssociatedAssetProperty",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
      "Condition": {
        "StringLikeIfExists": {
```

```

        "iotsitewise:isAssociatedWithAssetProperty": "true"
    }
},
{
    "Sid": "DenyPutAssetPropertyValuesUsingAliasWithNoAssociatedAssetProperty",
    "Effect": "Deny",
    "Action": "iotsitewise:BatchPutAssetPropertyValue",
    "Resource": "arn:aws:iotsitewise:*:*:time-series/*",
    "Condition": {
        "StringLikeIfExists": {
            "iotsitewise:isAssociatedWithAssetProperty": "false"
        }
    }
}
]
}

```

Exemple Politique utilisateur IAM qui désactive toute ingestion de données depuis la passerelle

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DenyPutAssetPropertyValues",
            "Effect": "Deny",
            "Action": "iotsitewise:BatchPutAssetPropertyValue",
            "Resource": {
                "arn:aws:iotsitewise:*:*:asset/*",
                "arn:aws:iotsitewise:*:*:time-series/*"
            }
        }
    ]
}

```

Associer un flux de données à une propriété d'actif

Gérez vos flux de données à l'aide du Console AWS IoT SiteWise ou AWS CLI.

Console

Utilisez la AWS IoT SiteWise console pour gérer vos flux de données.

Pour gérer les flux de données (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, sélectionnez Data streams.
3. Choisissez un flux de données en filtrant sur l'alias du flux de données ou en sélectionnant Flux de données dissociés dans le menu déroulant du filtre.
4. Sélectionnez le flux de données à mettre à jour. Vous pouvez sélectionner plusieurs flux de données. Cliquez sur Gérer les flux de données en haut à droite.
5. Sélectionnez le flux de données à associer dans Mettre à jour les associations de flux de données, puis cliquez sur le bouton Choisir une mesure.
6. Dans la section Choisir une mesure, recherchez la propriété de mesure de l'actif correspondante. Sélectionnez la mesure, puis cliquez sur Choisir.
7. Effectuez les étapes 4 et 5 pour les autres flux de données sélectionnés à l'étape 3. Attribuez des propriétés d'actifs à tous les flux de données.
8. Choisissez Mettre à jour pour valider les modifications. Une bannière de confirmation de réussite s'affiche pour confirmer la mise à jour.

AWS CLI

Pour associer un flux de données (identifié par son alias) à une propriété d'actif (identifiée par son IDs), exécutez la commande suivante :

```
aws iotsitewise associate-time-series-to-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Dissocier un flux de données d'une propriété d'actif

Console

Utilisez la AWS IoT SiteWise console pour dissocier votre flux de données d'une propriété d'actif.

Pour dissocier les flux de données d'une propriété d'actif (console)

1. Accédez à la [console AWS IoT SiteWise](#).

2. Dans le volet de navigation, sélectionnez Data streams.
3. Choisissez un flux de données en filtrant sur l'alias du flux de données ou en sélectionnant Flux de données associés dans le menu déroulant du filtre.
4. Sélectionnez le flux de données à dissocier. La colonne Alias du flux de données doit contenir un alias. Les colonnes Nom de la ressource et Nom de la propriété de la ressource doivent contenir les valeurs de la propriété de la ressource à laquelle le flux de données est associé. Vous pouvez sélectionner plusieurs flux de données.
5. Cliquez sur Gérer les flux de données en haut à droite.
6. Dans la section Mettre à jour les associations de flux de données, cliquez sur X dans la colonne Nom de la mesure. Un submitted statut doit apparaître dans la colonne État.
7. Choisissez Mettre à jour pour valider les modifications. Le flux de données est désormais dissocié de la propriété de l'actif, et l'alias est désormais utilisé pour identifier le flux de données.

AWS CLI

Pour dissocier un flux de données d'une propriété d'actif (identifiée par son ID s et son alias), exécutez la commande suivante :

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Le flux de données est désormais dissocié de la propriété de l'actif, et l'alias est utilisé pour identifier le flux de données. L'alias n'est plus associé à la propriété de l'actif, car il est désormais associé au flux de données.

Supprimer un flux de données

Lorsqu'une propriété est supprimée d'un modèle d'actifs, les AWS IoT SiteWise propriétés et leurs flux de données sont supprimés de tous les actifs gérés par le modèle d'actifs. Il supprime également toutes les propriétés et leurs flux de données d'un actif lorsque celui-ci est supprimé. Si les données d'un flux de données doivent être préservées, elles doivent être dissociées de la propriété de l'actif avant d'être supprimées.

 Warning

Lorsqu'une propriété est supprimée d'un actif, le flux de données associé est également supprimé. Pour préserver le flux de données, dissociez-le d'abord de la propriété de l'actif, avant de supprimer la propriété du modèle d'actif ou de supprimer l'actif.

Console

Utilisez la AWS IoT SiteWise console pour dissocier votre flux de données d'une propriété d'actif.

Pour supprimer un flux de données (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, sélectionnez Data streams.
3. Choisissez un flux de données en filtrant sur l'alias du flux de données.
4. Sélectionnez le flux de données à supprimer. Vous pouvez sélectionner plusieurs flux de données.
5. Cliquez sur le bouton Supprimer pour supprimer le flux de données.

AWS CLI

Utilisez l' [DeleteTimeSeries](#) API pour supprimer un flux de données spécifique, par son alias.

```
aws iotsitewise delete-time-series \  
  --alias <data-stream-alias>
```

Mettre à jour un alias de propriété d'actif

Les alias doivent être uniques au sein d'une AWS région. Cela inclut les alias des propriétés des actifs et des flux de données. N'attribuez pas d'alias à une propriété d'actif si une autre propriété ou un autre flux de données utilise cet alias.

Console

Utilisez la AWS IoT SiteWise console pour mettre à jour un alias de propriété d'actif.

Pour mettre à jour un alias de propriété d'actif (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Sélectionnez l'actif dans le tableau.
4. Cliquez sur le bouton Edit (Modifier).
5. Sélectionnez le type de propriété dans le tableau Propriétés.
6. Recherchez la propriété et saisissez le nouvel alias dans le champ de texte de l'alias de propriété.
7. Cliquez sur le bouton Enregistrer pour enregistrer les modifications.

AWS CLI

Pour mettre à jour un alias sur une propriété d'actif, exécutez la commande suivante :

```
aws iotsitewise update-asset-property \  
  --asset-id <asset-ID> \  
  --property-id <property-ID> \  
  --property-alias <asset-property-alias> \  
  --property-notification-state <ENABLED|DISABLED>
```

Note

Si les notifications de propriétés sont actuellement activées, elles doivent être fournies à nouveau pour garantir qu'elles continuent de l'être.

Scénarios courants

Déplacer un flux de données

Pour modifier l'association d'un flux de données à une autre propriété d'actif, dissociez d'abord le flux de données de la propriété d'actif actuelle. Lorsque vous dissociez un flux de données d'une propriété d'actif, un alias doit être attribué à cette propriété d'actif.

```
aws iotsitewise disassociate-time-series-from-asset-property \  
  --alias <asset-property-alias> \  
  --assetId <asset-ID> \  
  --propertyId <property-ID>
```

Réattribuez maintenant le flux de données à la nouvelle propriété de l'actif.

```
aws iotsitewise associate-time-series-from-asset-property \  
  --alias <data-stream-alias> \  
  --assetId <new-asset-ID> \  
  --propertyId <new-property-ID>
```

Erreur lors de l'attribution d'un alias à une propriété d'actif

Lorsque vous utilisez l'UpdateAssetPropertyAPI pour attribuer un alias à une propriété, le message d'erreur suivant peut s'afficher :

```
Given alias <data-stream-alias> for property <property-name> with ID <property-ID>  
already in use by another property or data stream
```

Ce message d'erreur indique que l'alias n'est pas attribué à la propriété, car il est actuellement utilisé par une autre propriété ou un flux de données.

Cela se produit si les données sont ingérées AWS IoT SiteWise avec un alias. Lorsque des données sont envoyées avec un alias qui n'est pas utilisé par un autre flux de données ou une autre propriété d'actif, un nouveau flux de données est créé avec cet alias. Les deux options ci-dessous permettent de résoudre le problème.

- Utilisez AssociateTimeSeriesToAssetProperty l'API pour associer le flux de données à son alias à la propriété de l'actif.
- Arrêtez temporairement l'ingestion de données et supprimez le flux de données. Utilisez UpdateAssetProperty l'API pour attribuer l'alias à la propriété de l'actif, puis réactivez l'ingestion de données.

Erreur lors de l'association d'un flux de données à une propriété d'actif

Lorsque vous associez un flux de données à une propriété d'actif, le message d'erreur suivant s'affiche.


```
assetProperty <property-name> with assetId <asset-ID> propertyId <property-ID> contains  
data
```

Ce message d'erreur indique que la propriété de l'actif est déjà associée à un flux de données contenant des données. Ce flux de données doit être dissocié ou supprimé avant d'associer un autre flux de données à cette propriété d'actif.

 Note

Lors de la dissociation d'un flux de données d'une propriété d'actif, l'alias attribué à la propriété est attribué au flux de données. Pour que cet alias reste attribué à la propriété, attribuez-lui un nouvel alias avant de dissocier le flux de données.

Pour conserver les données stockées dans la propriété de l'actif, procédez comme suit :

- Assurez-vous qu'aucune donnée n'est ingérée dans la propriété de l'actif, afin d'éviter la création d'un nouveau flux de données.
- Utilisez `UpdateAssetProperty` l'API pour définir un nouvel alias attribué au flux de données actuellement attribué.
- Utilisez `DisassociateTimeSeriesFromAssetProperty` l'API pour dissocier le flux de données actuel de la propriété de l'actif.
- Utilisez `AssociateTimeSeriesToAssetProperty` l'API pour associer le flux de données souhaité à la propriété de l'actif.

Si les données stockées dans la propriété de l'actif doivent être supprimées, procédez comme suit :

- Assurez-vous qu'aucune donnée n'est ingérée dans la propriété de l'actif, afin d'éviter la création d'un nouveau flux de données.
- Utilisez `DeleteTimeSeries` l'API pour supprimer le flux de données actuellement attribué.
- Utilisez `AssociateTimeSeriesToAssetProperty` l'API pour associer le flux de données souhaité à la propriété de l'actif.

Ingérez des données avec AWS IoT SiteWise APIs

Utilisez-le AWS IoT SiteWise APIs pour envoyer des données industrielles horodatées vers les attributs et les propriétés de mesure de vos actifs. APIs Accepte les structures contenant de la charge utile timestamp-quality-value (TQV).

BatchPutAssetPropertyValue API

Utilisez l'opération [BatchPutAssetPropertyValue](#) pour charger vos données. Grâce à cette opération, vous pouvez télécharger plusieurs entrées de données à la fois pour collecter des données provenant de plusieurs appareils et les envoyer en une seule demande.

Important

L'[BatchPutAssetPropertyValue](#) opération est soumise aux quotas suivants :

- Jusqu'à 10 [entrées](#) par demande.
- Jusqu'à 10 [valeurs de propriété](#) (points de données TQV) par entrée.
- AWS IoT SiteWise rejette toutes les données dont l'horodatage date de plus de 7 jours dans le passé ou de plus de 10 minutes dans le futur.

Pour de plus amples informations sur ces quotas, veuillez consulter [BatchPutAssetPropertyValue](#) dans la Référence de l'API AWS IoT SiteWise .

Pour identifier une propriété d'actif, spécifiez l'une des options suivantes :

- La `assetId` fin `propertyId` de la propriété de l'actif à laquelle les données sont envoyées.
- Le `propertyAlias`, qui est un alias de flux de données (par exemple, `/company/windfarm/3/turbine/7/temperature`). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour définir des alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

L'exemple suivant montre comment envoyer des lectures de température et de rotations par minute (RPM) d'une éolienne à partir d'une charge utile stockée dans un fichier JSON.

```
aws iotsitewise batch-put-asset-property-value --cli-input-json file://batch-put-payload.json
```

L'exemple de charge utile `batch-put-payload.json` contient le contenu suivant.

```
{
  "enablePartialEntryProcessing": true,
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 15.09
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          },
          "quality": "GOOD"
        }
      ]
    },
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/rpm",
      "propertyValues": [
        {
          "value": {
```

```

    "nullValue":{"valueType": "D"}
      },
      "timestamp": {
        "timeInSeconds": 1575691200
      },
      "quality": "BAD"
    }
  ]
}
]
}

```

La spécification « `enablePartialEntryProcessing` as » `true` permet l'ingestion de toutes les valeurs qui n'entraînent pas de défaillance. Le comportement par défaut est `false`. Si une valeur n'est pas valide, l'entrée entière échoue à être ingérée.

Chaque entrée de la charge utile contient un `entryId` que vous pouvez définir sous la forme d'une chaîne unique. Si des entrées de demande échouent, chaque erreur contiendra l'`entryId` de la demande correspondante afin que vous sachiez quelles demandes réessayer.

Chaque structure de la liste des `propertyValues` est une structure timestamp-quality-value (TQV) qui contient `a.value`, `a.timestamp`, `a` et éventuellement `a.quality`

- `value`— Structure contenant l'un des champs suivants, en fonction du type de propriété défini :
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `nullValue`— Une structure dont le champ suivant indique le type de la valeur de propriété avec la valeur `Null` et la qualité de `BAD` ou `UNCERTAIN`.
 - `valueType`— Enum de {`"B` », `"D` », `"S` », `"I` }
- `timestamp`— Une structure qui contient l'heure actuelle d'Unix en secondes, `timeInSeconds`. Vous pouvez également définir la `offsetInNanos` clé dans la `timestamp` structure si vous disposez de données temporellement précises. AWS IoT SiteWise rejette tous les points de données dont l'horodatage date de plus de 7 jours dans le passé ou de moins de 10 minutes dans le futur.
- `quality`— (Facultatif) L'une des chaînes de qualité suivantes :

- GOOD— (Par défaut) Les données ne sont affectées par aucun problème.
- BAD— Les données sont affectées par un problème tel qu'une défaillance du capteur.
- UNCERTAIN— Les données sont affectées par un problème tel que l'imprécision du capteur.

Pour plus d'informations sur la gestion de AWS IoT SiteWise la qualité des données dans les calculs, consultez la section [Qualité des données dans les expressions de formule](#).

CreateBulkImportJob API

Utilisez l>CreateBulkImportJobAPI pour importer de grandes quantités de données depuis Amazon S3. Vos données doivent être enregistrées au format CSV dans Amazon S3. Les fichiers de données peuvent comporter les colonnes suivantes.

Note

Les données antérieures au 1er janvier 1970 00:00:00 UTC ne sont pas prises en charge. Pour identifier une propriété d'actif, spécifiez l'une des options suivantes.

- La ASSET_ID fin PROPERTY_ID de la propriété de l'actif à laquelle vous envoyez des données.
- LeALIAS, qui est un alias de flux de données (par exemple,/company/windfarm/3/turbine/7/temperature). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour découvrir comment définir des alias de propriété de ressource, consultez [the section called "Gestion des flux de données"](#).

- ALIAS— L'alias qui identifie la propriété, tel que le chemin de flux de données d'un serveur OPC UA (par exemple,/company/windfarm/3/turbine/7/temperature). Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).
- ASSET_ID— L'ID de l'actif.
- PROPERTY_ID— L'ID de la propriété de l'actif.
- DATA_TYPE— Le type de données de la propriété peut être l'un des suivants.
 - STRING— Chaîne de 1024 octets maximum.
 - INTEGER— Un entier signé de 32 bits dont la plage est comprise entre [-2 147 483 648, 2 147 483 647].

- **DOUBLE**— Un nombre à virgule flottante avec une plage $[-10^{100}, 10^{100}]$ et une double précision IEEE 754.
- **BOOLEAN**— `true` ou `false`.
- **TIMESTAMP_SECONDS**— L'horodatage du point de données, à l'époque Unix.
- **TIMESTAMP_NANO_OFFSET**— Le décalage en nanosecondes converti à partir de **TIMESTAMP_SECONDS**.
- **QUALITY**— (Facultatif) La qualité de la valeur de la propriété de l'actif. La valeur peut être l'une des suivantes.
 - **GOOD**— (Par défaut) Les données ne sont affectées par aucun problème.
 - **BAD**— Les données sont affectées par un problème tel qu'une défaillance du capteur.
 - **UNCERTAIN**— Les données sont affectées par un problème tel que l'imprécision du capteur.

Pour plus d'informations sur la gestion de AWS IoT SiteWise la qualité des données dans les calculs, consultez la section [Qualité des données dans les expressions de formule](#).

- **VALUE**— La valeur de la propriété de l'actif.

Exemple fichier (s) de données au format .csv

```
asset_id,property_id,DOUBLE,1635201373,0,GOOD,1.0
asset_id,property_id,DOUBLE,1635201374,0,GOOD,2.0
asset_id,property_id,DOUBLE,1635201375,0,GOOD,3.0
```

```
unmodeled_alias1,DOUBLE,1635201373,0,GOOD,1.0
unmodeled_alias1,DOUBLE,1635201374,0,GOOD,2.0
unmodeled_alias1,DOUBLE,1635201375,0,GOOD,3.0
unmodeled_alias1,DOUBLE,1635201376,0,GOOD,4.0
unmodeled_alias1,DOUBLE,1635201377,0,GOOD,5.0
unmodeled_alias1,DOUBLE,1635201378,0,GOOD,6.0
unmodeled_alias1,DOUBLE,1635201379,0,GOOD,7.0
unmodeled_alias1,DOUBLE,1635201380,0,GOOD,8.0
unmodeled_alias1,DOUBLE,1635201381,0,GOOD,9.0
unmodeled_alias1,DOUBLE,1635201382,0,GOOD,10.0
```

AWS IoT SiteWise fournit les opérations d'API suivantes pour créer une tâche d'importation en bloc et obtenir des informations sur une tâche existante.

- [CreateBulkImportJob](#)— Crée une nouvelle tâche d'importation en bloc.

- [DescribeBulkImportJob](#)— Récupère les informations relatives à une tâche d'importation groupée.
- [ListBulkImportJob](#)— Récupère une liste paginée de résumés de toutes les tâches d'importation en bloc.

Création d'une tâche d'importation AWS IoT SiteWise en bloc (AWS CLI)

Utilisez l'opération [CreateBulkImportJob](#) d'API pour transférer des données d'Amazon S3 vers AWS IoT SiteWise. L'[CreateBulkImportJob](#) API permet l'ingestion de gros volumes de données historiques et l'ingestion en mémoire tampon de flux de données analytiques par petits lots. Il fournit une primitive rentable pour l'ingestion de données. L'exemple suivant repose sur AWS CLI.

Important

Avant de créer une tâche d'importation en bloc, vous devez activer le niveau AWS IoT SiteWise chaud ou le niveau AWS IoT SiteWise froid. Pour de plus amples informations, veuillez consulter [Configurer les paramètres de stockage dans AWS IoT SiteWise](#).

L'[CreateBulkImportJob](#) API prend en charge l'ingestion de données historiques AWS IoT SiteWise avec la possibilité de définir le `adaptive-ingestion-flag` paramètre.

- Lorsqu'elle est définie sur `false`, l'API ingère les données historiques sans déclencher de calculs ni de notifications.
- Lorsqu'elle est définie sur `true`, l'API ingère de nouvelles données, calcule des métriques et transforme les données afin d'optimiser les analyses et les notifications continues dans un délai de sept jours.

Exécutez la commande suivante. Remplacez *file-name* par le nom du fichier contenant la configuration de la tâche d'importation en bloc.

```
aws iotsitewise create-bulk-import-job --cli-input-json file://file-name.json
```

Exemple Configuration des tâches d'importation en bloc

Voici des exemples de paramètres de configuration :

- Remplacez *adaptive-ingestion-flag* par `true` ou `false`.
 - Si cette valeur est définie sur `false`, la tâche d'importation en bloc ingère les données historiques dans AWS IoT SiteWise.

- Si cette valeur est définie sur `true`, la tâche d'importation en bloc effectue les opérations suivantes :
 - Ingère de nouvelles données dans AWS IoT SiteWise
 - Calcule les métriques et les transforme, et prend en charge les notifications pour les données dont l'horodatage est établi dans les sept jours.
- Remplacez `delete-files-after-import-flag` par `true` pour supprimer les données du compartiment de données Amazon S3 après les avoir ingérées dans un système de stockage à AWS IoT SiteWise chaud.
- Remplacez `amzn-s3-demo-bucket` par le `-for-errors` nom du compartiment Amazon S3 auquel les erreurs associées à cette tâche d'importation en masse sont envoyées.
- Remplacez `amzn-s3-demo-bucket` par le `-for-errors-prefix` préfixe du compartiment Amazon S3 auquel les erreurs associées à cette tâche d'importation en masse sont envoyées.

Amazon S3 utilise le préfixe comme nom de dossier pour organiser les données dans le compartiment. Chaque objet Amazon S3 possède une clé qui constitue son identifiant unique dans le compartiment. Chaque objet d'un compartiment possède une clé et une seule. Le préfixe doit se terminer par une barre oblique (/). Pour plus d'informations, consultez la section [Organisation des objets à l'aide de préfixes](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

- Remplacez `amzn-s3-demo-bucket` par le `-data` nom du compartiment Amazon S3 à partir duquel les données sont importées.
- `data-bucket-key` Remplacez-la par la clé de l'objet Amazon S3 qui contient vos données. Chaque objet a une clé qui est un identifiant unique. Chaque objet a exactement une clé.
- `data-bucket-version-id` Remplacez-le par l'ID de version pour identifier une version spécifique de l'objet Amazon S3 contenant vos données. Ce paramètre est facultatif.
- Remplacez `column-name` par le nom de colonne spécifié dans le fichier .csv.
- `job-name` Remplacez-le par un nom unique identifiant la tâche d'importation en bloc.
- Remplacez `job-role-arn` par le rôle IAM qui permet AWS IoT SiteWise de lire les données Amazon S3.

Note

Assurez-vous que votre rôle dispose des autorisations indiquées dans l'exemple suivant. Remplacez `amzn-s3-demo-bucket` par le `-data` nom du compartiment Amazon S3 qui contient vos données. Remplacez-le également `amzn-s3-demo-bucket-for-errors` par

le nom du compartiment Amazon S3 auquel les erreurs associées à cette tâche d'importation en masse sont envoyées.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-data",
        "arn:aws:s3:::amzn-s3-demo-bucket-data/*",
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors",
        "arn:aws:s3:::amzn-s3-demo-bucket-for-errors/*"
      ],
      "Effect": "Allow"
    }
  ]
}
```

```
{
  "adaptiveIngestion": adaptive-ingestion-flag,
  "deleteFilesAfterImport": delete-files-after-import-flag,
  "errorReportLocation": {
    "bucket": "amzn-s3-demo-bucket-for-errors",
    "prefix": "amzn-s3-demo-bucket-for-errors-prefix"
  },
  "files": [
    {
```

```
    "bucket": "amzn-s3-demo-bucket-data",
    "key": "data-bucket-key",
    "versionId": "data-bucket-version-id"
  },
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [ "column-name" ]
      }
    }
  },
  "jobName": "job-name",
  "jobRoleArn": "job-role-arn"
}
```

Exemple réponse

```
{
  "jobId": "f8c031d0-01d1-4b94-90b1-afe8bb93b7e5",
  "jobStatus": "PENDING",
  "jobName": "myBulkImportJob"
}
```

Décrire une tâche d'importation AWS IoT SiteWise en masse (AWS CLI)

Utilisez l'opération [DescribeBulkImportJob](#) API pour récupérer des informations sur une tâche d'importation en bloc spécifique dans AWS IoT SiteWise. Cette opération renvoie des informations telles que le statut de la tâche, l'heure de création et les informations d'erreur en cas d'échec de la tâche. Vous pouvez utiliser cette opération pour suivre l'avancement des tâches et résoudre les problèmes. Pour l'utiliser `DescribeBulkImportJob`, vous avez besoin de l'ID de tâche de l'`CreateBulkImportJob` opération. L'API renvoie les informations suivantes :

- Liste des fichiers en cours d'importation, y compris l'emplacement et les clés de leur compartiment Amazon S3
- Emplacement du rapport d'erreur (le cas échéant)
- Détails de configuration des tâches, tels que le format de fichier et les noms de colonnes CSV
- Horodatage de la création d'un job et de la dernière mise à jour
- État actuel de la tâche (par exemple, si la tâche est en cours, terminée ou échouée)

- ARN du rôle IAM utilisé pour la tâche d'importation

Pour les tâches terminées, passez en revue les résultats pour confirmer que l'intégration des données est réussie. Si une tâche échoue, examinez les détails de l'erreur pour diagnostiquer et résoudre les problèmes.

job-ID Remplacez-le par l'ID de la tâche d'importation en bloc que vous souhaitez récupérer.

```
aws iotsitewise describe-bulk-import-job --job-id job-ID
```

Exemple réponse

```
{
  "files": [
    {
      "bucket": "amzn-s3-demo-bucket1",
      "key": "100Tags12Hours.csv"
    },
    {
      "bucket": "amzn-s3-demo-bucket2",
      "key": "BulkImportData1MB.csv"
    },
    {
      "bucket": " amzn-s3-demo-bucket3",
      "key": "UnmodeledBulkImportData1MB.csv"
    }
  ],
  "errorReportLocation": {
    "prefix": "errors/",
    "bucket": "amzn-s3-demo-bucket-for-errors"
  },
  "jobConfiguration": {
    "fileFormat": {
      "csv": {
        "columnNames": [
          "ALIAS",
          "DATA_TYPE",
          "TIMESTAMP_SECONDS",
          "TIMESTAMP_NANO_OFFSET",
          "QUALITY",
          "VALUE"
        ]
      }
    }
  }
}
```

```
    }  
  }  
},  
"jobCreationDate":1645745176.498,  
"jobStatus":"COMPLETED",  
"jobName":"myBulkImportJob",  
"jobLastUpdateDate":1645745279.968,  
"jobRoleArn":"arn:aws:iam::123456789012:role/DemoRole",  
"jobId":"f8c031d0-01d1-4b94-90b1-afe8bb93b7e5"  
}
```

Répertorier les tâches d'importation AWS IoT SiteWise en bloc (AWS CLI)

Utilisez l'opération [ListBulkImportJobs](#) API pour récupérer une liste de résumés pour les tâches d'importation en bloc dans AWS IoT SiteWise. Cette opération constitue un moyen efficace de surveiller et de gérer vos processus d'importation de données. Il renvoie les informations clés suivantes pour chaque tâche :

- Job ID. Un identifiant unique pour chaque tâche d'importation en masse
- Nom du job. Le nom que vous avez attribué à la tâche lors de sa création
- État actuel. État actuel de la tâche (par exemple, COMPLETED, RUNNING, FAILED)

ListBulkImportJobs est particulièrement utile pour obtenir une vue d'ensemble complète de toutes vos tâches d'importation en vrac. Cela peut vous aider à suivre plusieurs importations de données, à identifier les tâches nécessitant une attention particulière et à maintenir un flux de travail organisé. L'opération prend en charge la pagination, ce qui vous permet de récupérer efficacement un grand nombre de résumés de tâches. Vous pouvez utiliser la tâche IDs renvoyée par cette opération avec l'[DescribeBulkImportJob](#) opération pour récupérer des informations plus détaillées sur des tâches spécifiques. Ce processus en deux étapes vous permet d'abord d'obtenir une vue d'ensemble de tous les emplois, puis d'approfondir les détails des emplois qui vous intéressent. Lors de l'utilisation ListBulkImportJobs, vous pouvez appliquer des filtres pour affiner les résultats. Par exemple, vous pouvez filtrer les tâches en fonction de leur statut pour récupérer uniquement les tâches terminées ou uniquement les tâches en cours d'exécution. Cette fonctionnalité vous permet de vous concentrer sur les informations les plus pertinentes pour votre tâche en cours. L'opération renvoie également un nextToken si d'autres résultats sont disponibles. Vous pouvez utiliser ce jeton lors des appels suivants pour récupérer le prochain ensemble de résumés de tâches, ce qui vous permet de parcourir toutes vos tâches d'importation groupée, même si vous en avez un grand

nombre. L'exemple suivant montre comment utiliser `ListBulkImportJobs` le AWS CLI pour récupérer une liste des tâches terminées.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED
```

Exemple Filtre de réponse aux tâches terminées

```
{
  "jobSummaries":[
    {
      "id":"bdbbfa52-d775-4952-b816-13ba1c7cb9da",
      "name":"myBulkImportJob",
      "status":"COMPLETED"
    },
    {
      "id":"15ffc641-dbd8-40c6-9983-5cb3b0bc3e6b",
      "name":"myBulkImportJob2",
      "status":"COMPLETED"
    }
  ]
}
```

Cette commande montre comment récupérer une `ListBulkImportJobs` liste de tâches qui se sont terminées par des échecs. Le maximum est fixé à 50 résultats et nous utilisons le jeton suivant pour les résultats paginés.

```
aws iotsitewise list-bulk-import-jobs --filter COMPLETED_WITH_FAILURES --max-results 50
--next-token "string"
```

Ingérer des données pour AWS IoT SiteWise utiliser des règles AWS IoT Core

Envoyez des données AWS IoT SiteWise depuis des AWS IoT objets et d'autres AWS services à l'aide de règles dans AWS IoT Core. Les règles transforment les messages MQTT et exécutent des actions pour interagir avec les AWS services. L'action de AWS IoT SiteWise règle transmet les données des messages à l'[BatchPutAssetPropertyValue](#) opération depuis l' AWS IoT SiteWise API. Pour plus d'informations, consultez la section [Règles](#) et [AWS IoT SiteWise actions](#) du Guide du AWS IoT développeur.

Pour suivre un didacticiel décrivant les étapes nécessaires à la configuration d'une règle qui ingère des données par le biais des ombres des appareils, voir [Ingérez des données dans AWS IoT SiteWise des objets AWS IoT](#).

Vous pouvez également envoyer des données depuis AWS IoT SiteWise d'autres AWS services. Pour de plus amples informations, veuillez consulter [Interagissez avec d'autres AWS services](#).

Rubriques

- [Accordez AWS IoT l'accès requis](#)
- [Configurer l'action de la AWS IoT SiteWise règle](#)
- [Réduisez les coûts avec Basic Ingest in AWS IoT SiteWise](#)

Accordez AWS IoT l'accès requis

Vous utilisez les rôles IAM pour contrôler les AWS ressources auxquelles chaque règle a accès. Avant de créer une règle, vous devez créer un rôle IAM avec une politique qui permet à la règle d'effectuer des actions sur la AWS ressource requise. AWS IoT assume ce rôle lors de l'exécution d'une règle.

Si vous créez l'action de règle dans la AWS IoT console, vous pouvez choisir une ressource racine pour créer un rôle ayant accès à une hiérarchie de ressources sélectionnée. Pour plus d'informations sur la façon de définir manuellement un rôle pour une règle, consultez les sections [Accorder AWS IoT les autorisations d'accès et de transfert de rôle requises](#) dans le Guide du AWS IoT développeur.

Pour l'action de la AWS IoT SiteWise règle, vous devez définir un rôle qui autorise l'iotsitewise:BatchPutAssetPropertyValue accès aux propriétés des actifs auxquels la règle envoie des données. Pour améliorer la sécurité, vous pouvez spécifier un chemin hiérarchique des AWS IoT SiteWise actifs dans la Condition propriété.

L'exemple de stratégie d'approbation suivant permet d'accéder à une ressource spécifique et à ses enfants.

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

```

    "Resource": "*",
    "Condition": {
      "StringLike": {
        "iotsitewise:assetHierarchyPath": [
          "/root node asset ID",
          "/root node asset ID/*"
        ]
      }
    }
  }
]
}

```

ConditionSupprimez-le de la politique pour autoriser l'accès à tous vos actifs. L'exemple de stratégie d'approbation suivant permet d'accéder à toutes vos ressources dans la région active.

```

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

```

Configurer l'action de la AWS IoT SiteWise règle

L'action de AWS IoT SiteWise règle envoie les données du message MQTT à l'origine de la règle aux propriétés des actifs dans AWS IoT SiteWise. Vous pouvez télécharger plusieurs entrées de données sur différentes propriétés d'actifs en même temps, afin d'envoyer des mises à jour pour tous les capteurs d'un appareil en un seul message. Vous pouvez également charger plusieurs points de données à la fois pour chaque entrée de données.

Note

Lorsque vous envoyez des données à AWS IoT SiteWise avec l'action de règle, vos données doivent répondre à toutes les exigences de l'BatchPutAssetPropertyValueopération. Par exemple, vos données ne peuvent pas être horodatées de moins de 7 jours par rapport

à l'époque Unix actuelle. Pour plus d'informations, consultez [Ingestion de données avec l'API AWS IoT SiteWise](#).

Pour chaque entrée de données de l'action de règle, vous identifiez une propriété de ressource et spécifiez l'horodatage, la qualité et la valeur de chaque point de données pour cette propriété. L'action de règle accepte des chaînes pour tous les paramètres.

Pour identifier une propriété de ressource, vous pouvez spécifier l'une des options suivantes :

- ID de ressource (assetId) et ID de propriété (propertyId) pour la propriété de ressource à laquelle vous envoyez des données. Vous pouvez trouver l'ID d'actif et l'ID de propriété à l'aide du Console AWS IoT SiteWise. Si vous connaissez l'identifiant de l'actif, vous pouvez utiliser le AWS CLI [DescribeAsset](#) to call pour trouver l'identifiant de la propriété.
- Alias de propriété (propertyAlias), qui est un alias de flux de données (par exemple, / company/windfarm/3/turbine/7/temperature). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour découvrir comment définir des alias de propriété de ressource, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Pour l'horodatage de chaque entrée, utilisez l'horodatage indiqué par votre équipement ou l'horodatage fourni par. AWS IoT Core L'horodatage comporte deux paramètres :

- Durée en secondes (timeInSeconds) — Durée de l'époque Unix, en secondes, à laquelle le capteur ou l'équipement a rapporté les données.
- Décalage en nanos (offsetInNanos) — (Facultatif) Le décalage de nanosecondes par rapport au temps en secondes.

Important

Si votre horodatage est une chaîne, comporte une partie décimale ou n'est pas exprimé en secondes, AWS IoT SiteWise rejette la demande. Vous devez convertir l'horodatage en secondes et en décalage de nanosecondes. Utilisez les fonctionnalités du moteur de AWS IoT règles pour convertir l'horodatage. Pour plus d'informations, consultez les ressources suivantes :

- [Obtenir des horodatages pour les appareils qui n'indiquent pas l'heure exacte](#)

- [Conversion des horodatages au format chaîne](#)

Vous pouvez utiliser des modèles de substitution pour plusieurs paramètres de l'action afin d'effectuer des calculs, d'invoquer des fonctions et d'extraire des valeurs de la charge utile du message. Pour plus d'informations, consultez la section [Modèles de substitution](#) dans le Guide du AWS IoT développeur.

Note

Comme une expression dans un modèle de substitution est évaluée séparément de l'instruction SELECT, vous ne pouvez pas utiliser un modèle de substitution pour référencer un alias créé à l'aide d'une clause AS. Vous pouvez référencer uniquement les informations présentes dans la charge utile d'origine, en plus des fonctions et opérateurs pris en charge.

Rubriques

- [Obtenir des horodatages pour les appareils qui n'indiquent pas l'heure exacte](#)
- [Conversion des horodatages au format chaîne](#)
- [Conversion de chaînes d'horodatage d'une précision de la nanoseconde](#)
- [Exemples de configurations de règles](#)
- [Dépannage de l'action de règle](#)

Obtenir des horodatages pour les appareils qui n'indiquent pas l'heure exacte

Si votre capteur ou équipement ne fournit pas de données temporelles précises, obtenez l'heure actuelle de l'époque Unix à partir du moteur de AWS IoT règles avec [timestamp](#) (). Cette fonction affiche le temps en millisecondes. Vous devez donc convertir la valeur en temps en secondes et en décalage en nanosecondes. Pour ce faire, utilisez les conversions suivantes :

- Pour Time in seconds (Délai en secondes) (timeInSeconds), utilisez `${floor(timestamp() / 1E3)}` pour convertir le temps de millisecondes en secondes.
- Pour Offset in nanos (Décalage en nanosecondes) (offsetInNanos), utilisez `${(timestamp() % 1E3) * 1E6}` pour calculer le décalage en nanosecondes de l'horodatage.

Conversion des horodatages au format chaîne

Si votre capteur ou équipement indique les données temporelles sous forme de chaîne (par exemple, `2020-03-03T14:57:14.699Z`), utilisez [time_to_epoch](#) (String, String). Cette fonction saisit l'horodatage et le modèle de format sous forme de paramètres et affiche le temps en millisecondes. Ensuite, vous devez convertir le temps en temps en secondes et en décalage en nanosecondes. Pour ce faire, utilisez les conversions suivantes :

- Pour Time in seconds (`timeInSeconds`), utilisez cette option `${floor(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") / 1E3)}` pour convertir la chaîne d'horodatage en millisecondes, puis en secondes.
- Pour Offset in nanos (`offsetInNanos`), utilisez `${(time_to_epoch("2020-03-03T14:57:14.699Z", "yyyy-MM-dd'T'HH:mm:ss'Z'") % 1E3) * 1E6}` pour calculer le décalage en nanosecondes de la chaîne d'horodatage.

Note

La `time_to_epoch` fonction prend en charge les chaînes d'horodatage d'une précision maximale de quelques millisecondes. Pour convertir des chaînes avec une précision de la microseconde ou de la nanoseconde, configurez une AWS Lambda fonction appelée par votre règle pour convertir l'horodatage en valeurs numériques. Pour de plus amples informations, veuillez consulter [Conversion de chaînes d'horodatage d'une précision de la nanoseconde](#).

Conversion de chaînes d'horodatage d'une précision de la nanoseconde

Si votre appareil envoie des informations d'horodatage sous forme de chaîne avec une précision de la nanoseconde (par exemple, `2020-03-03T14:57:14.699728491Z`), suivez la procédure suivante pour configurer l'action de votre règle. Vous pouvez créer une AWS Lambda fonction qui convertit l'horodatage d'une chaîne en heure en secondes (`timeInSeconds`) et en décalage en nanos (`offsetInNanos`). Utilisez ensuite [aws_lambda \(FunctionArn, InputJson\) dans les paramètres d'action de votre règle pour appeler cette](#) fonction Lambda et utiliser la sortie de votre règle.

 Note

Cette section contient des instructions avancées qui supposent que vous êtes familier avec la création des ressources suivantes :

- Fonctions Lambda. Pour plus d'informations, consultez la section [Création de votre première fonction Lambda](#) dans le Guide du AWS Lambda développeur.
- AWS IoT règles avec l'action de la AWS IoT SiteWise règle. Pour de plus amples informations, veuillez consulter [Ingérer des données pour AWS IoT SiteWise utiliser des règles AWS IoT Core](#).

Pour créer une action de AWS IoT SiteWise règle qui analyse les chaînes d'horodatage

1. Créez une fonction Lambda avec les propriétés suivantes :

- Nom de la fonction : utilisez un nom de fonction descriptif (par exemple, **ConvertNanosecondTimestampFromString**).
- Runtime — Utilisez un environnement d'exécution Python 3, tel que Python 3.11 (python3.11).
- Autorisations — Créez un rôle avec des autorisations Lambda de base () AWS LambdaBasicExecutionRole.
- Couches — Ajoutez la couche AWS SDKPandas-Python311 à utiliser par la fonction Lambda. numpy
- Code de fonction — Utilisez le code de fonction suivant, qui utilise un argument de chaîne nommé timestamp timeInSeconds et génère des offsetInNanos valeurs pour cet horodatage.

```
import json
import math
import numpy

# Converts a timestamp string into timeInSeconds and offsetInNanos in Unix epoch
time.
# The input timestamp string can have up to nanosecond precision.
def lambda_handler(event, context):
    timestamp_str = event['timestamp']
    # Parse the timestamp string as nanoseconds since Unix epoch.
```

```
nanoseconds = numpy.datetime64(timestamp_str, 'ns').item()
time_in_seconds = math.floor(nanoseconds / 1E9)
# Slice to avoid precision issues.
offset_in_nanos = int(str(nanoseconds)[-9:])
return {
    'timeInSeconds': time_in_seconds,
    'offsetInNanos': offset_in_nanos
}
```

Cette fonction Lambda saisit des chaînes d'horodatage au format ISO 8601 en utilisant `datetime64` from NumPy

 Note

Si vos chaînes d'horodatage ne sont pas au format ISO 8601, vous pouvez implémenter une solution avec pandas qui définit le format d'horodatage. Pour plus d'informations, consultez [pandas.to_datetime](https://pandas.pydata.org/pandas-docs/stable/timeseries.html#timeseries-timestamp-convert).

2. Lorsque vous configurez l' AWS IoT SiteWise action pour votre règle, utilisez les modèles de substitution suivants pour le temps en secondes (`timeInSeconds`) et le décalage en nanos (`offsetInNanos`). Ces modèles de substitution supposent que votre charge utile de message contient la chaîne d'horodatage dans `timestamp`. La fonction `aws_lambda` utilise une structure JSON pour son second paramètre, de sorte que vous pouvez modifier les modèles de substitution ci-dessous si nécessaire.

- Pour Time in seconds (Temps en secondes) (`timeInSeconds`), utilisez le modèle de substitution suivant.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).timeInSeconds}
```

- Pour Offset in nanos (Décalage en nanos) (`offsetInNanos`), utilisez le modèle de substitution suivant.

```
${aws_lambda('arn:aws:lambda:region:account-id:function:ConvertNanosecondTimestampFromString', {'timestamp': timestamp}).offsetInNanos}
```

Pour chaque paramètre, remplacez *region* et *account-id* par votre région et votre numéro de AWS compte. Si vous avez utilisé un autre nom pour votre fonction Lambda, modifiez-le également.

3. Accordez AWS IoT les autorisations nécessaires pour appeler votre fonction avec `lambda:InvokeFunction` cette autorisation. Pour de plus amples informations, veuillez consulter [aws_lambda\(functionArn, inputJson\)](#).
4. Testez votre règle (par exemple, utilisez le client de test AWS IoT MQTT) et vérifiez qu'il AWS IoT SiteWise reçoit les données que vous envoyez.

Si votre règle ne fonctionne pas comme prévu, veuillez consulter [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#).

Note

Cette solution appelle la fonction Lambda deux fois pour chaque chaîne d'horodatage. Vous pouvez créer une autre règle pour réduire le nombre d'appels de fonctions Lambda si votre règle gère plusieurs points de données ayant le même horodatage dans chaque charge utile. Pour ce faire, créez une règle avec une action de republication qui invoque le Lambda et publie la charge utile d'origine avec la chaîne d'horodatage convertie en `et.timeInSeconds` et `offsetInNanos`. Créez ensuite une règle avec une action de AWS IoT SiteWise règle pour consommer la charge utile convertie. Avec cette approche, vous réduisez le nombre de fois que la règle invoque le Lambda, mais vous augmentez le nombre AWS IoT d'actions de règle exécutées. Pensez à la tarification de chaque service si vous appliquez cette solution à votre cas d'utilisation.

Exemples de configurations de règles

Cette section contient des exemples de configurations de règles permettant de créer une règle avec une AWS IoT SiteWise action.

Exemple Exemple d'action de règle qui utilise des alias de propriété comme rubriques de message

L'exemple suivant crée une règle avec une AWS IoT SiteWise action qui utilise le topic (via [topic \(\)](#)) comme alias de propriété pour identifier les propriétés des actifs. Utilisez cet exemple pour définir une règle pour l'ingestion de données de type double pour toutes les éoliennes de tous les parcs éoliens.

Cet exemple nécessite que vous définissiez des alias de propriété sur les propriétés de tous les actifs de turbine. Vous devez définir une deuxième règle similaire pour ingérer des données de type entier.

```
aws iot create-topic-rule \  
  --rule-name SiteWiseWindFarmRule \  
  --topic-rule-payload file://sitewise-rule-payload.json
```

L'exemple de charge utile dans `sitewise-rule-payload.json` comporte le contenu suivant.

```
{  
  "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+' WHERE type = 'double'",  
  "description": "Sends data to the wind turbine asset property with the same alias as  
the topic",  
  "ruleDisabled": false,  
  "awsIotSqlVersion": "2016-03-23",  
  "actions": [  
    {  
      "iotSiteWise": {  
        "putAssetPropertyValueEntries": [  
          {  
            "propertyAlias": "${topic()}",  
            "propertyValues": [  
              {  
                "timestamp": {  
                  "timeInSeconds": "${timeInSeconds}"  
                },  
                "value": {  
                  "doubleValue": "${value}"  
                }  
              }  
            ]  
          }  
        ],  
        "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"  
      }  
    }  
  ]  
}
```

Avec cette action de règle, envoyez le message suivant à un alias de propriété d'éolienne (par exemple, `/company/windfarm/3/turbine/7/temperature`) en tant que sujet pour ingérer des données.

```
{
  "type": "double",
  "value": "38.3",
  "timeInSeconds": "1581368533"
}
```

Exemple Exemple d'action de règle qui utilise `timestamp()` pour déterminer l'heure

L'exemple suivant crée une règle avec une AWS IoT SiteWise action qui identifie une propriété d'actif par IDs et utilise [timestamp\(\)](#) pour déterminer l'heure actuelle.

```
aws iot create-topic-rule \
  --rule-name SiteWiseAssetPropertyRule \
  --topic-rule-payload file://sitewise-rule-payload.json
```

L'exemple de charge utile dans `sitewise-rule-payload.json` comporte le contenu suivant.

```
{
  "sql": "SELECT * FROM 'my/asset/property/topic'",
  "description": "Sends device data to an asset property",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "iotSiteWise": {
        "putAssetPropertyValueEntries": [
          {
            "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
            "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
            "propertyValues": [
              {
                "timestamp": {
                  "timeInSeconds": "${floor(timestamp() / 1E3)}",
                  "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
                },
                "value": {
                  "doubleValue": "${value}"
                }
              }
            ]
          }
        ]
      }
    }
  ],
}
```

```
    "roleArn": "arn:aws:iam::account-id:role/MySiteWiseActionRole"
  }
}
]
```

Avec cette action de règle, envoyez le message suivant au pour `my/asset/property/topic` ingérer des données.

```
{
  "type": "double",
  "value": "38.3"
}
```

Dépannage de l'action de règle

Pour résoudre les problèmes liés à votre action de AWS IoT SiteWise règle dans AWS IoT Core, configurez CloudWatch les journaux ou configurez une action d'erreur de republication pour votre règle. Pour de plus amples informations, veuillez consulter [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#).

Réduisez les coûts avec Basic Ingest in AWS IoT SiteWise

AWS IoT Core [fournit une fonctionnalité appelée Basic Ingest que vous pouvez utiliser pour envoyer des données AWS IoT Core sans encourir AWS IoT de frais de messagerie](#). Cette fonctionnalité optimise le flux de données pour les charges de travail relatives à l'ingestion de grands volumes de données en supprimant l'agent de messages de publication et d'abonnement dans le circuit d'ingestion. Vous pouvez utiliser l'ingestion de base si vous connaissez les règles vers lesquelles vos messages doivent être acheminés.

Pour utiliser l'ingestion de base, vous envoyez des messages directement à une règle spécifique à l'aide d'une rubrique spéciale, `$aws/rules/rule-name`. Par exemple, pour envoyer un message à une règle nommée `SiteWiseWindFarmRule`, vous envoyez un message à la rubrique `$aws/rules/SiteWiseWindFarmRule`.

Si votre action de règle utilise des modèles de substitution contenant [topic\(Decimal\)](#), vous pouvez transmettre la rubrique d'origine à la fin de la rubrique spéciale d'ingestion de base, par exemple `$aws/rules/rule-name/original-topic`. Par exemple, pour utiliser l'ingestion de base avec l'exemple d'alias de propriété de parc éolien de la section précédente, vous pouvez envoyer des messages à la rubrique suivante.


```
$aws/rules/SiteWiseWindFarmRule//company/windfarm/3/turbine/7/temperature
```

Note

L'exemple ci-dessus inclut une deuxième barre oblique (//) car AWS IoT le préfixe Basic Ingest (\$aws/rules/*rule-name*/) est supprimé de la rubrique visible par l'action de la règle. Dans cet exemple, la règle reçoit la rubrique /company/windfarm/3/turbine/7/temperature.

Pour plus d'informations, consultez [la section Réduction des coûts de messagerie grâce à l'ingestion de base](#) dans le Guide du AWS IoT développeur.

Ingérer des données vers AWS IoT SiteWiseAWS IoT Events

Vous pouvez ainsi créer des applications complexes de surveillance des événements pour votre flotte IoT dans le AWS cloud. AWS IoT Events Utilisez l' SiteWise action IoT AWS IoT Events pour envoyer des données aux propriétés des actifs AWS IoT SiteWise lorsqu'un événement se produit.

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 Events est conçu pour rationaliser le développement d'applications de surveillance des événements pour les appareils et systèmes IoT dans le AWS cloud. En utilisant AWS IoT Events, vous pouvez :

- Détectez les changements, les anomalies ou les conditions spécifiques au sein de votre flotte IoT et répondez-y.
- Améliorez votre efficacité opérationnelle et permettez une gestion proactive de votre écosystème IoT.

En l'intégrant AWS IoT SiteWise via l' AWS IoT SiteWise action, il AWS IoT Events étend ses fonctionnalités, vous permettant de mettre à jour automatiquement les propriétés des actifs AWS IoT

SiteWise en réponse à des événements spécifiques. Cette interaction peut simplifier l'ingestion et la gestion des données. Il peut également vous fournir des informations exploitables.

Pour plus d'informations, consultez les rubriques suivantes du guide du AWS IoT Events développeur :

- [Qu'est-ce que c'est AWS IoT Events ?](#)
- [Actions AWS IoT Events](#)
- [SiteWise Action dans le domaine de l'IoT](#)

Utiliser le gestionnaire de AWS IoT Greengrass flux dans AWS IoT SiteWise

AWS IoT Greengrass le gestionnaire de flux est une fonctionnalité d'intégration qui facilite le transfert de flux de données depuis des sources locales vers le AWS Cloud. Il agit comme une couche intermédiaire qui gère les flux de données, permettant aux appareils fonctionnant en périphérie de collecter et de stocker les données avant qu'elles ne soient envoyées AWS IoT SiteWise, pour une analyse et un traitement plus approfondis.

Ajoutez une destination de données en configurant une source locale sur la AWS IoT SiteWise console. Vous pouvez également utiliser le gestionnaire de flux dans votre AWS IoT Greengrass solution personnalisée pour ingérer AWS IoT SiteWise des données.

Note

Pour ingérer des données provenant de sources OPC UA, configurez une passerelle AWS IoT SiteWise Edge qui s'exécute sur. AWS IoT Greengrass Pour de plus amples informations, veuillez consulter [Utiliser les passerelles AWS IoT SiteWise Edge](#).

Pour plus d'informations sur la configuration d'une destination pour les données source locales, consultez [Comprendre les destinations AWS IoT SiteWise Edge](#).

Pour plus d'informations sur la façon d'ingérer des données à l'aide du gestionnaire de flux dans une AWS IoT Greengrass solution personnalisée, consultez les rubriques suivantes du guide du AWS IoT Greengrass Version 2 développeur :

- [Qu'est-ce que c'est AWS IoT Greengrass ?](#)

- [Gérez les flux de données sur le AWS IoT Greengrass cœur](#)
- [Exportation de données vers les propriétés des AWS IoT SiteWise actifs](#)

Utiliser les passerelles AWS IoT SiteWise Edge

AWS IoT SiteWise Edge étend les fonctionnalités du cloud aux environnements industriels de pointe, permettant le traitement des données, l'analyse et la prise de décision au niveau local. SiteWise Edge s'intègre à AWS IoT SiteWise d'autres AWS services pour fournir des solutions complètes d'IoT industriel. Les passerelles servent d'intermédiaire entre votre équipement industriel et AWS IoT SiteWise.

SiteWise Les passerelles Edge s'exécutent sur deux cibles de déploiement différentes :

- AWS IoT Greengrass V2
- Siemens Industrial Edge

Vous pouvez utiliser une passerelle SiteWise Edge pour collecter des données en périphérie et les publier dans le cloud. Pour les passerelles exécutées AWS IoT Greengrass, vous pouvez également traiter les données à la périphérie à l'aide de modèles d'actifs et d'actifs.

L'application AWS IoT SiteWise Edge Siemens Industrial Edge permet l'intégration entre les équipements industriels et vous permet AWS IoT SiteWise d'agréger et de traiter les données brutes des machines et d'exécuter des analyses localement avant d'envoyer des données raffinées vers le AWS Cloud.

Concepts clés des passerelles SiteWise Edge

SiteWise Edge possède plusieurs fonctionnalités utiles pour l'informatique de pointe dans les environnements industriels.

Collecte et traitement des données au niveau local

Prend en charge la collecte de données à partir d'actifs industriels à l'aide de protocoles tels que OPC-UA et MQTT. Les passerelles s'exécutent sur des appareils AWS IoT Greengrass Core ou Siemens Industrial Edge.

Fonctionnement hors ligne

Continue de collecter et de traiter les données en cas de panne d'Internet, en se synchronisant avec le cloud lorsque la connectivité est rétablie.

Informatique de pointe avec AWS IoT Greengrass composants

Utilisez l'éditeur IoT SiteWise pour transférer les données vers le cloud et le AWS IoT SiteWise processeur à des fins de transformations et de calculs locaux. L'éditeur et le processeur sont tous deux AWS IoT Greengrass V2 des composants. Pour plus d'informations sur AWS IoT Greengrass les composants, consultez la section [Composants AWS fournis](#).

Intégration avec AWS IoT SiteWise pour étendre les fonctionnalités du cloud

Fonctionne avec les fonctionnalités du AWS IoT SiteWise cloud, en étendant les modèles d'actifs, les analyses et les tableaux de bord jusqu'à la périphérie.

Pour les passerelles sur lesquelles un pack de traitement de données est activé, vous pouvez utiliser AWS OpsHub AWS IoT SiteWise pour gérer de manière centralisée vos passerelles SiteWise Edge. AWS OpsHub fournit des fonctionnalités de gestion et de surveillance à distance. Pour de plus amples informations, veuillez consulter [Gérez les passerelles SiteWise Edge à l'aide de AWS OpsHubAWS IoT SiteWise](#).

Intégration des sources de données des partenaires

Connectez une source de données partenaire à votre passerelle et recevez les données du partenaire sur votre passerelle SiteWise Edge et dans le AWS cloud. Pour de plus amples informations, veuillez consulter [Sources de données partenaires sur les passerelles SiteWise Edge](#).

Visualisation locale sur le bord

Fournit des tableaux de bord personnalisés pour obtenir des informations en temps réel à la périphérie.

Surveillez les données localement dans votre établissement à l'aide des portails de SiteWise surveillance installés sur vos appareils locaux. Pour de plus amples informations, veuillez consulter [Activation de votre AWS IoT SiteWise portail à la périphérie](#).

Avantages de la mise en œuvre d' SiteWise Edge

SiteWise Edge offre de nombreux avantages qui peuvent améliorer de manière significative les opérations industrielles et les processus décisionnels.

- Informations opérationnelles en temps réel sans délais de traitement dans le cloud

- Continuité opérationnelle dans des environnements déconnectés
- Réduction des coûts de bande passante et de stockage grâce au prétraitement périphérique
- Fiabilité accrue grâce à la capacité de prendre des décisions locales basées sur les données

Hébergez vous-même une passerelle AWS IoT SiteWise Edge avec AWS IoT Greengrass V2

Configurez AWS IoT SiteWise Edge pour collecter, traiter et visualiser les données des équipements industriels localement avant de les envoyer vers le cloud. Auto-hébergeur en utilisant AWS IoT Greengrass Version 2.

Une passerelle AWS IoT SiteWise Edge sert d'intermédiaire entre votre équipement industriel et AWS IoT SiteWise. Lorsqu'elle est exécutée AWS IoT Greengrass Version 2, la passerelle SiteWise Edge prend en charge la collecte et le traitement des données sur site. Surveillez les données localement au sein de votre établissement via des portails de SiteWise surveillance sur vos appareils locaux avec le pack de traitement des données activé et AWS OpsHub installé.

Il existe deux types de passerelles auto-hébergées :

Passerelle V3 compatible MQTT

L'architecture de passerelle V3 compatible MQTT fournit des capacités d'ingestion de données améliorées. Il utilise le protocole MQTT pour une communication de données efficace et propose des destinations de données configurables. Il s'agit notamment des options d'ingestion de données en mémoire tampon à l'aide d'Amazon S3, ainsi que d'ingestion de données en temps réel. Vous pouvez implémenter des filtres de chemin pour vous abonner à des sujets MQTT spécifiques, permettant ainsi une collecte de données ciblée. Notez que la passerelle V3 compatible MQTT ne prend pas en charge la fonctionnalité Data Processing Pack. Pour de plus amples informations, veuillez consulter [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#).

Streams classiques, passerelle V2

La passerelle Classic Streams, V2 représente l'architecture de passerelle AWS IoT SiteWise Edge traditionnelle. Il convient parfaitement aux déploiements SiteWise Edge existants et aux utilisateurs habitués au flux de travail établi. Bien que la passerelle Classic diffuse, la passerelle V2 prend en charge le pack de traitement de données, notez que les données générées par

le pack de traitement de données ne peuvent pas être ingérées via Amazon S3. Utilisez la passerelle Classic Streams, V2 si vous devez maintenir la compatibilité avec les déploiements existants ou si vous avez besoin de la fonctionnalité du pack de traitement des données. Pour de plus amples informations, veuillez consulter [Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise](#).

Rubriques

- [AWS IoT SiteWise Exigences relatives à la passerelle auto-hébergée Edge](#)
- [Créez une passerelle SiteWise Edge auto-hébergée](#)
- [Installez le logiciel de passerelle AWS IoT SiteWise Edge sur votre appareil local](#)
- [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#)
- [Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise](#)
- [Ajoutez des sources de données à votre passerelle AWS IoT SiteWise Edge](#)
- [AWS IoT Greengrass composants pour AWS IoT SiteWise Edge](#)
- [Filtrer les actifs sur une passerelle SiteWise Edge](#)
- [Configuration de la prise en charge des proxys et gestion des magasins de confiance pour AWS IoT SiteWise Edge](#)
- [Utiliser AWS IoT SiteWise APIs sur le bord](#)

AWS IoT SiteWise Exigences relatives à la passerelle auto-hébergée Edge

AWS IoT SiteWise Les passerelles Edge fonctionnent AWS IoT Greengrass V2 sous la forme d'un ensemble de AWS IoT Greengrass composants qui prennent en charge la collecte, le traitement et la publication de données sur site. Pour configurer une passerelle SiteWise Edge qui s'exécute sur AWS IoT Greengrass V2, créez une passerelle dans le logiciel SiteWise Edge Gateway AWS Cloud et exécutez le logiciel de passerelle Edge pour configurer votre appareil local. Lorsque vous utilisez le AWS Management Console pour créer la passerelle SiteWise Edge, un script d'installation est fourni. Exécutez ce script sur votre passerelle cible pour configurer les logiciels et les dépendances nécessaires.

Exigences relatives aux appareils locaux

Les appareils locaux doivent répondre aux exigences suivantes pour installer et exécuter le logiciel de passerelle SiteWise Edge.

- Supporte la version [v2.3.0](#) ou ultérieure du logiciel AWS IoT Greengrass V2 Core. Pour plus d'informations, consultez la section [Exigences](#) du guide du AWS IoT Greengrass Version 2 développeur.

- L'une des plateformes prises en charge suivantes :

- Système d'exploitation : Ubuntu 20.04 ou version ultérieure

Architecture : x86_64 (AMD64) ou ARMv8 (Aarch64)

- Système d'exploitation : Red Hat Enterprise Linux (RHEL) 8

Architecture : x86_64 (AMD64) ou ARMv8 (Aarch64)

- Système d'exploitation : Amazon Linux 2

Architecture : x86_64 (AMD64) ou ARMv8 (Aarch64)

- Système d'exploitation : Debian 11

Architecture : x86_64 (AMD64) ou ARMv8 (Aarch64)

- Système d'exploitation : Windows Server 2019 et versions ultérieures

Architecture : x86_64 (AMD64)

Note

Les plateformes ARM prennent en charge les passerelles SiteWise Edge uniquement avec le pack de collecte de données. Le pack de traitement des données n'est pas pris en charge.

- Minimum de 4 Go de RAM.
- Au moins 10 Go d'espace disque disponible pour le logiciel de passerelle SiteWise Edge.
- Configurez votre appareil local pour vous assurer que les ports appropriés sont accessibles. Pour obtenir la liste complète des points de terminaison de service sortants requis, voir Points de terminaison de [service requis pour les AWS IoT SiteWise passerelles](#) Edge.
- Java Runtime Environment (JRE) version 11 ou supérieure. Java doit être disponible sur la variable d'PATH environnement de l'appareil. Pour utiliser Java pour développer des composants personnalisés, vous devez installer un kit de développement Java (JDK). [Nous vous recommandons d'utiliser Amazon Corretto ou OpenJDK.](#)

Buckets Amazon S3 à autoriser pour les appareils locaux

Configurez votre appareil local pour fournir un accès au pare-feu au compartiment Amazon S3 suivant. Configurez l'accès en fonction des régions respectives de vos appareils.

Région	Point de terminaison
Asie-Pacifique (Tokyo)	https://iot-sitewise-gateway-ap-northeast-1-785558802005.s3.ap-northeast-1.amazonaws.com
Asie-Pacifique (Séoul)	https://iot-sitewise-gateway-ap-northeast-2-310055672453.s3.ap-northeast-2.amazonaws.com
Asie-Pacifique (Mumbai)	https://iot-sitewise-gateway-ap-south-1-677656657204.s3.ap-south-1.amazonaws.com
Asie-Pacifique (Singapour)	https://iot-sitewise-gateway-ap-southeast-1-475191558554.s3.ap-southeast-1.amazonaws.com
Asie-Pacifique (Sydney)	https://iot-sitewise-gateway-ap-southeast-2-396319432685.s3.ap-southeast-2.amazonaws.com
Canada (Centre)	https://iot-sitewise-gateway-ca-central-1-842060018567.s3.ca-central-1.amazonaws.com
Chine (Beijing)	https://iot-sitewise-gateway-cn-north-1-237124890262.s3.cn-north-1.amazonaws.com.cn
Europe (Francfort)	https://iot-sitewise-gateway-eu-central-1-748875242063.s3.eu-central-1.amazonaws.com
Europe (Irlande)	https://iot-sitewise-gateway-eu-west-1-383414315062.s3.eu-west-1.amazonaws.com
USA Est (Virginie du Nord)	https://iot-sitewise-gateway-us-east-1-223558168232.s3.us-east-1.amazonaws.com et https://iot-sitewise-gateway-us-east-1-223558168232.s3.amazonaws.com/
USA Est (Ohio)	https://iot-sitewise-gateway-us-east-2-005072661813.s3.us-east-2.amazonaws.com

Région	Point de terminaison
AWS GovCloud (US-Ouest)	https://iot-sitewise-gateway-us-gov-west-1-599984565679.s3.us-gov-west-1.amazonaws.com/
USA Ouest (Oregon)	https://iot-sitewise-gateway-us-west-2-502577205460.s3.us-west-2.amazonaws.com

Exigences relatives aux packs de traitement des données

- Si vous prévoyez d'utiliser le pack de traitement des données en périphérie AWS IoT SiteWise, votre appareil local doit également répondre aux exigences suivantes :
 - Dispose d'un processeur quadricœur x86 64 bits.
 - Dispose d'au moins 16 Go de RAM.
 - Dispose d'au moins 32 Go de RAM en cas d'utilisation Microsoft Windows.
 - Disposait d'au moins 256 Go d'espace disque libre.
 - Le périphérique local doit autoriser le trafic réseau entrant sur le port 443.
 - Les ports suivants sont réservés pour être utilisés par AWS IoT SiteWise : 80, 443, 3001, 4569, 4572, 8000, 8081, 8082, 8084, 8085, 8445, 8086, 9000, 9500, 11080 et 50010. L'utilisation d'un port réservé pour le trafic peut entraîner l'interruption de la connexion.

Note

Le composant AWS IoT Greengrass V2 Stream manager a ses propres exigences. Pour plus d'informations, consultez [la section Configuration](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

- Les exigences minimales en matière d'espace disque et de capacité de calcul dépendent de divers facteurs propres à votre implémentation et à votre cas d'utilisation.
 - L'espace disque requis pour la mise en cache des données pour une connectivité Internet intermittente dépend des facteurs suivants :
 - Nombre de flux de données chargés
 - Points de données par flux de données par seconde
 - Taille de chaque point de données

- Vitesses de communication
- Temps d'arrêt du réseau attendu
- La capacité de calcul requise pour interroger et charger les données dépend des facteurs suivants :
 - Nombre de flux de données chargés
 - Points de données par flux de données par seconde

Configurer les autorisations pour utiliser les passerelles SiteWise Edge

Vous devez disposer des autorisations suivantes pour utiliser les passerelles SiteWise Edge :

Note

Si vous utilisez la AWS IoT SiteWise console pour créer votre passerelle SiteWise Edge, ces autorisations sont ajoutées pour vous.

- Le rôle IAM de votre passerelle SiteWise Edge doit vous permettre d'utiliser une passerelle SiteWise Edge sur un AWS IoT Greengrass V2 appareil pour traiter les données des modèles d'actifs et les données des actifs.

Le rôle permet au service suivant d'assumer le rôle `:credentials.iot.amazonaws.com`.

Détails de l'autorisation

Le rôle doit disposer des autorisations suivantes :

- `iotsitewise`— Permet aux principaux de récupérer les données des modèles d'actifs et les données des actifs à la périphérie.
- `iot`— Permet à vos AWS IoT Greengrass V2 appareils d'interagir avec AWS IoT.
- `logs`— Permet à vos AWS IoT Greengrass V2 appareils d'envoyer des journaux à Amazon CloudWatch Logs.
- `s3`— Permet à vos AWS IoT Greengrass V2 appareils de télécharger des artefacts de composants personnalisés depuis Amazon S3.

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

```

```

    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:List*",
        "iotsitewise:Describe*",
        "iotsitewise:Get*"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeCertificate",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "logs:DescribeLogStreams",
        "s3:GetBucketLocation",
        "s3:GetObject",
        "iot:Connect",
        "iot:Publish",
        "iot:Subscribe",
        "iot:Receive",
        "iot:DescribeEndpoint"
      ],
      "Resource": "*"
    }
  ]
}
```

Créez une passerelle SiteWise Edge auto-hébergée

Utilisez la AWS IoT SiteWise console ou AWS CLI créez une passerelle SiteWise Edge auto-hébergée. Cette procédure explique comment créer une passerelle SiteWise Edge auto-hébergée que vous installerez sur votre propre matériel. Pour plus d'informations sur la création d'une passerelle SiteWise Edge qui s'exécute sur Siemens Industrial Edge, consultez [Hébergez une passerelle SiteWise Edge sur Siemens Industrial Edge](#).

Création d'une passerelle SiteWise Edge

Console

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Cliquez sur Create gateway (Créer une passerelle).
4. Pour Choisir une cible de déploiement, choisissez une passerelle auto-hébergée.
5. Sélectionnez soit une passerelle MQTT compatible avec MQTT, soit des flux classiques, une passerelle V2. Pour plus d'informations sur chaque option, consultez [Hébergez vous-même une passerelle AWS IoT SiteWise Edge avec AWS IoT Greengrass V2](#). La passerelle V3 compatible MQTT est recommandée pour ses fonctionnalités prêtes pour l'avenir.
6. Dans la section Configuration de la passerelle, entrez le nom de votre passerelle SiteWise Edge ou utilisez le nom généré par AWS IoT SiteWise.
7. Sous le système d'exploitation de l'appareil Greengrass, sélectionnez le système d'exploitation de l'appareil sur lequel vous allez installer cette passerelle SiteWise Edge.

Note

Le pack de traitement des données n'est disponible que sur les plateformes x86. Il n'est disponible que sur les streams classiques, passerelle V2

8. (Facultatif) Pour traiter et organiser les données à la périphérie, sous Fonctionnalités de la périphérie, sélectionnez Data Processing Pack.

Note

Pour autoriser les groupes d'utilisateurs de votre annuaire d'entreprise à accéder à cette passerelle SiteWise Edge, voir [Configurer la fonctionnalité Edge dans SiteWise Edge](#)

9. (Facultatif) Dans la section Configuration avancée, procédez comme suit :
 - Pour l'appareil Greengrass Core, choisissez l'une des options suivantes :
 - Configuration par défaut : utilise AWS automatiquement les paramètres par défaut pour créer un appareil Greengrass Core dans. AWS IoT Greengrass V2

1. Entrez un nom pour le périphérique principal de Greengrass ou utilisez le nom généré par. AWS IoT SiteWise
- Configuration avancée — Choisissez cette option si vous souhaitez utiliser un appareil principal Greengrass existant ou en créer un manuellement.
 1. Choisissez un appareil Greengrass core ou choisissez Create Greengrass core device pour en créer un dans la console. AWS IoT Greengrass V2 Pour plus d'informations, consultez [Configuration des appareils Core AWS IoT Greengrass V2](#) dans le Guide du développeur AWS IoT Greengrass Version 2 .
10. Cliquez sur Create gateway (Créer une passerelle).
11. Dans la boîte de dialogue Generate SiteWise Edge Gateway installer, choisissez Generate and download. AWS IoT SiteWise génère automatiquement un programme d'installation que vous pouvez utiliser pour configurer votre appareil local.

 Important

Vous ne pouvez pas régénérer ce fichier. Assurez-vous d'enregistrer le fichier d'installation dans un emplacement sécurisé, car vous l'utiliserez ultérieurement.

AWS CLI

Pour créer une passerelle auto-hébergée à l'aide de AWS CLI, attribuez un nom à la passerelle, spécifiez la plate-forme et la version de la passerelle. Il existe de nombreuses autres options que vous pouvez spécifier lors de la création d'une passerelle. Pour plus d'informations, voir [create-gateway](#) dans le manuel de référence des AWS CLI commandes pour AWS IoT SiteWise

Pour utiliser cet exemple, remplacez les espaces réservés d'entrée utilisateur par vos propres informations.

```
aws iotsitewise create-gateway \  
  --gateway-name your-gateway-name \  
  --gateway-platform greengrassV2={coreDeviceThingName=your-core-device-thing-name, coreDeviceOperatingSystem=LINUX_AMD64} \  
  --gateway-version 3 \  
  [--cli-input-json your-configuration]
```

- gateway-name— Nom unique pour la passerelle.

- `gateway-platform`— Spécifie la configuration de la plateforme de passerelle. Pour les passerelles auto-hébergées, entrez `greengrassV2`. Pour plus d'informations, consultez la section [Options](#) de la section `create-gateway` de AWS CLI Command Reference for AWS IoT SiteWise.
- `coreDeviceThingName`— Le nom de l'objet AWS IoT pour votre appareil AWS IoT Greengrass V2 principal.
- `coreDeviceOperatingSystem`— Le système d'exploitation de l'appareil principal dans AWS IoT Greengrass V2. La spécification du système d'exploitation est requise `gateway-version 3` et ne s'applique pas à `gateway-version 2`. Les options incluent : `LINUX_AARCH64`, `LINUX_AMD64`, et `WINDOWS_AMD64`.
- `gateway-version`— La version de la passerelle.
 - Pour créer une passerelle V3 compatible MQTT, utilisez la version 3 de passerelle.
 - Pour créer une passerelle Classic Streams, version V2, 2 utilisez-la pour la version passerelle.
- `cli-input-json`— Un fichier JSON contenant les paramètres de requête.

Maintenant que vous avez créé la passerelle SiteWise Edge, [Installez le logiciel de passerelle AWS IoT SiteWise Edge sur votre appareil local](#).

Installez le logiciel de passerelle AWS IoT SiteWise Edge sur votre appareil local

Après avoir créé une passerelle AWS IoT SiteWise Edge, installez le logiciel de passerelle SiteWise Edge sur votre appareil local. Le logiciel de passerelle Edge peut être installé sur des appareils locaux sur lesquels des systèmes d'exploitation Linux ou Microsoft Windows serveur sont installés.

Important

Assurez-vous que votre appareil local est connecté à Internet.

Linux

La procédure suivante utilise SSH pour se connecter à votre appareil local. Vous pouvez également utiliser une clé USB ou d'autres outils pour transférer le fichier d'installation sur votre

appareil local. Si vous ne souhaitez pas utiliser SSH, passez à l'étape 2 : Installation du logiciel de passerelle SiteWise Edge ci-dessous.

Prérequis SSH

Avant de vous connecter à votre appareil via SSH, remplissez les conditions préalables suivantes.

- Linux et macOS : téléchargez et installez OpenSSH. Pour de plus amples informations, veuillez consulter <https://www.openssh.com>.

Étape 1 : Copiez le programme d'installation sur votre périphérique de passerelle SiteWise Edge

Les instructions suivantes expliquent comment vous connecter à votre appareil local à l'aide d'un client SSH.

1. Pour vous connecter à votre appareil, exécutez la commande suivante dans une fenêtre de terminal de votre ordinateur, en remplaçant *username* et *IP* par un nom d'utilisateur doté de privilèges élevés et d'une adresse IP.

```
ssh username@IP
```

2. Pour transférer le fichier d'installation AWS IoT SiteWise généré sur votre périphérique de passerelle SiteWise Edge, exécutez la commande suivante.

Note

- Remplacez-le *path-to-saved-installer* par le chemin sur votre ordinateur que vous avez utilisé pour enregistrer le fichier d'installation et le nom du fichier d'installation.
- Remplacez *IP-address* par l'adresse IP de votre appareil local.
- *directory-to-receive-installer* Remplacez-le par le chemin sur le périphérique local que vous utilisez pour recevoir le fichier d'installation.

```
scp path-to-saved-installer.sh user-name@IP-address:directory-to-receive-installer
```


Étape 2 : Installation du logiciel de passerelle SiteWise Edge

Dans les procédures suivantes, exécutez les commandes dans une fenêtre de terminal sur votre périphérique de passerelle SiteWise Edge.

1. Donnez au fichier d'installation l'autorisation d'exécution.

```
chmod +x path-to-installer.sh
```

2. Exécutez le programme d'installation.

```
sudo ./path-to-installer.sh
```

Windows Server

Prérequis

Vous devez remplir les conditions préalables suivantes pour installer le logiciel de passerelle SiteWise Edge :

- Microsoft WindowsServer 2019 ou version ultérieure installé
- Privilèges d'administrateur
- PowerShell version 5.1 ou ultérieure installée
- SiteWise Le programme d'installation de la passerelle Edge a été téléchargé sur le serveur Windows où il sera provisionné

Étape 1 : Exécuter en PowerShell tant qu'administrateur

1. Sur le serveur Windows sur lequel vous souhaitez installer la passerelle SiteWise Edge, connectez-vous en tant qu'administrateur.
2. Entrez PowerShell dans la barre de recherche Windows.
3. Dans les résultats de recherche, ouvrez le menu contextuel (clic droit) de l' PowerShell application Windows. Choisissez Exécuter en tant qu'administrateur.

Étape 2 : Installation du logiciel de passerelle SiteWise Edge

Exécutez les commandes suivantes dans une fenêtre de terminal sur votre périphérique SiteWise Edge Gateway.

1. Débloquez le programme d'installation de la passerelle SiteWise Edge.

```
unblock-file path-to-installer.ps1
```

2. Lancez le programme d'installation.

```
./path-to-installer.ps1
```

 Note

Si l'exécution du script est désactivée sur le système, remplacez la politique d'exécution du script par `RemoteSigned`.

```
Set-ExecutionPolicy RemoteSigned
```

L'étape suivante dépend du type de passerelle auto-hébergée dont vous avez besoin. Continuez vers [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#) ou [Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise](#).

Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise

AWS IoT SiteWise peut utiliser des passerelles V3 compatibles MQTT, ce qui représente une avancée significative dans l'architecture des passerelles Edge. Ce type de passerelle utilise le protocole MQTT (Message Queuing Telemetry Transport) pour la communication de données, offrant ainsi une flexibilité et une efficacité accrues dans les déploiements industriels de l'IoT.

La passerelle V3 compatible MQTT utilise MQTT pour le transfert de données, permettant ainsi un protocole réseau léger de publication et d'abonnement qui transporte efficacement les messages entre les appareils et le cloud. Vous pouvez configurer différentes destinations de données, notamment l'ingestion de données en temps réel directement dans Amazon S3 AWS IoT SiteWise et l'ingestion de données mise en mémoire tampon. Pour permettre une collecte de données précise, vous pouvez implémenter des filtres de chemin pour vous abonner à des sujets MQTT spécifiques.

Les passerelles V3 compatibles MQTT sont fournies avec une destination en temps réel préconfigurée avec des filtres définis sur « # » (tous les sujets), que vous pouvez personnaliser ou supprimer selon vos besoins. Pour rationaliser la gestion des données, une seule destination en temps réel peut exister dans chaque passerelle.

L'architecture compatible MQTT est très différente de la passerelle Classic Streams, V2. Alors que la V2 utilise une approche basée sur les flux, la V3 utilise MQTT, offrant des destinations de données et des options de filtrage plus configurables. Notez toutefois que la V3 ne prend pas en charge le pack de traitement de données, qui est disponible dans la version 2.

La passerelle V3 compatible MQTT offre plusieurs avantages :

- Évolutivité améliorée, grâce à la légèreté du MQTT, permettant une meilleure gestion de nombreux appareils et une meilleure transmission de données à haute fréquence.
- Contrôle des données amélioré grâce à des filtres de chemin, permettant une gestion granulaire de la collecte de données et réduisant le transfert et le traitement inutiles des données.
- Gestion flexible des données, permettant la configuration entre le traitement en temps réel et le stockage en mémoire tampon en fonction des besoins spécifiques.
- Harmonisation avec les normes de communication IoT modernes, ouvrant la voie à de futures améliorations et intégrations.

Envisagez d'adopter la passerelle V3 compatible MQTT pour les nouveaux déploiements, en particulier lorsque vous avez besoin d'options d'ingestion de données flexibles et d'un contrôle précis de la collecte de données.

Note

Pour les déploiements existants ou les scénarios nécessitant le pack de traitement des données, la passerelle Classic Streams, V2 reste une option viable.

En proposant les deux types de passerelles AWS IoT SiteWise , vous pouvez choisir la solution qui répond le mieux à vos besoins spécifiques en matière d'IoT industriel, que vous privilégiez les fonctionnalités MQTT avancées ou la compatibilité avec les systèmes existants.

Filtres de destinations et de chemins

Consultez les rubriques suivantes pour en savoir plus sur les destinations et les filtres de chemin dans les passerelles compatibles MQTT :

- [Comprendre les destinations AWS IoT SiteWise Edge](#)
- [Ajouter une destination AWS IoT SiteWise Edge en temps réel](#)

- [Ajouter une destination AWS IoT SiteWise mise en mémoire tampon à l'aide d'Amazon S3](#)
- [Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge](#)
- [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#)
- [Gérer les destinations AWS IoT SiteWise Edge](#)

Connectez des applications externes au broker EMQX

Ce guide explique comment connecter des applications externes à votre passerelle AWS IoT SiteWise Edge via un courtier EMQX sur votre passerelle V3 compatible MQTT déployée. Les applications externes peuvent inclure des outils de surveillance personnalisés, des logiciels de visualisation tiers ou des systèmes existants qui doivent interagir avec vos données industrielles à la périphérie.

Nous aborderons les étapes de configuration pour Linux et les Microsoft Windows environnements, notamment la configuration du déploiement d'EMQX, la configuration TLS pour des connexions sécurisées et les règles d'autorisation pour contrôler l'accès à des sujets spécifiques.

Note

EMQX n'est ni un fournisseur ni un fournisseur d' AWS IoT SiteWise Edge.

Important

Pour sécuriser les connexions à votre passerelle, nous vous recommandons vivement d'utiliser l'authentification par certificat via la fonction d'authentification de l'appareil AWS IoT Greengrass client. Cette méthode fournit une sécurité robuste grâce à l'authentification TLS mutuelle (mTLS). Pour plus d'informations, voir [Connecter les appareils clients aux appareils principaux](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Si vous n'êtes pas en mesure d'utiliser l'authentification basée sur des certificats, suivez ce guide pour configurer l'authentification à l'aide de noms d'utilisateur et de mots de passe.

Prérequis

- Une passerelle V3 compatible SiteWise Edge MQTT qui a été déployée et est en ligne

- Accès à l'hôte de la passerelle
- Accès aux AWS IoT Greengrass consoles AWS IoT SiteWise et

Rubriques

- [Format de charge utile des messages pour le courtier EMQX sur Edge AWS IoT SiteWise](#)
- [Configuration du broker EMQX](#)
- [Connecter une application au broker EMQX sur Edge AWS IoT SiteWise](#)
- [Configurer des règles d'autorisation pour AWS IoT SiteWise Edge dans EMQX](#)

Format de charge utile des messages pour le courtier EMQX sur Edge AWS IoT SiteWise

Pour que le composant SiteWise éditeur IoT puisse consommer les données de votre application externe et les publier AWS IoT SiteWise dans le cloud, la charge utile envoyée au courtier doit répondre à des exigences spécifiques.

Comprendre le format de charge utile est essentiel pour une communication MQTT réussie avec AWS IoT SiteWise Edge. Bien que le processus de configuration de la connexion soit abordé dans les sections suivantes, nous présentons d'abord les exigences en matière de charge utile pour vous aider à planifier votre mise en œuvre.

Exigences relatives aux rubriques MQTT

Il n'y a aucune restriction quant à la structure des rubriques MQTT, y compris le nombre de niveaux ou de caractères utilisés. Cependant, nous recommandons que le sujet corresponde au `propertyAlias` champ de la charge utile.

Exemple Exemple d'alias de propriété

Si le sujet du MQTT l'est `site1/line1/compressor1/temperature`, assurez-vous que les `propertyAlias` correspondances sont correctes.

```
{
  "assetId": "compressor_asset_01",
  "propertyAlias": "site1/line1/compressor1/temperature",
  "propertyId": "temperature_sensor_01",
  "propertyValues": [
    {
      "quality": "GOOD",
      "timestamp": {
```

```
        "offsetInNanos": 0,
        "timeInSeconds": 1683000000
    },
    "value": {
        "doubleValue": 23.5
    }
}
]
```

Structure de charge utile JSON

La charge utile des messages MQTT est écrite en JSON et suit le format de PutAssetPropertyValueEntry message défini dans la référence de l'[AWS IoT SiteWise API](#).

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
      "timestamp": {
        "offsetInNanos": number,
        "timeInSeconds": number
      },
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      }
    }
  ]
}
```

Note

Pour qu'un message soit considéré comme valide, seule l'une des conditions suivantes peut être vraie :

- Le propertyAlias est défini, ou

- Les deux `assetId` et `propertyId` sont définis

`PutAssetPropertyValueEntry` contient un `entryId` champ qui n'est pas obligatoire dans ce contexte.

Configuration du broker EMQX

Cette section explique comment ajouter des noms d'utilisateur et des mots de passe. Il explique également comment établir une connexion TLS à partir d'une source externe à l'aide du nom d'utilisateur et du mot de passe ajoutés. Vous pouvez configurer le broker EMQX à l'aide de Linux ou. Microsoft Windows

Note

Pour configurer le broker, vous avez besoin d'un périphérique principal configuré avec la configuration EMQX par défaut dans votre passerelle V3 compatible MQTT.

Important

Une fois cette procédure terminée, nous vous recommandons vivement de configurer les règles d'autorisation. Pour de plus amples informations, veuillez consulter [Configurer des règles d'autorisation pour AWS IoT SiteWise Edge dans EMQX](#). Les règles d'autorisation pour les utilisateurs ajoutés renforcent la sécurité.

Mettre à jour la configuration de déploiement d'EMQX pour l'authentification

Pour mettre à jour la configuration de déploiement d'EMQX pour l'authentification

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Choisissez la passerelle à configurer.
4. Dans la section Configuration de la passerelle Edge, copiez la valeur principale de votre appareil Greengrass. Conservez-le pour une utilisation ultérieure.
5. Ouvrez la [AWS IoT console](#).

6. Dans le menu de navigation de gauche, dans la section Gérer, choisissez Greengrass devices, puis Deployments.
7. Recherchez la valeur principale de l'appareil que vous avez enregistrée précédemment et cliquez sur ce lien pour ouvrir le déploiement.
8. Cliquez sur le bouton déroulant Actions, puis sur Réviser.
9. Lisez le message qui s'affiche, puis choisissez Revise le déploiement. La page Spécifier la cible apparaît.
10. Choisissez Next jusqu'à ce que vous atteigniez l'étape Configurer les composants.
11. Sélectionnez le bouton `aws.greengrass.clientdevices.mqtt.EMQX radio`.
12. Cliquez sur le bouton Configurer le composant. Une page de configuration s'affiche pour le composant.
13. Sous Mise à jour de la configuration, choisissez Rétablir la configuration par défaut pour la version du composant : 2.*.*.
14. Entrez la configuration suivante dans la section Configuration à fusionner en fonction de votre système d'exploitation.

Linux

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",
            "fail_if_no_peer_cert": false
          }
        }
      }
    }
  }
}
```



```

    }
  },
  "authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
      "iterations": 210000,
      "mac_fun": "sha512",
      "name": "pbkdf2"
    },
    "user_id_type": "username"
  },
  "dashboard": {
    "listeners": {
      "http": {
        "bind": 18083
      }
    }
  },
  "authMode": "bypass",
  "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883\n-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e\nEMQX_NODE__NAME=emqx@local",
  "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {

```

```
        "default": {
            "enabled": true,
            "enable_authn": true,
            "ssl_options": {
                "verify": "verify_none",
                "fail_if_no_peer_cert": false
            }
        }
    },
    "authentication": {
        "enable": true,
        "backend": "built_in_database",
        "mechanism": "password_based",
        "password_hash_algorithm": {
            "iterations": 210000,
            "mac_fun": "sha512",
            "name": "pbkdf2"
        },
        "user_id_type": "username"
    },
    "dashboard": {
        "listeners": {
            "http": {
                "bind": 18083
            }
        }
    },
    "authMode": "bypass",
    "requiresPrivilege": "true"
}
```

Le `dockerOptions` champ concerne uniquement les passerelles Linux.

15. Choisissez Confirmer.
16. Choisissez Next jusqu'à ce que vous atteigniez l'étape de révision.
17. Choisissez Déployer.
18. Une fois le déploiement réussi, passez à l'étape suivante.

Activer l'authentification par nom d'utilisateur et mot de passe

Cette section explique comment ajouter des noms d'utilisateur et des mots de passe via l'interface graphique du tableau de bord EMQX.

Note

Les instructions relatives à EMQX fournies sont fournies à titre de référence uniquement. Étant donné que la documentation et les fonctionnalités d'EMQX peuvent changer au fil du temps et que nous ne conservons pas leur documentation, nous vous recommandons de consulter la [documentation officielle d'EMQX](#) pour obtenir les informations les plus récentes.

EMQX Dashboard

Pour activer l'authentification par nom d'utilisateur et mot de passe via le tableau de bord EMQX

1. Assurez-vous que vous vous trouvez dans l'hôte de la passerelle.
2. Ouvrez une fenêtre de navigateur et rendez-vous sur <http://localhost:18083/>.
3. Entrez le nom d'utilisateur **admin** et le mot de passe par défaut **depublic**. Pour plus d'informations, consultez le tableau de [bord EMQX](#) dans la documentation EMQX.
4. Une fois connecté, vous êtes invité à modifier votre mot de passe. Mettez à jour votre mot de passe pour accéder au tableau de bord EMQX.
5. Dans le volet de navigation de gauche, choisissez l'icône en forme de bouclier, puis Authentification.
6. Dans la ligne Base de données intégrée, cliquez sur le bouton Utilisateurs.
7. Cliquez sur le bouton représentant le signe plus pour ajouter des utilisateurs. Un écran d'ajout apparaît.
8. Entrez un nom d'utilisateur et un mot de passe pour l'utilisateur de l'application externe.
9. Choisissez Enregistrer. Le nom d'utilisateur que vous avez choisi apparaît dans le tableau de la page d'authentification.

Note

Les règles d'autorisation existantes ou par défaut s'appliquent au nouvel utilisateur. Il est recommandé de les examiner et de les adapter aux besoins de votre application externe.

EMQX Management with Linux

Utilisez l'outil AWS IoT SiteWise CLI EMQX à l'adresse. `/greengrass/v2/bin/swe-emqx-cli`

Pour activer l'authentification par nom d'utilisateur et mot de passe via EMQX Management sous Linux

1. Modifiez le mot de passe administrateur en exécutant la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli admin change-pwd
```

2. Lorsque vous y êtes invité, procédez comme suit :

1. Entrez votre nom d'utilisateur administrateur (par défaut `admin`) et votre mot de passe (par défaut `public`).
2. Entrez et confirmez votre nouveau mot de passe.

En cas de réussite, le message suivant s'affiche :

```
admin password changed successfully
```

3. Ajoutez des utilisateurs pour les applications externes en exécutant la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli users add
```

4. Lorsque vous y êtes invité, procédez comme suit :

1. Entrez le nom d'utilisateur du nouvel utilisateur.
2. Entrez et confirmez le mot de passe du nouvel utilisateur.

En cas de réussite, le message suivant s'affiche :

```
User '[username]' created successfully
```

5. Vérifiez la configuration utilisateur en exécutant la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli users list
```

Le résultat montre tous les utilisateurs configurés :

```
Users:
- [your-added-username]

Total users: 1
```

EMQX Management with Windows

Utilisez l'outil AWS IoT SiteWise CLI EMQX à l'un des emplacements suivants :

- PowerShell: `C:\greengrass\v2\bin\swe-emqx-cli.ps1`
- Invite de commande : `C:\greengrass\v2\bin\swe-emqx-cli.bat`

Pour activer l'authentification par nom d'utilisateur et mot de passe via EMQX Management sous Windows

1. Modifiez le mot de passe administrateur en exécutant la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 admin change-pwd
```

2. Lorsque vous y êtes invité, procédez comme suit :
 1. Entrez votre nom d'utilisateur administrateur (par défaut `admin`) et votre mot de passe (par défaut `public`).
 2. Entrez et confirmez votre nouveau mot de passe.

En cas de réussite, le message suivant s'affiche :

```
admin password changed successfully
```

3. Ajoutez des utilisateurs pour les applications externes en exécutant la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users add
```

4. Lorsque vous y êtes invité, procédez comme suit :
 1. Entrez le nom d'utilisateur du nouvel utilisateur.

2. Entrez et confirmez le mot de passe du nouvel utilisateur.

En cas de réussite, le message suivant s'affiche :

```
User '[username]' created successfully
```

5. Vérifiez la configuration utilisateur en exécutant la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 users list
```

Le résultat montre tous les utilisateurs configurés :

```
Users:
- [your-added-username]

Total users: 1
```

Connecter une application au broker EMQX sur Edge AWS IoT SiteWise

Le broker EMQX utilise le protocole TLS (Transport Layer Security) sur le port 8883 pour chiffrer toutes les communications, garantissant ainsi la protection de vos données pendant la transmission. Cette section explique les étapes à suivre pour établir des connexions entre vos applications et le broker EMQX. Le respect de ces étapes permet de préserver l'intégrité et la confidentialité de vos données industrielles. Le processus de connexion implique deux approches principales : l'utilisation de la découverte automatique des adresses IP par le biais de composants ou la configuration manuelle des noms DNS et des adresses IP en tant que noms alternatifs de sujet (SANs) dans vos certificats TLS. Chaque méthode présente ses propres avantages en fonction de la configuration de votre réseau et de vos exigences en matière de sécurité. Cette documentation vous guidera à travers les deux options.

Rubriques

- [Configuration du protocole TLS pour des connexions sécurisées au courtier EMQX sur Edge AWS IoT SiteWise](#)
- [Testez la connexion du broker EMQX sur Edge AWS IoT SiteWise](#)
- [Utilisez votre propre CA](#)
- [Ouvrez le port 8883 pour les connexions de pare-feu externes](#)

Configuration du protocole TLS pour des connexions sécurisées au courtier EMQX sur Edge AWS IoT SiteWise

AWS IoT Greengrass Génère par défaut un certificat de serveur TLS pour le courtier EMQX qui est signé par l'autorité de certification (CA) du périphérique principal. Pour plus d'informations, consultez [Connecter des appareils clients à un périphérique AWS IoT Greengrass Core avec un courtier MQTT](#).

Récupérez le certificat TLS

Pour obtenir le certificat CA, exécutez la commande suivante sur l'hôte de la passerelle :

Linux

Exécutez la commande suivante dans une session shell sur l'hôte de passerelle :

```
/greengrass/v2/bin/swe-emqx-cli cert
```

Cette commande affiche l'emplacement du certificat et imprime le contenu du certificat.

Vous pouvez également enregistrer le certificat dans un fichier à l'aide de cette commande :

```
/greengrass/v2/bin/swe-emqx-cli cert --output /path/to/certificate.pem
```

Windows

Exécutez la commande suivante dans une PowerShell session sur l'hôte de la passerelle :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert
```

Cette commande affiche l'emplacement du certificat et imprime le contenu du certificat.

Vous pouvez également enregistrer le certificat dans un fichier à l'aide de cette commande :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 cert --output C:\path\to\certificate.pem
```

La CLI localise automatiquement le certificat quel que soit le chemin exact sur votre système.

Copiez le contenu du fichier ca.pem dans l'application externe que vous connectez au broker. Enregistrez-le sous BrokerCoreDeviceCA.pem.

Ajouter des noms DNS/adresses IP personnalisés au certificat du serveur TLS

Le nom alternatif du sujet (SAN) sur le certificat généré par AWS IoT Greengrass est `localhost`. Lors de l'établissement d'une connexion TLS depuis l'extérieur de l'hôte de la passerelle, l'étape de vérification TLS échoue car le nom d'hôte du courtier ne correspond pas au nom d'hôte figurant `localhost` sur le certificat du serveur.

Pour résoudre le problème de non-concordance des noms d'hôte, AWS IoT Greengrass propose deux méthodes de gestion des points de terminaison des appareils principaux. Cette section couvre les deux options. Pour des informations plus détaillées, consultez la section [Gérer les principaux points de terminaison des appareils](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

- Pour vous connecter au broker EMQX à l'aide de l'adresse IP de l'appareil principal, utilisez la section Découverte automatique des adresses IP.
- Pour vous connecter au broker EMQX à l'aide d'un nom DNS au lieu d'une adresse IP, utilisez la section Gestion manuelle.

Automated IP discovery

Cette option permet à votre appareil principal de découvrir automatiquement son adresse IP et de l'ajouter en tant que nom alternatif du sujet (SAN) au certificat du courtier.

1. Ajoutez le `aws.greengrass.clientdevices.IPDetector` composant au déploiement de votre appareil principal.
2. Déployez les modifications sur votre appareil
3. Attendez que le déploiement soit terminé.

Une fois le déploiement terminé, vous pouvez établir une connexion TLS sécurisée à l'aide de l'adresse IP du courtier.

L'adresse IP est automatiquement ajoutée en tant que SAN au certificat du courtier.

Manual DNS and IP Configuration

Vous pouvez ajouter manuellement des noms DNS et des adresses IP en tant que noms alternatifs de sujet (SANs) à votre certificat TLS. Cette méthode est utile lorsque vous avez configuré un nom DNS pour l'hôte de votre passerelle.

⚠ Important

Si vous utilisez le IPDetector composant, supprimez-le de votre déploiement avant de continuer. Le IPDetector composant remplace les configurations manuelles des points de terminaison.

Pour configurer manuellement les points de terminaison

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Choisissez la passerelle à configurer.
4. Dans la section Configuration de la passerelle Edge, choisissez l'URL de votre appareil principal Greengrass. La page de l'appareil principal s'affiche.
5. Choisissez l'onglet Appareils clients.
6. Choisissez Gérer les points de terminaison.
7. Dans la boîte de dialogue Gérer les points de terminaison, entrez le ou les noms DNS et les adresses IP que vous souhaitez ajouter. SANs Utilisez le port 8883.
8. Choisissez Mettre à jour.

Le certificat de serveur TLS du courtier est automatiquement mis à jour pour inclure vos nouveaux points de terminaison.

Pour vérifier la mise à jour du certificat du serveur TLS sous Linux

1. Démarrez une session shell sur votre hôte de passerelle.

```
docker exec emqx openssl x509 -in ./data/cert.pem -text -noout | grep -A1  
"Subject Alternative Name"
```

2. La commande renvoie un résultat similaire à ce qui suit :

```
X509v3 Subject Alternative Name:  
DNS:endpoint_you_added, DNS:localhost
```

3. Vérifiez que votre point de terminaison apparaît dans la liste des SANs.

Pour vérifier la mise à jour du certificat du serveur TLS à l'aide de Windows

1. Démarrez une session shell sur votre hôte de passerelle.

```
(Get-PfxCertificate -FilePath "C:\greengrass\v2\work
\aws.greengrass.clientdevices.mqtt.EMQX\v2\data\cert.pem").Extensions | Where-
Object { $_.Oid.FriendlyName -eq "Subject Alternative Name" } | ForEach-Object
{ "Subject Alternative Name:", ($_.Format($true) -split "`n")[0..1] }
```

2. La commande renvoie un résultat similaire à ce qui suit :

```
Subject Alternative Name:
DNS Name=your-endpoint
DNS Name=localhost
```

3. Vérifiez que le point de terminaison que vous avez ajouté figure dans la liste des SANs.

Testez la connexion du broker EMQX sur Edge AWS IoT SiteWise

Après avoir configuré votre broker EMQX avec des certificats TLS et des informations d'authentification, il est important de vérifier que votre configuration fonctionne correctement. Le test de la connexion permet de s'assurer que vos configurations de sécurité sont correctement mises en œuvre et que les clients peuvent établir avec succès des connexions chiffrées avec le courtier. Cette section explique comment tester votre connexion au courtier à l'aide du client d'interface de ligne de commande (CLI) Mosquitto, un outil client MQTT largement utilisé qui prend en charge le chiffrement et l'authentification TLS.

Utiliser le client Mosquitto CLI pour tester la connexion du broker EMQX

Au cours de cette étape, nous utiliserons le client CLI Mosquitto pour tester notre configuration et nous assurer que nous pouvons nous connecter avec succès au courtier en utilisant le nom d'utilisateur et le mot de passe que nous avons créés précédemment. Pour `BrokerCoreDeviceCA.pem` suivre les étapes décrites à l'étape 3 : Configuration du protocole TLS.

```
mosquitto_sub -h hostname/ip address \
  -p 8883 \
  -t "#" \
  -q 1 \
  -u username -P password \
  --cafile BrokerCoreDeviceCA.pem
```

 Note

Une erreur SSL:Verify peut s'afficher si l' hostname/IP adresse à laquelle vous vous connectez ne correspond pas au nom alternatif du sujet (SAN) figurant sur le certificat CA que vous transmettez au client. Reportez-vous à la section « Ajout d' names/IP adresses DNS personnalisées au certificat du serveur TLS » à l'étape 3 : Configuration du protocole TLS pour savoir comment obtenir un certificat avec le SAN approprié.

À ce stade, tous les utilisateurs peuvent publier et s'abonner à tous les sujets du courtier. Passez à [Configurer des règles d'autorisation pour AWS IoT SiteWise Edge dans EMQX](#).

Utilisez votre propre CA

AWS IoT Greengrass explique comment configurer le composant d'authentification de votre appareil client pour utiliser votre propre autorité de certification (CA). Le composant d'authentification du dispositif client (`aws.greengrass.clientdevices.Auth`) authentifie les dispositifs clients et autorise les actions du dispositif client. Pour plus d'informations, consultez la section [Utilisation de votre propre autorité de certification](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Pour utiliser votre propre autorité de certification, ajoutez le `aws.greengrass.clientdevices.Auth` composant à votre déploiement afin de pouvoir spécifier une configuration personnalisée.

Ouvrez le port 8883 pour les connexions de pare-feu externes

Linux

Dans la règle de pare-feu de votre hôte Linux, ajoutez une règle entrante pour le port 8883 afin d'autoriser les connexions entrantes provenant de l'extérieur de l'hôte de passerelle. Si des pare-feux sont en place, assurez-vous que les connexions TLS entrantes sur le port 8883 sont autorisées.

Windows

Dans la règle de pare-feu de votre Microsoft Windows hôte, ajoutez une règle entrante pour le port 8883 afin d'autoriser les connexions entrantes provenant de l'extérieur de l'hôte de la passerelle. Assurez-vous que la règle est une règle d'autorisation, de type port, spécifiant le port 8883. Vous pouvez le configurer en fonction de la configuration de votre réseau pour autoriser les connexions entre vos applications externes et le broker.

Configurer des règles d'autorisation pour AWS IoT SiteWise Edge dans EMQX

EMQX prend en charge l'ajout de règles d'autorisation basées sur des identifiants tels que le nom d'utilisateur, l'adresse IP ou l'identifiant du client. Cela est utile si vous souhaitez limiter le nombre d'applications externes se connectant à différentes opérations ou sujets.

Rubriques

- [Configurer l'autorisation à l'aide de la base de données intégrée sous Linux](#)
- [Configuration de l'autorisation à l'aide de la base de données intégrée avec Windows](#)
- [Mettre à jour la configuration de déploiement d'EMQX pour obtenir une autorisation](#)
- [Ajoutez des règles d'autorisation via le tableau de bord EMQX pour les utilisateurs](#)

Configurer l'autorisation à l'aide de la base de données intégrée sous Linux

Lorsque vous configurez des règles d'autorisation, il existe deux choix de configuration qui dépendent de la configuration de votre déploiement.

- Docker— Si vous exécutez une Docker installation standard sans Litmus Edge, utilisez la configuration de la passerelle Docker Bridge. C'est généralement le cas lorsque vous n'avez déployé que AWS IoT SiteWise des composants.
- Litmus Edge— Si vous l'avez Litmus Edge installé sur votre passerelle, utilisez la configuration du Litmus Edgesous-réseau.

Note

Si vous configurez initialement la passerelle de Docker pont et que vous l'installez ultérieurement Litmus Edge, reconfigurez les règles d'autorisation à l'aide de l'option de Litmus Edge sous-réseau pour garantir une communication correcte entre tous les composants.

Pour ajouter des règles d'autorisation de base

1. Vérifiez que le broker EMQX est déployé et en cours d'exécution.
2. Démarrez une session shell sur votre hôte de passerelle.

Docker without Litmus Edge

Pour une Docker installation standard sans Litmus Edge, exécutez :

```
/greengrass/v2/bin/swe-emqx-cli acl init
```

Litmus Edge network subnet

Si vous utilisez Litmus Edge, déterminez l'adresse IP du sous-réseau Litmus Edge :

```
docker network inspect LitmusNetwork | grep IPAM -A9
```

Notez la valeur du sous-réseau à partir de la sortie et exécutez la commande suivante. Remplacez `litmus_subnet_ip` par la valeur du sous-réseau de l'étape précédente.

```
/greengrass/v2/bin/swe-emqx-cli acl init litmus_subnet_ip
```

L'outil crée et applique automatiquement des règles d'autorisation pour autoriser les connexions entre l'adresse IP fournie et le courtier. Il permet d'accéder à tous les sujets. Cela inclut le collecteur IoT SiteWise OPC UA et l' SiteWise éditeur IoT.

3. Passez à [Mettre à jour la configuration de déploiement d'EMQX pour obtenir une autorisation](#).

Configuration de l'autorisation à l'aide de la base de données intégrée avec Windows

Cette section décrit la configuration des règles d'autorisation à l'aide de la base de données intégrée pour les déploiements Windows.

Pour ajouter des règles d'autorisation de base

1. Vérifiez que le broker EMQX est déployé et en cours d'exécution.
2. Exécutez l'outil AWS IoT SiteWise CLI EMQX :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 acl init
```

L'outil crée et applique automatiquement des règles ACL autorisant les connexions entre localhost (127.0.0.1) et le broker. Il permet d'accéder à tous les sujets. Cela inclut le collecteur IoT SiteWise OPC UA et l' SiteWise éditeur IoT.

3. Passez à [Mettre à jour la configuration de déploiement d'EMQX pour obtenir une autorisation](#).

Mettre à jour la configuration de déploiement d'EMQX pour obtenir une autorisation

Pour mettre à jour la configuration de déploiement d'EMQX à des fins d'autorisation

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Choisissez la passerelle à configurer.
4. Dans la section Configuration de la passerelle Edge, copiez la valeur principale de votre appareil Greengrass. Conservez-le pour une utilisation ultérieure.
5. Ouvrez la [AWS IoT console](#).
6. Dans le menu de navigation de gauche, dans la section Gérer, choisissez Greengrass devices, puis Deployments.
7. Recherchez la valeur principale de l'appareil que vous avez enregistrée précédemment et cliquez sur ce lien pour ouvrir le déploiement.
8. Cliquez sur le bouton déroulant Actions, puis sur Réviser.
9. Lisez le message qui s'affiche, puis choisissez Revise le déploiement. La page Spécifier la cible apparaît.
10. Choisissez Next jusqu'à ce que vous atteigniez l'étape Configurer les composants.
11. Sélectionnez le bouton `aws.greengrass.clientdevices.mqtt.EMQX radio`.
12. Cliquez sur le bouton Configurer le composant. Une page de configuration s'affiche pour le composant.
13. Sous Mise à jour de la configuration, choisissez Rétablir la configuration par défaut pour la version du composant : 2.*.*.
14. Collez le contenu suivant dans la section Configuration pour fusionner en fonction de votre système d'exploitation.

Linux

```
{
```

```
"emqxConfig": {
  "authorization": {
    "no_match": "deny",
    "sources": [
      {
        "type": "built_in_database"
      },
      {
        "type": "file",
        "path": "data/authz/acl.conf"
      }
    ]
  },
  "listeners": {
    "tcp": {
      "default": {
        "enabled": true,
        "enable_authn": false
      }
    },
    "ssl": {
      "default": {
        "enabled": true,
        "enable_authn": true,
        "ssl_options": {
          "verify": "verify_none",
          "fail_if_no_peer_cert": false
        }
      }
    }
  },
  "authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
      "iterations": 210000,
      "mac_fun": "sha512",
      "name": "pbkdf2"
    },
    "user_id_type": "username"
  },
  "dashboard": {
    "listeners": {
```

```

        "http": {
            "bind": 18083
        }
    },
    "authMode": "bypass",
    "dockerOptions": "-p 8883:8883 -p 127.0.0.1:1883:1883\n-p 127.0.0.1:18083:18083 -v emqx-data:/opt/emqx/data -e\nEMQX_NODE__NAME=emqx@local",
    "requiresPrivilege": "true"
}

```

Windows

```

{
  "emqxConfig": {
    "authorization": {
      "no_match": "deny",
      "sources": [
        {
          "type": "built_in_database"
        },
        {
          "type": "file",
          "path": "C:\\greengrass\\v2\\work\\aws.greengrass.clientdevices.mqtt.EMQX\\v2\\data\\authz\\acl.conf"
        }
      ]
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      },
      "ssl": {
        "default": {
          "enabled": true,
          "enable_authn": true,
          "ssl_options": {
            "verify": "verify_none",

```



```
        "fail_if_no_peer_cert": false
      }
    }
  },
  "authentication": {
    "enable": true,
    "backend": "built_in_database",
    "mechanism": "password_based",
    "password_hash_algorithm": {
      "iterations": 210000,
      "mac_fun": "sha512",
      "name": "pbkdf2"
    },
    "user_id_type": "username"
  },
  "dashboard": {
    "listeners": {
      "http": {
        "bind": 18083
      }
    }
  },
  "authMode": "bypass",
  "requiresPrivilege": "true"
}
```

15. Choisissez Confirmer.
16. Choisissez Next jusqu'à ce que vous atteigniez l'étape de révision.
17. Choisissez Déployer.

 Note

À partir de ce moment, vous ne pouvez pas modifier le fichier ACL pour mettre à jour les règles d'autorisation. Vous pouvez également procéder [Ajoutez des règles d'autorisation via le tableau de bord EMQX pour les utilisateurs](#) après un déploiement réussi.

Ajoutez des règles d'autorisation via le tableau de bord EMQX pour les utilisateurs

Vous pouvez ajouter ou mettre à jour des règles d'autorisation à l'aide du tableau de bord EMQX ou de l'outil AWS IoT SiteWise EMQX CLI. L'outil AWS IoT SiteWise EMQX CLI gère les autorisations à l'aide de la base de données intégrée d'EMQX.

Note

L'ajout de règles d'autorisation est une étape de configuration avancée qui nécessite de comprendre les modèles de sujets MQTT et le contrôle d'accès. Pour plus d'informations sur la création de règles d'autorisation à l'aide de la base de données intégrée d'EMQX, consultez la section [Utiliser une base de données intégrée](#) dans la documentation EMQX.

Note

Les instructions relatives à EMQX fournies sont fournies à titre de référence uniquement. Étant donné que la documentation et les fonctionnalités d'EMQX peuvent changer au fil du temps et que nous ne conservons pas leur documentation, nous vous recommandons de consulter la [documentation officielle d'EMQX](#) pour obtenir les informations les plus récentes.

EMQX dashboard

Cette procédure montre comment ajouter des règles d'autorisation sur le tableau de bord EMQX.

Le tableau de bord EMQX n'est accessible que depuis l'hôte de la passerelle. Si vous essayez de vous connecter depuis l'extérieur de l'hôte de la passerelle, vous ne pouvez pas accéder au tableau de bord.

Pour ajouter des règles d'autorisation à l'aide du tableau de bord EMQX

1. Assurez-vous que vous vous trouvez dans l'hôte de la passerelle.
2. Ouvrez une fenêtre de navigateur et rendez-vous sur <http://localhost:18083/>.
3. Connectez-vous au tableau de bord EMQX. Cette procédure suppose que vous avez remplacé vos informations de connexion par défaut par celles de votre choix. Pour plus d'informations sur la configuration initiale, consultez [Activer l'authentification par nom d'utilisateur et mot de passe](#).

4. Choisissez l'icône en forme de bouclier, puis Autorisation dans le menu déroulant.
5. Cliquez sur le bouton Autorisations sur la ligne de base de données intégrée.
6. Dans la section Autorisation de base de données intégrée, ajoutez ou mettez à jour les règles d'autorisation utilisateur en fonction des besoins de votre entreprise. Pour plus d'informations sur la création de règles, consultez la section [Utiliser une base de données intégrée](#) dans la documentation EMQX.

AWS IoT SiteWise CLI tool using Linux

Pour gérer les règles d'autorisation à l'aide de l'outil AWS IoT SiteWise CLI EMQX sous Linux :

- Ajoutez des règles d'autorisation pour un utilisateur en utilisant le format suivant :

```
/greengrass/v2/bin/swe-emqx-cli auth add your-username your-action your-permission your-topic [your-action-permission-topic]
```

Exemple Ajouter des règles d'autorisation pour un utilisateur

Cet exemple montre comment ajouter des règles pour un utilisateur nommé `system1` :

```
/greengrass/v2/bin/swe-emqx-cli auth add system1 \  
  publish allow "sensors/#" \  
  subscribe allow "control/#" \  
  all deny "#"
```

Exemple : Afficher les règles d'autorisation d'un utilisateur

Pour consulter les règles d'autorisation des `system1` utilisateurs, exécutez la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli auth list system1
```

Exemple : Afficher toutes les règles d'autorisation existantes

Pour afficher toutes les règles d'autorisation dont vous disposez actuellement, exécutez la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli auth list
```

Exemple : Supprimer toutes les règles d'autorisation pour un utilisateur

Pour supprimer toutes les règles d'autorisation appliquées à un utilisateur en particulier, exécutez la commande suivante :

```
/greengrass/v2/bin/swe-emqx-cli auth delete system1
```

Vous êtes invité à confirmer la suppression.

AWS IoT SiteWise CLI tool using Windows

Pour gérer les règles d'autorisation à l'aide de l'outil AWS IoT SiteWise CLI EMQX dans :
Windows PowerShell

- Ajoutez des règles d'autorisation pour un utilisateur en utilisant le format suivant :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add your-username your-action your-  
permission your-topic [your-action-permission-topic]
```

Exemple : Ajouter des règles d'autorisation pour un utilisateur

Cet exemple montre comment ajouter des règles pour un utilisateur nommé system1 :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth add system1 `
  publish allow "sensors/#" `
  subscribe allow "control/#" `
  all deny "#"
```

Exemple : Afficher les règles d'autorisation d'un utilisateur

Pour consulter les règles d'autorisation des system1 utilisateurs, exécutez la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list system1
```

Exemple : Afficher toutes les règles d'autorisation existantes

Pour afficher toutes les règles d'autorisation dont vous disposez actuellement, exécutez la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth list
```

Exemple : Supprimer toutes les règles d'autorisation pour un utilisateur

Pour supprimer toutes les règles d'autorisation appliquées à un utilisateur en particulier, exécutez la commande suivante :

```
C:\greengrass\v2\bin\swe-emqx-cli.ps1 auth delete system1
```

Vous êtes invité à confirmer la suppression.

Traitez et visualisez les données avec SiteWise Edge et les outils open source

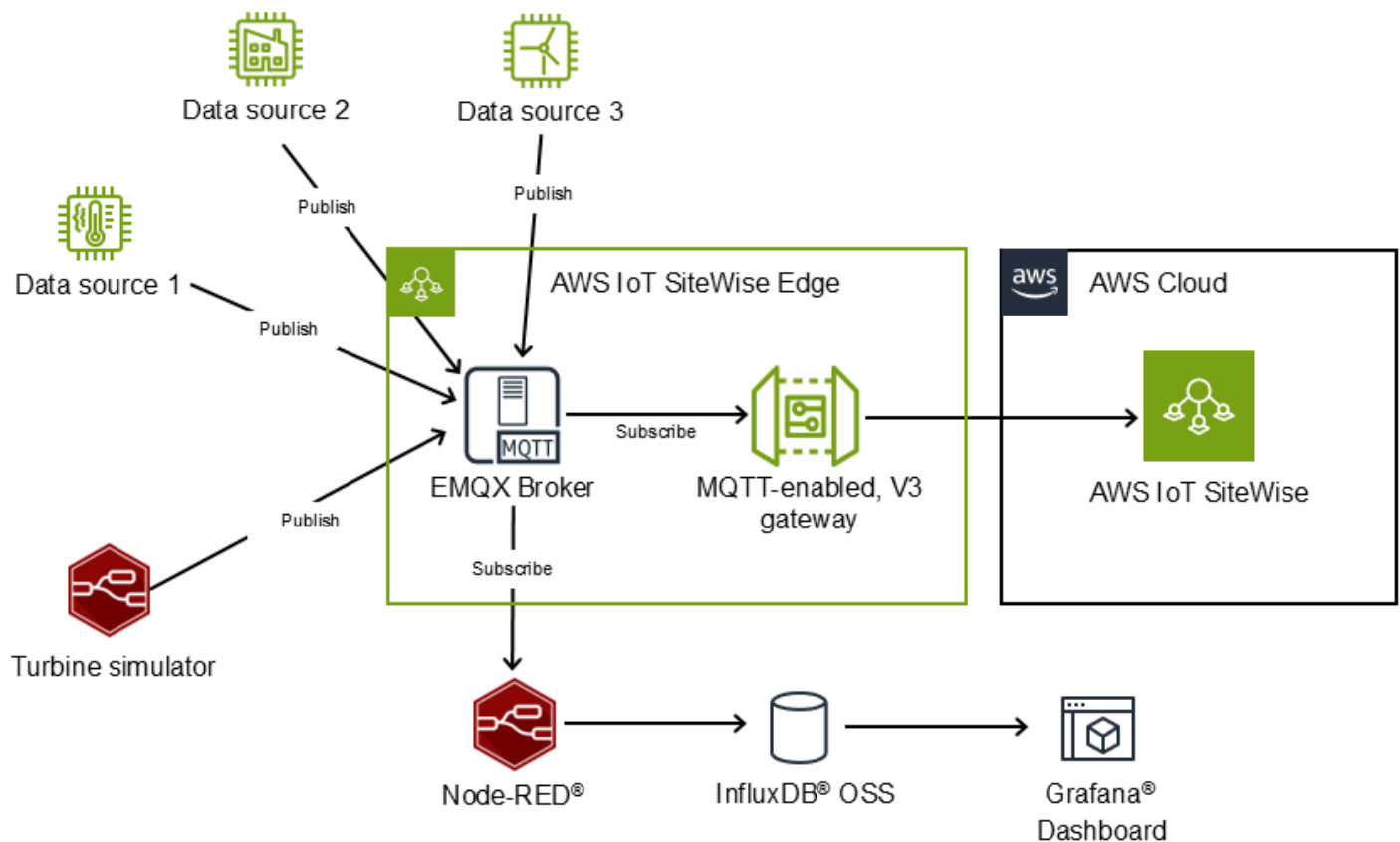
Configurez les passerelles AWS IoT SiteWise Edge compatibles MQTT avec des outils open source pour le traitement et la visualisation locaux afin d'améliorer vos capacités de gestion des données industrielles.

Avec SiteWise Edge, vous pouvez créer un pipeline de traitement de données local à l'aide d'outils open source externes. [Utilisez Node-RED® pour stocker des données de séries chronologiques avec InfluxDB® et surveiller les opérations via les tableaux de bord Grafana®.](#)

Node-RED traite et transforme vos flux de données, tandis qu'InfluxDB fournit un stockage de données chronologiques. Grafana affiche vos données opérationnelles en temps réel. Utilisez ces outils avec SiteWise Edge pour synchroniser les données entre votre environnement local et le AWS Cloud, en vous fournissant à la fois des informations locales immédiates et des capacités d'analyse basées sur le cloud à long terme.

Note

Node-RED®, InfluxDB® et Grafana® ne sont ni des fournisseurs ni des fournisseurs d'Edge. SiteWise



Note

Dans ce guide, nous utilisons la version open source de [Grafana](#) SiteWise for Edge par opposition au service [Amazon Managed Grafana](#).

Options de déploiement

Vous pouvez déployer cette solution en utilisant l'une des deux approches suivantes. Avec une configuration Microsoft Windows manuelle, vous contrôlez la configuration des composants et leur intégration à votre infrastructure. Avec Linux, vous pouvez utiliser Docker pour déployer des composants préconfigurés dans des conteneurs.

Choisissez la méthode qui répond à vos exigences opérationnelles.

- [Configuration manuelle des intégrations open source \(Windows\)](#)— Pour les configurations personnalisées ou l'infrastructure existante
- [Configurer des intégrations open source avec Docker \(Linux\)](#)— Pour un déploiement rapide avec des composants préconfigurés

Aperçu d'un exemple de parc éolien

Ce guide utilise un exemple de parc éolien pour montrer comment surveiller la vitesse du vent d'une éolienne installée sur un parc éolien. Ce scénario pratique illustre les besoins courants de surveillance industrielle pour lesquels la visibilité locale et basée sur le cloud est précieuse pour l'efficacité opérationnelle.

Grâce à cette intégration, vous pouvez :

- Collectez des données provenant d'équipements industriels à l'aide d'une passerelle AWS IoT SiteWise Edge
- Traitez les données localement avec Node-RED, InfluxDB et Grafana
- Stockez les données localement à l'aide d'InfluxDB
- Surveillez les données en temps réel à l'aide des tableaux de bord Grafana

Tout au long de ce guide, nous prenons l'exemple d'un parc éolien. Nous utilisons Node-RED pour simuler une turbine qui génère des données sur la vitesse du vent. Node-RED traduit la charge utile des données, publie les données sur le broker SiteWise Edge MQTT, s'abonne pour recevoir des données du courtier et stocke les données localement dans InfluxDB. Cette approche garantit que toutes les données opérationnelles sont disponibles à la fois localement pour un accès immédiat et dans le cloud pour des analyses supplémentaires. En mettant en œuvre ce modèle, vous gagnez en résilience face aux perturbations du réseau tout en conservant la capacité d'effectuer des analyses avancées dans le AWS Cloud. Grafana se connecte à InfluxDB pour une surveillance locale, offrant aux opérateurs une visibilité en temps réel sur les métriques sans dépendance au cloud. Une passerelle SiteWise Edge compatible MQTT se connecte au même courtier MQTT auquel envoyer des données AWS IoT SiteWise, créant ainsi un pont entre vos opérations de périphérie et les services basés sur le cloud.

Vous pouvez utiliser vos propres données et configurations pour créer un flux de travail similaire adapté à vos exigences industrielles spécifiques, que vous surveilliez des équipements de fabrication, des infrastructures utilitaires ou d'autres actifs industriels.

Exigences relatives aux intégrations open source

Avant de mettre en œuvre des intégrations open source avec SiteWise Edge, assurez-vous que votre environnement répond aux exigences requises.

- Configuration matérielle requise : le matériel de votre passerelle doit répondre aux exigences des passerelles SiteWise Edge. Pour plus d'informations, voir [AWS IoT SiteWise Exigences relatives](#)

à la [passerelle auto-hébergée Edge](#) pour les passerelles MQTT compatibles avec MQTT, les passerelles V3 et. [Exigences relatives à l'application AWS IoT SiteWise Edge](#)

 Important

Lorsque vous déployez des composants open source supplémentaires, assurez-vous que votre matériel répond aux exigences d'InfluxDB, Node-RED et Grafana.

- La configuration de votre réseau doit prendre en charge à la fois la communication locale entre les composants et la connectivité cloud pour SiteWise Edge.
- Tous les services doivent être exécutés sur le même hôte.

Considérations sur la sécurité

Nous vous recommandons de chiffrer toutes les communications entre les composants, en particulier lorsque vous accédez à des interfaces depuis des réseaux non locaux. Mettez en œuvre des contrôles d'accès appropriés pour chaque composant et suivez les AWS meilleures pratiques en matière de configuration de la passerelle AWS IoT SiteWise Edge et de sécurité des AWS comptes.

Environnement de développement

Ce guide montre que Node-RED, InfluxDB et Grafana s'exécutent et sont accessibles localement sur un hôte de passerelle. Pour les déploiements de production nécessitant un accès externe, mettez en œuvre des mesures de sécurité, notamment le chiffrement, l'authentification et l'autorisation TLS. Suivez les meilleures pratiques de sécurité de chaque application.

Logiciels tiers

Cette solution utilise des logiciels tiers non gérés par AWS, notamment InfluxDB, Node-RED, Grafana et le plugin. `node-red-contrib-influxdb` Avant le déploiement, assurez-vous que ces composants sont conformes aux exigences de sécurité, aux normes de conformité et aux politiques de gouvernance de votre entreprise.

 Important

Ce guide fait référence à des logiciels tiers qui ne sont ni détenus ni gérés par AWS. Avant la mise en œuvre, assurez-vous que tous les composants répondent à vos exigences de sécurité, de conformité et de gouvernance. Maintenez tous les logiciels à jour avec

les derniers correctifs de sécurité et suivez les meilleures pratiques pour sécuriser votre déploiement en périphérie.

InfluxDB, Node-red, Grafana ne sont ni des fournisseurs ni des fournisseurs d'Edge. SiteWise

Autres considérations

Tenez compte de ces facteurs supplémentaires lors de la mise en œuvre d'intégrations open source avec SiteWise Edge.

- Utilisez les dernières versions de tous les services, outils et composants.
- Filtrez et agrégez les données localement avant leur transmission dans le cloud afin de réduire les coûts d'ingestion de AWS IoT SiteWise données. Configurez les périodes de conservation des données appropriées dans InfluxDB et dimensionnez correctement votre matériel de passerelle. Pour en savoir plus, consultez [Pricing AWS IoT SiteWise](#) (Tarification).
- Mettez en œuvre des procédures de sauvegarde régulières pour toutes les données.
- Surveillez l'utilisation des ressources sur votre passerelle et configurez les limites de ressources appropriées pour chaque composant. Mettez en œuvre des politiques de conservation des données dans InfluxDB pour gérer l'utilisation du disque.

Configuration manuelle des intégrations open source (Windows)

Utilisez ce guide pour créer manuellement un seau de séries chronologiques pour les données de vitesse du vent qui se connecte à Grafana® et à Node-RED®.

Installez et configurez manuellement Node-RED, InfluxDB® et Grafana pour contrôler la configuration de votre déploiement. Microsoft Windows Vous pouvez stocker et gérer les données de séries chronologiques de vos appareils à l'aide d'InfluxDB.

Conditions préalables à la configuration manuelle

Avant de commencer, remplissez les conditions suivantes :

Note

Exécutez tous les services (SiteWise Edge, InfluxDB, Node-red et Grafana) sur le même hôte.

- Installez une passerelle V3 compatible MQTT. Pour de plus amples informations, veuillez consulter [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#).
- Installez et exécutez ces services localement :
 - InfluxDB OSS v2. Pour les étapes d'installation, voir [Installer InfluxDB](#).
 - Nœud rouge. Pour les étapes d'installation, voir [Installer Node-RED localement](#).
 - Grafana. Pour les étapes d'installation, voir [Installer Grafana](#).

Configurer le stockage local avec InfluxDB

Avec InfluxDB®, vous pouvez stocker des données de séries chronologiques à partir de vos appareils localement. L'objectif de la capacité de stockage local est de maintenir la visibilité opérationnelle en cas de perturbation du réseau et de réduire le temps de latence pour les applications critiques. Vous pouvez effectuer des analyses et des visualisations à la périphérie tout en ayant la possibilité de transférer les données de manière sélective vers le cloud.

Dans cette section, vous allez créer un bucket de séries chronologiques pour les données de vitesse du vent des éoliennes et générer un jeton d'API pour la connectivité Grafana® et Node-RED®. Le bucket InfluxDB sert de conteneur de stockage dédié pour vos données de séries chronologiques, comme une base de données dans les systèmes traditionnels. Le jeton API permet un accès programmatique sécurisé à vos données.

Pour configurer InfluxDB

1. Après avoir effectué les étapes préalables et vérifié que tous les outils s'exécutent sur le même hôte, ouvrez votre navigateur Web et accédez à <http://127.0.0.1:8086>.
2. (Facultatif) Activez le chiffrement TLS pour améliorer la sécurité. Pour plus d'informations, voir [Activer le chiffrement TLS](#) dans le InfluxData Documentation.
3. Créez un bucket InfluxDB de série chronologique pour stocker les données de Node-RED. Le bucket servira de conteneur dédié aux données de votre parc éolien, ce qui vous permettra d'organiser et de gérer les politiques de rétention spécifiques à cet ensemble de données. Pour plus d'informations, consultez la section [Gérer les compartiments](#) dans le InfluxData Documentation.
4. (Facultatif) Configurez la période de conservation des données pour votre emplacement périphérique. La définition de périodes de conservation appropriées permet de gérer efficacement les ressources de stockage en supprimant automatiquement les anciennes données qui ne sont plus nécessaires aux opérations locales.

Pour plus d'informations sur la conservation des données, voir [Conservation des données dans InfluxDB](#) dans le InfluxData Documentation

5. Générez un jeton d'API pour le bucket. Ce jeton permettra une communication sécurisée entre InfluxDB et d'autres composants tels que Node-RED et Grafana. Ainsi, seuls les services autorisés peuvent lire ou écrire dans votre magasin de données. Pour plus d'informations, consultez la section [Créer un jeton](#) dans le InfluxData Documentation.

Une fois ces étapes terminées, vous pouvez stocker les données de séries chronologiques dans votre instance InfluxDB, fournissant ainsi une base pour la persistance et l'analyse des données locales dans votre environnement périphérique.

Configurer les flux Node-RED pour AWS IoT SiteWise l'intégration des données

Avec Node-RED®, vous pouvez implémenter deux flux pour gérer les données entre vos appareils et AWS IoT SiteWise. Ces flux fonctionnent ensemble pour créer une solution complète de gestion des données qui gère à la fois les flux de données locaux et cloud.

- Flux de publication des données : publie dans le cloud. Le flux de publication des données envoie des données à AWS IoT SiteWise. Ce flux simule un dispositif de turbine en générant des données de capteur, en les traduisant au AWS IoT SiteWise format et en les publiant sur le broker SiteWise Edge MQTT. Cela vous permet de tirer parti des capacités AWS IoT SiteWise du cloud pour le stockage, l'analyse et l'intégration avec d'autres AWS services.

Pour de plus amples informations, veuillez consulter [Configuration du flux de publication des données](#).

- Flux de rétention des données — Stocke les données à la périphérie. Le flux de rétention des données s'abonne au broker SiteWise Edge MQTT pour recevoir les données, les traduire au format InfluxDB® et les stocker localement à des fins de surveillance. Ce stockage local fournit un accès immédiat aux données opérationnelles, réduit le temps de latence pour les applications critiques et garantit la continuité en cas de perturbation du réseau.

Pour de plus amples informations, veuillez consulter [Configuration du flux de conservation des données](#).

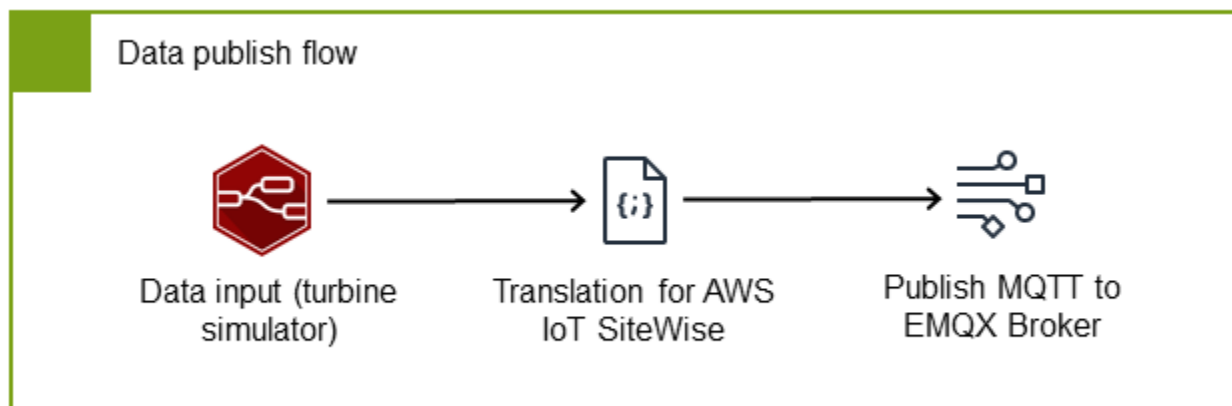
Ces deux flux fonctionnent ensemble pour garantir que les données sont à la fois envoyées AWS IoT SiteWise et stockées localement pour un accès immédiat.

Pour accéder à votre console Node-RED, rendez-vous sur <http://127.0.0.1:1880>. Pour plus d'informations sur l'activation du protocole TLS, consultez la section [Activer le chiffrement TLS](#).

Configuration du flux de publication des données

Le flux de publication des données utilise trois nœuds pour créer un pipeline qui envoie vos données industrielles vers le cloud. Ce flux est essentiel pour permettre l'analyse basée sur le cloud, le stockage à long terme et l'intégration avec d'autres AWS services. Tout d'abord, les données de l'appareil simulé sont envoyées au broker SiteWise Edge MQTT. La passerelle récupère les données auprès du courtier, ce qui permet de les transmettre au AWS IoT SiteWise cloud, où vous pouvez tirer parti de puissantes fonctionnalités d'analyse et de visualisation.

- Entrée de données - Reçoit les données de l'appareil à partir de votre équipement industriel ou de vos simulateurs
- Traducteur de données pour AWS IoT SiteWise : traduit les données au AWS IoT SiteWise format pour garantir la compatibilité avec la passerelle SiteWise Edge
- Éditeur MQTT : publie les données sur le courtier SiteWise Edge MQTT, les mettant à la disposition des consommateurs locaux et du cloud



Configuration du nœud de saisie de données

Dans cet exemple, le nœud de saisie de données utilise un dispositif d'éolienne simulé qui génère des données sur la vitesse du vent. Ce nœud sert de point d'entrée pour vos données industrielles, qu'elles proviennent de sources simulées (comme dans notre exemple) ou d'équipements industriels réels dans des environnements de production.

Nous utilisons un format JSON personnalisé pour la charge utile des données afin de fournir une structure standardisée qui fonctionne efficacement avec les outils de traitement locaux et le service AWS IoT SiteWise cloud. Ce format inclut des métadonnées essentielles telles que les horodatages et les indicateurs de qualité, ainsi que les valeurs de mesure réelles, permettant une gestion complète des données et un suivi de la qualité tout au long de votre pipeline. Importez le nœud d'injection pour recevoir des données simulées dans ce format JSON standardisé avec des horodatages, des indicateurs de qualité et des valeurs.

Pour plus d'informations sur le nœud d'injection Node-RED, consultez la section [Inject](#) de la Node-RED documentation.

Le simulateur de turbine génère des données sur la vitesse du vent chaque seconde dans ce format JSON standardisé :

Exemple : Charge utile de données sur les turbines

```
{
  name: string,          // Property name/identifier
  timestamp: number,     // Epoch time in nanoseconds
  quality: "GOOD" | "UNCERTAIN" | "BAD",
  value: number | string | boolean
}
```

Ce format présente plusieurs avantages :

- Le `name` champ identifie la propriété ou la mesure spécifique, ce qui vous permet de suivre plusieurs points de données à partir du même appareil
- La valeur `timestamp` en nanosecondes garantit un suivi précis du temps pour une analyse historique précise
- L'`quality` indicateur vous aide à filtrer et à gérer les données en fonction de leur fiabilité
- Le `value` champ flexible prend en charge différents types de données pour s'adapter aux différentes sorties de capteurs

Exemple : Nœud d'injection d'un simulateur de turbine

```
[
  {
    "id": "string",
    "type": "inject",
```

```

    "z": "string",
    "name": "Turbine Simulator",
    "props": [
      {
        "p": "payload.timestamp",
        "v": "",
        "vt": "date"
      },
      {
        "p": "payload.quality",
        "v": "GOOD",
        "vt": "str"
      },
      {
        "p": "payload.value",
        "v": "$random()",
        "vt": "jsonata"
      },
      {
        "p": "payload.name",
        "v": "/Renton/WindFarm/Turbine/WindSpeed",
        "vt": "str"
      }
    ],
    "repeat": "1",
    "crontab": "",
    "once": false,
    "onceDelay": "",
    "topic": "",
    "x": 270,
    "y": 200,
    "wires": [
      [
        "string"
      ]
    ]
  }
]

```

Configuration d'un nœud pour la traduction des données

La passerelle SiteWise Edge nécessite des données dans un format spécifique pour garantir la compatibilité avec AWS IoT SiteWise le cloud. Le nœud de traduction est un composant important

qui convertit vos données d'entrée au format de AWS IoT SiteWise charge utile requis. Cette étape de traduction garantit que vos données industrielles peuvent être correctement traitées, stockées et analysées ultérieurement dans l'environnement AWS IoT SiteWise cloud.

En normalisant le format des données à ce stade, vous permettez l'intégration entre vos appareils périphériques et le service cloud, où vous pouvez utiliser les fonctionnalités de modélisation, d'analyse et de visualisation des actifs. Utilisez cette structure :

Exemple : Structure de charge utile pour l'analyse des données SiteWise Edge

```
{
  "propertyAlias": "string",
  "propertyValues": [
    {
      "value": {
        "booleanValue": boolean,
        "doubleValue": number,
        "integerValue": number,
        "stringValue": "string"
      },
      "timestamp": {
        "timeInSeconds": number,
        "offsetInNanos": number
      },
      "quality": "GOOD" | "UNCERTAIN" | "BAD",
    }
  ]
}
```

Note

Faites correspondre le `propertyAlias` à votre hiérarchie de sujets MQTT (par exemple, `/Renton/WindFarm/Turbine/WindSpeed`). Cela garantit que vos données sont correctement associées à la propriété d'actif appropriée dans AWS IoT SiteWise. Pour plus d'informations, consultez le concept « Alias de flux de données » dans [AWS IoT SiteWise concepts](#).

1. Importez le nœud de fonction d'exemple pour la traduction de AWS IoT SiteWise la charge utile. Cette fonction gère la conversion de votre format d'entrée standardisé vers un format AWS IoT

- Préserve l'indicateur de qualité pour le suivi de la fiabilité des données
 - Détecte automatiquement le type de valeur et le formate en fonction des AWS IoT SiteWise besoins
3. Connectez le nœud à votre flux, en le liant entre le nœud d'entrée de données et l'éditeur MQTT.

Pour obtenir des conseils sur la rédaction d'une fonction spécifique aux besoins de votre entreprise, consultez la section [Fonctions d'écriture](#) dans la Node-RED documentation

Configuration de l'éditeur MQTT

Après la traduction, les données sont prêtes à être publiées sur le broker SiteWise Edge MQTT.

Configurez l'éditeur MQTT avec les paramètres suivants pour envoyer des données au courtier SiteWise Edge MQTT :

Pour importer le nœud MQTT sortant

1. Importez un nœud de configuration MQTT en utilisant "type": "mqtt out". Les nœuds de sortie MQTT vous permettent de partager la configuration d'un broker.
2. Entrez des paires clé-valeur pour les informations relatives à la connexion au courtier MQTT et au routage des messages.

Importez le mqtt out nœud d'exemple.

Exemple

```
[
  {
    "id": "string",
    "type": "mqtt out",
    "z": "string",
    "name": "Publish to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "retain": "",
    "respTopic": "",
    "contentType": "",
    "userProps": "",
    "correl": "",
    "expiry": ""
```

```
    "broker": "string",
    "x": 830,
    "y": 200,
    "wires": []
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientId": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]
```

L'exemple de nœud de sortie MQTT crée la connexion MQTT avec les informations suivantes :

- Serveur : 127.0.0.1
- Port : 1883
- Protocole : MQTT V5

Le nœud de sortie MQTT configure ensuite le routage des messages avec les informations suivantes :

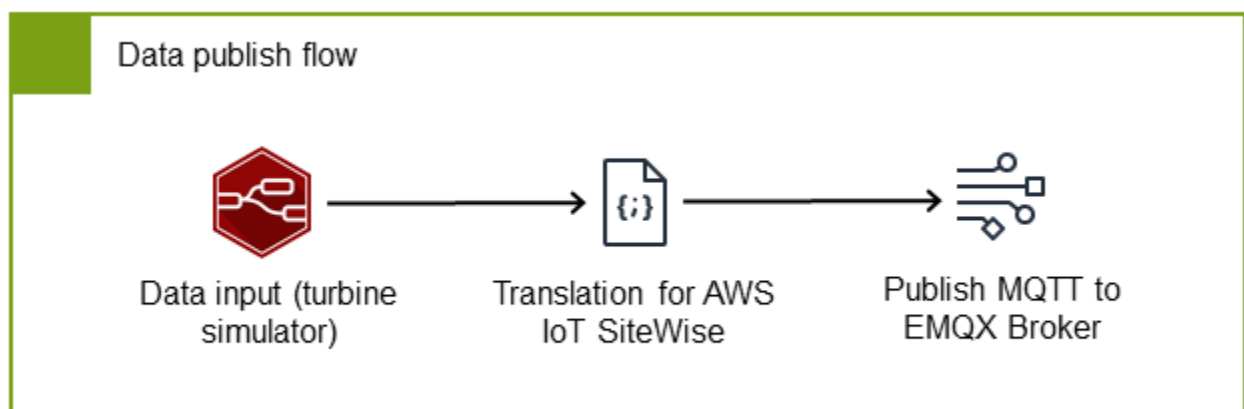
- Sujet: /Renton/WindFarm/Turbine/WindSpeed
- QoS : 1

Déployer et vérifier les nœuds

Après avoir configuré les trois nœuds du flux de publication des données, procédez comme suit pour déployer le flux et vérifier que les données sont correctement transmises à AWS IoT SiteWise

Pour déployer et vérifier les connexions

1. Connectez les trois nœuds comme indiqué dans le flux de publication des données.



2. Choisissez Déployer pour appliquer toutes les modifications de connexion aux nœuds.
3. Accédez à la [AWS IoT SiteWise console](#) et choisissez Data streams.
4. Assurez-vous que le préfixe Alias est sélectionné dans le menu déroulant. Recherchez ensuite l'/Renton/WindFarm/Turbine/WindSpeedalias.

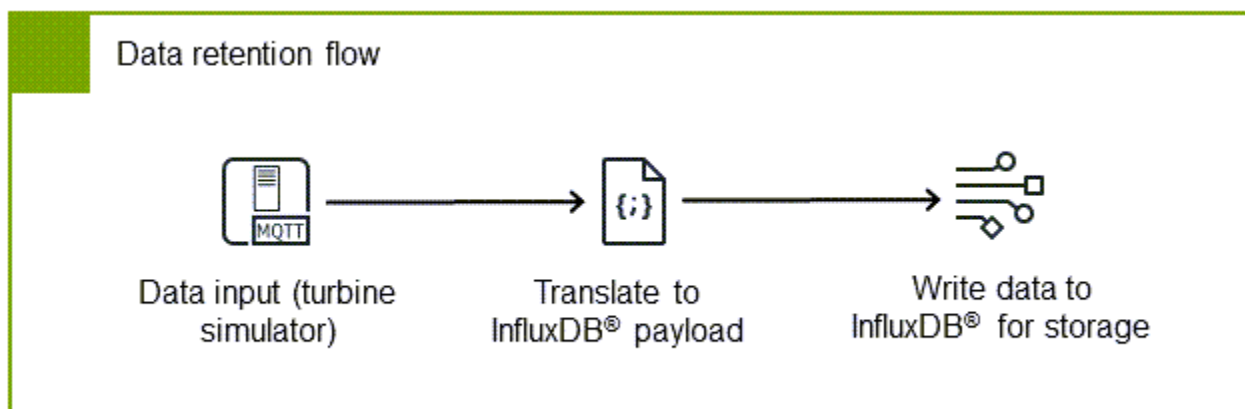
Si vous voyez le bon alias dans votre recherche, cela signifie que vous avez déployé le flux et vérifié la transmission des données.

Configuration du flux de conservation des données

Le flux de rétention des données peut être utilisé pour maintenir la visibilité opérationnelle à la périphérie. Cela est utile lors de perturbations du réseau ou lorsque vous avez besoin d'un accès immédiat à vos données. Ce flux s'abonne au broker MQTT pour recevoir les données de l'appareil, les convertit au format InfluxDB® et les stocke localement. En mettant en œuvre ce flux, vous créez un magasin de données local résilient auquel les opérateurs peuvent accéder sans dépendre du cloud, ce qui permet une surveillance en temps réel et une prise de décision à la périphérie.

Le flux se compose de trois éléments clés qui fonctionnent ensemble pour garantir que vos données sont correctement capturées et stockées :

- Client d'abonnement MQTT : reçoit les données du courtier, ce qui vous permet de saisir toutes les données industrielles pertinentes
- Translateur InfluxDB - Convertit la AWS IoT SiteWise charge utile au format InfluxDB, préparant les données pour un stockage efficace des séries chronologiques
- Rédacteur InfluxDB - Gère le stockage local, garantissant la persistance et la disponibilité des données pour les applications locales



Configuration du client d'abonnement MQTT

- Configurez le client d'abonnement MQTT dans Node-RED pour recevoir les données du courtier MQTT EMQX en AWS IoT SiteWise important l'exemple ci-dessous.

Exemple : MQTT dans le nœud

```
[
  {
    "id": "string",
    "type": "mqtt in",
    "z": "string",
    "name": "Subscribe to MQTT broker",
    "topic": "/Renton/WindFarm/Turbine/WindSpeed",
    "qos": "1",
    "datatype": "auto-detect",
    "broker": "string",
    "nl": false,
```

```
    "rap": true,
    "rh": 0,
    "inputs": 0,
    "x": 290,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  },
  {
    "id": "string",
    "type": "mqtt-broker",
    "name": "emqx",
    "broker": "127.0.0.1",
    "port": "1883",
    "clientId": "",
    "autoConnect": true,
    "usetls": false,
    "protocolVersion": "5",
    "keepalive": 15,
    "cleansession": true,
    "autoUnsubscribe": true,
    "birthTopic": "",
    "birthQos": "0",
    "birthPayload": "",
    "birthMsg": {},
    "closeTopic": "",
    "closePayload": "",
    "closeMsg": {},
    "willTopic": "",
    "willQos": "0",
    "willPayload": "",
    "willMsg": {},
    "userProps": "",
    "sessionExpiry": ""
  }
]
```

Cet abonnement garantit que toutes les données pertinentes publiées auprès du courtier sont capturées pour un stockage local, fournissant ainsi un enregistrement complet de vos opérations

industrielles. Le nœud utilise les mêmes paramètres de connexion MQTT que la [Configuration de l'éditeur MQTT](#) section, avec les paramètres d'abonnement suivants :

- Sujet — /Renton/WindFarm/Turbine/WindSpeed
- QoS — 1

Pour plus d'informations, consultez la section [Connect to an MQTT Broker](#) dans la Node-RED documentation.

Configurer le traducteur InfluxDB

InfluxDB organise les données à l'aide de [balises](#) pour l'indexation et de [champs](#) pour les valeurs. Cette organisation optimise les performances des requêtes et l'efficacité du stockage des données chronologiques. Importez l'exemple de nœud de fonction qui contient le JavaScript code pour convertir la AWS IoT SiteWise charge utile au format InfluxDB. Le traducteur divise les propriétés en deux groupes :

- Tags — Propriétés de qualité et de nom pour une indexation efficace
- Champs — Horodatage (en millisecondes depuis l'époque) et valeur

Exemple : Nœud de fonction permettant de traduire en une charge utile InfluxDB

```
[
  {
    "id": "string",
    "type": "function",
    "z": "string",
    "name": "Translate to InfluxDB payload",
    "func": "let data = msg.payload;\n\nlet timeInSeconds =
data.propertyValues[0].timestamp.timeInSeconds;\nlet offsetInNanos =
data.propertyValues[0].timestamp.offsetInNanos;\nlet timestampInMilliseconds =
(timeInSeconds * 1000) + (offsetInNanos / 1000000);\n\nmsg.payload = [\n  {\n    \"timestamp(milliseconds_since_epoch)\": timestampInMilliseconds,\n    \"value\": data.propertyValues[0].value.doubleValue\n  },\n  {\n    \"name\": data.propertyAlias,\n    \"quality\": data.propertyValues[0].quality\n  }\n]\n\nreturn msg",
    "outputs": 1,
    "timeout": "",
    "noerr": 0,
    "initialize": ""
  }
]
```

```
    "finalize": "",
    "libs": [],
    "x": 560,
    "y": 340,
    "wires": [
      [
        "string"
      ]
    ]
  }
]
```

Pour des options de configuration supplémentaires, consultez [node-red-contrib-influxdb](#) le référentiel Node-RED. GitHub

Configurer le rédacteur InfluxDB

Le nœud d'écriture InfluxDB est le dernier composant de votre flux de rétention de données, chargé de stocker vos données industrielles dans la base de données InfluxDB locale. Ce stockage local est important pour maintenir la visibilité opérationnelle en cas de perturbation du réseau et pour fournir un accès immédiat aux données pour les applications critiques.

1. Installez le `node-red-contrib-influxdb` package via l'option Gérer la palette. Ce paquet fournit les nœuds nécessaires pour connecter Node-RED à InfluxDB.
2. Ajoutez un nœud de sortie InfluxDB à votre flux. Ce nœud gérera l'écriture réelle des données dans votre base de données InfluxDB.
3. Configurez les propriétés du serveur pour établir une connexion sécurisée à votre instance InfluxDB :
 - a. Définir la version sur 2.0 - Cela indique que vous vous connectez à InfluxDB v2.x, qui utilise une API différente de celle des versions précédentes
 - b. Définissez l'URL sur `http://127.0.0.1:8086` - Cela pointe vers votre instance InfluxDB locale
 - c. Entrez votre jeton d'authentification InfluxDB. Ce jeton sécurisé autorise la connexion à votre base de données. Vous avez généré le jeton au cours de la [Configurer le stockage local avec InfluxDB](#) procédure.
4. Spécifiez les paramètres de l'emplacement de stockage pour définir où et comment vos données seront stockées :

- a. Entrez le nom de votre organisation InfluxDB — L'organisation est un espace de travail pour un groupe d'utilisateurs, auquel appartiennent vos buckets et tableaux de bord. Pour plus d'informations, voir [Gérer les organisations](#) dans le InfluxData Documentation.
- b. Spécifiez le bucket InfluxDB (par exemple, WindFarmData) — Le bucket est équivalent à une base de données dans les systèmes traditionnels, servant de conteneur pour vos données de séries chronologiques
- c. Définir la mesure InfluxDB (par exemple, TurbineData) — La mesure est similaire à une table dans les bases de données relationnelles, organisant les points de données associés

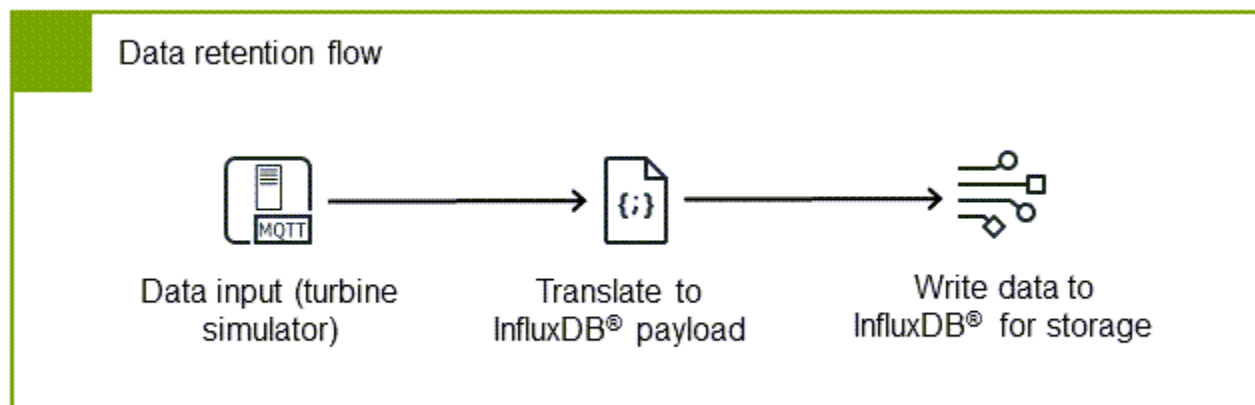
Note

Trouvez le nom de votre organisation dans la barre latérale gauche de l'instance InfluxDB. Les concepts d'organisation, de compartiment et de mesure sont fondamentaux pour le modèle d'organisation des données d'InfluxDB. Pour plus d'informations, consultez la documentation [InfluxDB](#).

Déployer et vérifier le flux de rétention

Après avoir configuré tous les composants du flux de rétention des données, vous devez déployer le système et vérifier qu'il fonctionne correctement. Cette vérification garantit que vos données industrielles sont correctement stockées localement pour un accès et une analyse immédiats.

1. Connectez les trois nœuds comme indiqué dans le diagramme du flux de conservation des données. Cela crée un pipeline complet allant de l'abonnement aux données au stockage local.



2. Choisissez Déployer pour appliquer vos modifications et activer le flux. Cela lance le processus de collecte et de stockage des données.
3. Utilisez l'explorateur de données InfluxDB pour interroger et visualiser vos données. Cet outil vous permet de vérifier que les données sont correctement stockées et de créer des visualisations initiales de vos séries chronologiques.

Dans l'explorateur de données, vous devriez être en mesure de voir les mesures de la vitesse du vent enregistrées au fil du temps, afin de confirmer que l'ensemble du pipeline, de la génération des données au stockage local, fonctionne correctement.

Pour plus d'informations, consultez la section [Requête dans l'explorateur de données](#) dans le InfluxData Documentation.

Avec le flux de publication et le flux de rétention des données déployés, vous disposez désormais d'un système complet qui envoie les données vers le AWS IoT SiteWise cloud tout en conservant une copie locale pour un accès et une résilience immédiats. Cette approche à double voie vous permet de bénéficier des avantages de l'analyse et du stockage basés sur le cloud tout en maintenant une visibilité opérationnelle à la périphérie.

Configurer Grafana pour Edge SiteWise

Grafana® vous permet de créer des tableaux de bord de surveillance locaux en temps réel pour vos données industrielles. En visualisant les données stockées dans InfluxDB®, vous pouvez fournir aux opérateurs des informations immédiates sur les performances des équipements, l'efficacité des processus et les problèmes potentiels. Cette visibilité à la périphérie est importante pour les opérations urgentes et pour assurer la continuité en cas de perturbations du réseau.

Configuration de la source de données

La connexion de Grafana à votre base de données InfluxDB crée une puissante couche de visualisation pour vos données industrielles. Cette connexion permet de disposer de tableaux de bord de surveillance en temps réel que les opérateurs peuvent utiliser pour prendre des décisions éclairées sans dépendre du cloud.

1. Accédez à votre instance Grafana localement en accédant à <http://127.0.0.1:3000> dans votre navigateur. Si l'activation du protocole TLS est requise, vous pouvez vous référer à la section [Configurer Grafana HTTPS pour un trafic Web sécurisé](#) dans Grafana Labs la documentation.

2. Ajoutez une source de données InfluxDB pointant vers le bucket de séries chronologiques InfluxDB dans lequel Node-RED écrit les données. Par exemple, WindFarmData. Cette connexion établit le lien entre vos données stockées et la plateforme de visualisation.
3. Pour des instructions détaillées, voir [Configurer la source de données InfluxDB](#) dans la Grafana Labs documentation.

Création d'un tableau de bord Grafana pour les données Edge SiteWise

La création d'un tableau de bord est la dernière étape de la création de votre solution de surveillance locale. Les tableaux de bord fournissent des représentations visuelles de vos données industrielles, ce qui permet d'identifier plus facilement les tendances, les anomalies et les problèmes potentiels en un coup d'œil.

- Suivez le guide pour créer un tableau de bord. Pour plus d'informations, consultez la section [Création de votre premier tableau de bord](#) dans la Grafana Labs documentation. Ce modèle suppose que votre compartiment porte un nom WindFarmData et que la mesure l'estTurbineData.

Vous pouvez également utiliser le guide de démarrage rapide en important l'exemple de modèle de tableau de bord fourni pour créer rapidement un tableau de bord avec un diagramme chronologique pour les données générées par Node-RED dans la section précédente. Ce modèle fournit un point de départ que vous pouvez personnaliser pour répondre à vos besoins de surveillance spécifiques.

```
{
  "__inputs": [
    {
      "name": "DS_WINDFARM-DEMO",
      "label": "windfarm-demo",
      "description": "",
      "type": "datasource",
      "pluginId": "influxdb",
      "pluginName": "InfluxDB"
    }
  ],
  "__elements": {},
  "__requires": [
    {
      "type": "grafana",
      "id": "grafana",
```

```
    "name": "Grafana",
    "version": "11.6.0-pre"
  },
  {
    "type": "datasource",
    "id": "influxdb",
    "name": "InfluxDB",
    "version": "1.0.0"
  },
  {
    "type": "panel",
    "id": "timeseries",
    "name": "Time series",
    "version": ""
  }
],
"annotations": {
  "list": [
    {
      "builtIn": 1,
      "datasource": {
        "type": "grafana",
        "uid": "-- Grafana --"
      },
      "enable": true,
      "hide": true,
      "iconColor": "rgba(0, 211, 255, 1)",
      "name": "Annotations & Alerts",
      "type": "dashboard"
    }
  ]
},
"editable": true,
"fiscalYearStartMonth": 0,
"graphTooltip": 0,
"id": null,
"links": [],
"panels": [
  {
    "datasource": {
      "type": "influxdb",
      "uid": "${DS_WINDFARM-DEMO}"
    },
    "fieldConfig": {
```

```
"defaults": {
  "color": {
    "mode": "palette-classic"
  },
  "custom": {
    "axisBorderShow": false,
    "axisCenteredZero": false,
    "axisColorMode": "text",
    "axisLabel": "",
    "axisPlacement": "auto",
    "barAlignment": 0,
    "barWidthFactor": 0.6,
    "drawStyle": "line",
    "fillOpacity": 0,
    "gradientMode": "none",
    "hideFrom": {
      "legend": false,
      "tooltip": false,
      "viz": false
    },
    "insertNulls": false,
    "lineInterpolation": "linear",
    "lineWidth": 1,
    "pointSize": 5,
    "scaleDistribution": {
      "type": "linear"
    },
    "showPoints": "auto",
    "spanNulls": false,
    "stacking": {
      "group": "A",
      "mode": "none"
    },
    "thresholdsStyle": {
      "mode": "off"
    }
  },
  "mappings": [],
  "thresholds": {
    "mode": "absolute",
    "steps": [
      {
        "color": "green"
      }
    ]
  },
```

```

        {
            "color": "red",
            "value": 80
        }
    ]
}
},
"overrides": []
},
"gridPos": {
    "h": 8,
    "w": 12,
    "x": 0,
    "y": 0
},
"id": 1,
"options": {
    "legend": {
        "calcs": [],
        "displayMode": "list",
        "placement": "bottom",
        "showLegend": true
    },
    "tooltip": {
        "hideZeros": false,
        "mode": "single",
        "sort": "none"
    }
},
"pluginVersion": "11.6.0-pre",
"targets": [
    {
        "datasource": {
            "type": "influxdb",
            "uid": "${DS_WINDFARM-DEMO}"
        },
        "query": "from(bucket: \"WindFarmData\")\n  |> range(start:
v.timeRangeStart, stop: v.timeRangeStop)\n  |> filter(fn: (r) => r[\"_measurement
\"] == \"TurbineData\")\n  |> filter(fn: (r) => r[\"_field\"] == \"value\")\n
  |> filter(fn: (r) => r[\"name\"] == \"/Renton/WindFarm/Turbine/WindSpeed\")\n
  |> filter(fn: (r) => r[\"quality\"] == \"GOOD\")\n  |> aggregateWindow(every:
v.windowPeriod, fn: mean, createEmpty: false)\n  |> yield(name: \"mean\")",
        "refId": "A"
    }
]

```

```
    ],  
    "title": "Panel Title",  
    "type": "timeseries"  
  }  
],  
"schemaVersion": 41,  
"tags": [],  
"templating": {  
  "list": []  
},  
"time": {  
  "from": "now-6h",  
  "to": "now"  
},  
"timepicker": {},  
"timezone": "browser",  
"title": "demo dashboard",  
"uid": "fejc0t08o6d4wb",  
"version": 1,  
"weekStart": ""  
}
```

Configurer des intégrations open source avec Docker (Linux)

Pour un processus de déploiement rationalisé, vous pouvez utiliser Docker pour configurer Node-RED®, InfluxDB® et Grafana® dans un environnement Linux. Cette méthode utilise des conteneurs préconfigurés, ce qui permet un déploiement rapide et une gestion simplifiée des composants.

Conditions préalables à la configuration de Docker

Avant de commencer, vérifiez que vous disposez des éléments suivants :

- Une passerelle V3 compatible MQTT. Pour de plus amples informations, veuillez consulter [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#).
- Le plugin Docker Compose. Pour les étapes d'installation, voir [Installer le Docker Compose plugin](#) dans la documentation des manuels Docker.

Déployez les services

Ce déploiement exécute SiteWise Edge, InfluxDB, Node-RED et Grafana sur le même hôte.

Configuration de l'environnement

1. Bénéficiez d'un accès root :

```
sudo -i
```

2. Créez un fichier .env ou exportez les variables d'environnement suivantes :

```
export INFLUXDB_PASSWORD=your-secure-influxdb-password  
export INFLUXDB_TOKEN=your-secure-influxdb-token  
export GRAFANA_PASSWORD=your-secure-grafana-password
```

Configuration du réseau Docker

- Créez un réseau de ponts en utilisant le nom SiteWiseEdgeNodeRedDemoNetwork.

```
docker network create --driver=bridge SiteWiseEdgeNodeRedDemoNetwork
```

Mettre à jour le déploiement d' SiteWise Edge

1. Accédez à la [AWS IoT console](#)
2. Choisissez Appareils Greengrass dans le menu de navigation de gauche sous la section Gérer, puis Appareils Core.
3. Sélectionnez le périphérique principal connecté à votre SiteWise Edge Gateway.
4. Choisissez l'onglet Déploiements, puis sélectionnez la valeur de l'ID de déploiement.
5. Choisissez Actions, puis sélectionnez Réviser.
6. Lisez le message contextuel, puis choisissez Revise le déploiement.
7. À l'étape 2 - Sélection des composants, sélectionnez les composants suivants, puis cliquez sur Suivant.
 - `aws.greengrass.clientdevices.mqtt.EMQX`
 - `aws.iot.SiteWiseEdgePublisher`
8. À l'étape 3 - Configuration des composants, sélectionnez la valeur du `aws.greengrass.clientdevices.mqtt.EMQX` composant et ajoutez la configuration réseau suivante :

```
{
  "emqxConfig": {
    "authorization": {
      "no_match": "allow"
    },
    "listeners": {
      "tcp": {
        "default": {
          "enabled": true,
          "enable_authn": false
        }
      }
    },
    "authMode": "bypass",
    "dockerOptions": "-p 127.0.0.1:1883:1883 --network=SiteWiseEdgeNodeRedDemoNetwork",
    "requiresPrivilege": "true"
  }
}
```

9. Choisissez Suivant.
10. À l'étape 4 - Configurer les paramètres avancés, choisissez Next.
11. Choisissez Deploy

Lancez les services

1. Démarrez les services à l'aide du fichier Docker Compose. Exécutez la commande suivante dans le répertoire contenant le `compose.yaml` fichier.

```
docker compose up -d
```

2. Créez un tunnel SSH pour accéder aux services :

```
ssh -i path_to_your_ssh_key -L 1880:127.0.0.1:1880 -L 3000:127.0.0.1:3000 -L 8086:127.0.0.1:8086 username@gateway_ip_address
```

Ce déploiement crée les services suivants dans `SiteWiseEdgeNodeRedDemoNetwork` network :

InfluxDB v2 (port 8086)

Inclut une organisation (iot-sitewise-edge) préconfigurée, un bucket WindFarmData InfluxDB et des informations d'identification d'administrateur

Node-rouge (port 1880)

Inclut des nœuds InfluxDB et des flux préconfigurés pour l'intégration AWS IoT SiteWise

Grafana (port 3000)

Comprend un utilisateur administrateur, une source de données InfluxDB et un tableau de bord de surveillance

Accédez aux services

Après le déploiement, accédez aux services à l'aide des informations suivantes URLs et des informations d'identification :

Note

Vous pouvez accéder à chaque service depuis votre hôte ou depuis la machine passerelle.

Détails d'accès au service

Service	URL	Informations d'identification
Nœud rouge	http://127.0.0.1:1880	Aucune information d'identification requise
InfluxDB	http://127.0.0.1:8086	Nom d'utilisateur : admin Mot de passe : \$INFLUXDB_PASSWORD
Grafana	http://127.0.0.1:3000	Nom d'utilisateur : admin Mot de passe : \$GRAFANA_PASSWORD

Vérification du déploiement

Pour garantir le succès de votre déploiement, effectuez les vérifications suivantes :

1. Pour Node-RED, vérifiez la présence de deux flux préchargés :
 - Flux de publication des données
 - Flux de conservation des données
2. Car AWS IoT SiteWise, dans la AWS IoT SiteWise console, confirmez la présence d'un flux de données avec l'alias/Renton/WindFarm/Turbine/WindSpeed.
3. Pour InfluxDB, utilisez l'explorateur de données pour vérifier le stockage des données dans la TurbineData mesure au sein du WindFarmData bucket.
4. Pour Grafana, consultez le tableau de bord pour confirmer l'affichage des données de séries chronologiques générées par Node-RED.

Traitez les données pour les intégrations open source

Les données peuvent être traitées (telles que la transformation ou l'agrégation) à différentes étapes à l'aide de divers outils, chacun répondant à des exigences de surveillance différentes.

Traitez les données avec les nœuds Node-RED

Transformez vos données en temps réel à l'aide des nœuds de traitement intégrés de Node-RED®. Configurez ces nœuds via la console Node-RED pour créer votre pipeline de données.

Nœuds de transformation de données

Transformez des points de données individuels, de la même manière que Transforms in AWS IoT SiteWise, à l'aide des nœuds suivants :

- change de nœud - Effectue de simples modifications de valeur sur vos données.
- function node - Permet des JavaScript transformations personnalisées pour le traitement de données complexes.

Nœuds de calcul des métriques

Combinez plusieurs points de données en une seule sortie, comme Metrics in AWS IoT SiteWise, à l'aide des nœuds suivants :

- nœud batch : regroupe plusieurs messages pour le traitement par lots.
- join node - Combine plusieurs flux de données en une seule sortie.
- nœud d'agrégation - Calcule les métriques agrégées à partir de plusieurs points de données.

Pour des options de nœud supplémentaires, consultez la bibliothèque [Node-RED](#).

Création de tâches InfluxDB

Bien que Node-RED excelle dans le traitement de base des données grâce à une configuration rapide, les calculs métriques complexes peuvent s'avérer difficiles dans le cadre de la programmation basée sur les flux. Les tâches InfluxDB® offrent une alternative via des scripts Flux planifiés pour les besoins de traitement avancés.

Utilisez les tâches InfluxDB pour :

- Agrégations statistiques sur de grands ensembles de données
- Opérations mathématiques sur plusieurs propriétés
- Mesures dérivées à partir de sources multiples

Caractéristiques des tâches

- Exécution planifiée - Exécute des tâches basées sur des expressions cron
- Traitement par lots - Optimisation des opérations pour les données de séries chronologiques
- Récupération des erreurs - Réessayez automatiquement les opérations ayant échoué
- Surveillance - Suivez l'exécution grâce à des journaux détaillés

Gérez les tâches via l'interface utilisateur, l'API ou la CLI d'InfluxDB. Pour plus d'informations, voir [Traiter les données avec les tâches InfluxDB](#).

Utiliser les transformations Grafana

Transformez la visualisation des données dans Grafana® sans modifier les données sources dans InfluxDB. Les transformations Grafana s'appliquent uniquement à la couche de visualisation.

- Visual Builder - Créez des transformations sans écrire de code
- Aperçu en direct : visualisez les résultats de transformation en temps réel

- Multi-Source : traitez les données provenant de plusieurs sources de base de données
- Stockage efficace : transformez les données au moment de la visualisation sans stocker de résultats intermédiaires

Pour plus d'informations, consultez la section [Transformation des données](#).

Résolution des problèmes liés aux intégrations open source

Pour plus d'informations sur les sujets de résolution des problèmes liés aux intégrations open source pour les passerelles SiteWise Edge, consultez. [Résolution des problèmes liés aux intégrations open source à la périphérie](#)

Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise

Découvrez les fonctionnalités et les limites des flux classiques, des passerelles V2 pour AWS IoT SiteWise Edge.

La passerelle Classic Streams, V2 conserve les fonctionnalités traditionnelles connues lors des AWS IoT SiteWise déploiements antérieurs avant l'introduction des passerelles V3 compatibles MQTT. Ces passerelles SiteWise Edge sont considérées comme des flux classiques, des passerelles V2. Ils sont rétrocompatibles et fonctionnent avec le pack de traitement des données. Bien que la passerelle Classic diffuse, la passerelle V2 offre des performances fiables pour les configurations existantes, elle présente des limites par rapport aux options de passerelle plus récentes. Plus précisément, ce type de passerelle n'est pas entièrement compatible avec les fonctionnalités avancées disponibles dans la destination de passerelle V3 compatible MQTT. Pour utiliser le protocole de messagerie MQTT, vous pouvez créer une nouvelle passerelle V3 compatible MQTT. Pour de plus amples informations, veuillez consulter [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#).

Rubriques

- [Utiliser des packs pour collecter et traiter des données dans SiteWise Edge](#)
- [Configuration du composant AWS IoT SiteWise éditeur](#)
- [Gestionnaire de destinations et de AWS IoT Greengrass flux](#)
- [Configuration des fonctionnalités Edge sur AWS IoT SiteWise Edge](#)
- [Configuration du traitement des données de pointe pour les AWS IoT SiteWise modèles et les actifs](#)

Utiliser des packs pour collecter et traiter des données dans SiteWise Edge

AWS IoT SiteWise Les passerelles Edge utilisent différents packs pour déterminer comment collecter et traiter vos données.

Les packs suivants sont actuellement disponibles :

- Pack de collecte de données : utilisez ce pack pour collecter vos données industrielles et les acheminer vers des destinations AWS cloud. Par défaut, ce pack est activé automatiquement pour votre passerelle SiteWise Edge.
- Pack de traitement des données : utilisez ce pack pour permettre la communication par passerelle SiteWise Edge avec des modèles d'actifs et des actifs configurés en périphérie. Vous pouvez utiliser la configuration périphérique pour contrôler les données d'actifs à calculer et à traiter sur site. Vous pouvez ensuite envoyer vos données à AWS IoT SiteWise d'autres AWS services. Pour plus d'informations sur le pack de traitement des données, consultez [the section called "Configuration du traitement des données de pointe"](#).

Packs d'amélioration

Important

La mise à niveau des versions du pack de traitement de données antérieures (et incluses) à la version 2.1.x vers la version 2.1.x entraînera une perte de données des mesures stockées localement.

SiteWise Les passerelles Edge utilisent différents packs pour déterminer comment collecter et traiter vos données. Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à niveau les packs.

Pour mettre à niveau des packs (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Dans la liste des passerelles, choisissez la passerelle SiteWise Edge contenant les packs que vous souhaitez mettre à niveau.
4. Dans la section Configuration de la passerelle, sélectionnez Mises à jour logicielles disponibles.
5. Sur la page Modifier les versions du logiciel, sélectionnez Mises à jour.

Note

Vous ne pouvez mettre à niveau que les packs qui sont activés. Pour trouver la liste des packs activés pour cette passerelle SiteWise Edge, choisissez Overview, puis consultez la section Fonctionnalités Edge.

6. Sur la page de modification des versions du logiciel, dans la section Mises à jour des composants Gateway, procédez comme suit :
 - Pour mettre à jour le collecteur OPC UA, choisissez une version, puis choisissez Deploy.
 - Pour mettre à jour l'éditeur, choisissez une version, puis choisissez Déployer.
 - Pour mettre à jour le pack de traitement des données, choisissez une version, puis choisissez Déployer.
7. Lorsque vous avez terminé de déployer les nouvelles versions, choisissez OK.

Si vous rencontrez des problèmes lors de la mise à niveau des packs, consultez [Impossible de déployer des packs sur les passerelles SiteWise Edge](#).

Configuration du composant AWS IoT SiteWise éditeur

Après avoir créé une passerelle AWS IoT SiteWise Edge et installé le logiciel, vous pouvez configurer le composant éditeur afin que votre passerelle SiteWise Edge puisse exporter des données vers le AWS cloud. Utilisez le composant éditeur pour activer des fonctionnalités supplémentaires ou configurer les paramètres par défaut. Pour plus d'informations, consultez [AWS IoT SiteWise l'éditeur](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Note

La configuration de l'éditeur varie en fonction du type de passerelle que vous utilisez. Pour les passerelles Classic Stream, V2, utilisez l'espace de noms `iotsitewise:publisher:2`. Pour les passerelles V3 compatibles MQTT, utilisez l'espace de noms `iotsitewise:publisher:3`.

Console

1. Accédez à la [console AWS IoT SiteWise](#).

2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle SiteWise Edge pour laquelle vous souhaitez configurer l'éditeur.
4. Dans la section Configuration de l'éditeur, choisissez Modifier
5. Pour Ordre de publication, sélectionnez l'une des options suivantes :
 - Publier les données les plus anciennes en premier — La passerelle SiteWise Edge publie d'abord les données les plus anciennes dans le cloud par défaut.
 - Publiez d'abord les données les plus récentes : la passerelle SiteWise Edge publie d'abord les données les plus récentes dans le cloud.
6. (Facultatif) Si vous ne souhaitez pas que la passerelle SiteWise Edge compresse vos données, désélectionnez Activer la compression lors du téléchargement des données.
7. (Facultatif) Si vous ne souhaitez pas publier d'anciennes données, choisissez Exclure les données expirées et procédez comme suit :
 - Pour Période limite, entrez une valeur et choisissez une unité. La période limite doit être comprise entre cinq minutes et sept jours. Par exemple, si la période limite est de trois jours, les données datant de plus de trois jours ne sont pas publiées dans le cloud.
8. (Facultatif) Pour définir des paramètres personnalisés concernant le traitement des données sur votre appareil local, choisissez Paramètres de stockage local et procédez comme suit :
 - a. Pour Période de rétention, entrez un nombre et choisissez une unité. La période de conservation doit être comprise entre une minute et 30 jours, et être supérieure ou égale à la période de rotation. Par exemple, si la période de conservation est de 14 jours, la passerelle SiteWise Edge supprime toutes les données de la périphérie qui sont antérieures à la période limite spécifiée après les avoir stockées pendant 14 jours.
 - b. Pour Période de rotation, entrez un nombre et choisissez une unité. La période de rotation doit être supérieure à une minute et égale ou inférieure à la période de conservation. Par exemple, supposons que la période de rotation soit de deux jours, la passerelle SiteWise Edge regroupe et enregistre les données antérieures à la période limite dans un seul fichier. Pour les passerelles auto-hébergées AWS IoT Greengrass V2, la passerelle SiteWise Edge transfère un lot de données vers le répertoire local suivant une fois tous les deux jours : `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`
 - c. Pour Capacité de stockage, entrez une valeur supérieure ou égale à 1. Si la capacité de stockage est de 2 Go, la passerelle SiteWise Edge commence à supprimer les données lorsque plus de 2 Go de données sont stockées localement.

9. Choisissez Enregistrer.

AWS CLI

Utilisez l'[UpdateGatewayCapabilityConfiguration](#) API pour configurer l'éditeur.

Définissez le paramètre `capabilityNamespace` sur `iotsitewise:publisher:2`.

Exemple : Configuration de l'éditeur pour les passerelles Classic Stream, V2

L'espace de noms de l'éditeur : `iotsitewise:publisher:2`

```
{
  "SiteWisePublisherConfiguration": {
    "publishingOrder": "TIME_ORDER",
    "enableCompression": true,
    "dropPolicy": {
      "cutoffAge": "7d",
      "exportPolicy": {
        "retentionPeriod": "7d",
        "rotationPeriod": "6h",
        "exportSizeLimitGB": 10
      }
    }
  },
  "SiteWiseS3PublisherConfiguration": {
    "accessRoleArn": "arn:aws:iam:123456789012:role/roleName",
    "streamToS3ConfigMapping": [
      {
        "streamName": "S3_OPC-UA_Data_Collector",
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/dataCollector",
        "publishPolicy": {
          "publishFrequency": "10m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,
          "enableDeleteAfterImport": true
        }
      }
    ]
  }
}
```


L'éditeur fournit les paramètres de configuration suivants que vous pouvez personnaliser :

SiteWisePublisherConfiguration

publishingOrder

Ordre dans lequel les données sont publiées dans le cloud. La valeur de ce paramètre peut être l'une des suivantes :

- **TIME_ORDER**(Publiez d'abord les données les plus anciennes) — Les données les plus anciennes sont publiées dans le cloud en premier, par défaut.
- **RECENT_DATA**(Publiez d'abord les données les plus récentes) — Les données les plus récentes sont d'abord publiées dans le cloud.

enableCompression

Définissez ce paramètre sur `true` pour compresser les données avant de les publier. La compression des données peut réduire l'utilisation de la bande passante.

dropPolicy

(Facultatif) Une politique qui contrôle les données publiées dans le cloud.

cutoffAge

L'âge maximal des données à publier est indiqué en jours, heures et minutes. Par exemple, 7d ou 1d7h16m. Les données plus anciennes que celles que vous spécifiez ne sont pas envoyées à AWS IoT SiteWise.

Les données antérieures à la période limite ne sont pas publiées dans le cloud. L'âge limite doit être compris entre cinq minutes et sept jours.

Vous pouvez utiliser `mh`, et `d` lorsque vous spécifiez un âge limite. Notez que cela `m` `h` représente les minutes, les heures et les `d` jours.

exportPolicy

(Facultatif) Une politique qui gère le stockage des données à la périphérie. Cette politique s'applique aux données antérieures à l'âge limite.

retentionPeriod

Votre passerelle SiteWise Edge supprime toutes les données de la périphérie antérieures à la période limite du stockage local une fois qu'elles ont été stockées pendant la période de conservation spécifiée. La période de conservation doit être

comprise entre une minute et 30 jours, et être supérieure ou égale à la période de rotation.

Vous pouvez utiliser mh, et d lorsque vous spécifiez une période de conservation. Notez que cela m h représente les minutes, les heures et les d jours.

`rotationPeriod`

Intervalle de temps pendant lequel les données antérieures à la période limite doivent être regroupées et enregistrées dans un seul fichier. La passerelle SiteWise Edge transfère un lot de données vers le répertoire local suivant à la fin de chaque période de rotation : `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`. La période de rotation doit être supérieure à une minute et égale ou inférieure à la période de conservation.

Vous pouvez utiliser mh, et d lorsque vous spécifiez une période de rotation. Notez que cela m h représente les minutes, les heures et les d jours.

`exportSizeLimitGB`

Taille maximale autorisée des données stockées localement, en Go. Si ce quota est dépassé, la passerelle SiteWise Edge commence à supprimer les données les plus anciennes jusqu'à ce que la taille des données stockées localement soit égale ou inférieure au quota. La valeur de ce paramètre doit être supérieure ou égale à 1.

`SiteWiseS3PublisherConfiguration`

`accessRoleArn`

Rôle d'accès qui AWS IoT SiteWise autorise la gestion du compartiment Amazon S3 dans lequel vous publiez.

`streamToS3ConfigMapping`

Un ensemble de configurations qui mappe un flux vers une configuration Amazon S3.

`streamName`

Le flux à lire et à publier dans la configuration Amazon S3.

`targetBucketArn`

Le compartiment ARN dans lequel publier.

`publishPolicy`

`publishFrequency`

Fréquence à laquelle la passerelle SiteWise Edge publie dans le compartiment Amazon S3.

`localSizeLimitGB`

Taille maximale des fichiers écrits sur le disque local. Si ce seuil est dépassé, l'éditeur publie toutes les données mises en mémoire tampon vers leur destination.

`siteWiseImportPolicy`

`enableSiteWiseStorageImport`

Définissez ce paramètre sur `true` pour importer les données d'un compartiment Amazon S3 vers le AWS IoT SiteWise stockage.

`enableDeleteAfterImport`

Définissez cette option `true` pour supprimer le fichier dans le compartiment Amazon S3 après son ingestion dans le AWS IoT SiteWise stockage.

Gestionnaire de destinations et de AWS IoT Greengrass flux

AWS IoT Greengrass Le gestionnaire de flux vous permet d'envoyer des données vers les AWS Cloud destinations suivantes : canaux AWS IoT Analytics entrants, flux dans Amazon Kinesis Data Streams, propriétés des actifs ou objets AWS IoT SiteWise dans Amazon Simple Storage Service (Amazon S3). Pour plus d'informations, voir [Gérer les flux de données sur le AWS IoT Greengrass Core](#) dans le Guide AWS IoT Greengrass Version 2 du développeur.

Exemple : Structure des messages du flux de données

L'exemple suivant montre la structure de message de flux de données requise transmise par le gestionnaire de AWS IoT Greengrass flux.

```
{
  "assetId": "string",
  "propertyAlias": "string",
  "propertyId": "string",
  "propertyValues": [
    {
      "quality": "string",
```

```
    "timestamp": {
      "offsetInNanos": number,
      "timeInSeconds": number
    },
    "value": {
      "booleanValue": boolean,
      "doubleValue": number,
      "integerValue": number,
      "stringValue": "string"
    }
  }
]
```

Note

Le message du flux de données doit inclure (`assetId` et `propertyId`) ou `propertyAlias` dans sa structure.

`assetId`

(Facultatif) L'ID de la ressource à mettre à jour.

`propertyAlias`

(Facultatif) Alias identifiant la propriété, tel qu'un chemin de flux de données du serveur OPC UA. Par exemple :

```
/company/windfarm/3/turbine/7/temperature
```

Pour plus d'informations, consultez la section [Gérer les flux de données](#) dans le guide de AWS IoT SiteWise l'utilisateur.

`propertyId`

(Facultatif) ID de la propriété de l'actif pour cette entrée.

`propertyValues`

(Obligatoire) La liste des valeurs de propriétés à télécharger. Vous pouvez spécifier jusqu'à 10 éléments `propertyValues` de tableau.

quality

(Facultatif) La qualité de la valeur de la propriété de l'actif.

timestamp

(Obligatoire) Horodatage de la valeur de la propriété de l'actif.

offsetInNanos

(Facultatif) Le décalage de nanosecondes par rapport à `timeInSeconds`.

timeInSeconds

(Obligatoire) Date d'horodatage, en secondes, au format Unix Epoch. Les données de nanosecondes fractionnaires sont fournies par `offsetInNanos`.

value

(Obligatoire) La valeur de la propriété de l'actif.

Note

Seule l'une des valeurs suivantes peut exister dans le `value` champ.

booleanValue

(Facultatif) Données de propriété de l'actif de type booléen (`true` ou `false`).

doubleValue

(Facultatif) Données de propriété des actifs de type double (nombre à virgule flottante).

integerValue

(Facultatif) Données de propriété de l'actif de type entier (nombre entier).

stringValue

(Facultatif) Données de propriété de l'actif de type chaîne (séquence de caractères).

Configuration des fonctionnalités Edge sur AWS IoT SiteWise Edge

Vous pouvez utiliser AWS IoT SiteWise Edge pour collecter et stocker temporairement des données afin de pouvoir organiser et traiter les données des appareils localement. En activant le traitement

périphérique, vous pouvez choisir de n'envoyer que des données agrégées au AWS Cloud afin d'optimiser votre utilisation de la bande passante et les coûts de stockage dans le cloud. À l'aide de AWS IoT SiteWise composants AWS IoT Greengrass, vous pouvez collecter et traiter les données à la périphérie avant de les envoyer au AWS Cloud, ou les gérer sur site à l'aide d' SiteWise Edge APIs.

La collecte de données s'effectue par le biais de packs de données et de AWS IoT SiteWise composants qui s'exécutent sur AWS IoT Greengrass.

Note

- AWS IoT SiteWise conserve vos données Edge sur vos passerelles SiteWise Edge jusqu'à 30 jours. La durée de conservation de vos données dépend de l'espace disque disponible sur votre appareil.
- Si votre passerelle SiteWise Edge est déconnectée du AWS Cloud depuis 30 jours, le [pack de traitement des données](#) est automatiquement désactivé.

Rubriques

- [Configurer la fonctionnalité Edge dans SiteWise Edge](#)

Configurer la fonctionnalité Edge dans SiteWise Edge

AWS IoT SiteWise fournit les packs suivants que votre passerelle SiteWise Edge peut utiliser pour déterminer comment collecter et traiter vos données. Sélectionnez des packs pour activer les fonctionnalités Edge de votre passerelle SiteWise Edge.

- Le pack de collecte de données permet à votre passerelle SiteWise Edge de collecter des données à partir de plusieurs serveurs OPC UA, puis de les exporter de la périphérie vers le AWS cloud. Il devient actif une fois que vous avez ajouté des sources de données à votre passerelle SiteWise Edge.
- Le pack de traitement des données permet à votre passerelle SiteWise Edge de traiter les données de votre équipement en périphérie. Par exemple, vous pouvez utiliser des modèles d'actifs pour calculer des métriques et des transformations. Pour plus d'informations sur les modèles d'actifs et les actifs, consultez [Modéliser des actifs industriels](#).

 Note

- Le pack de traitement des données n'est disponible que sur les plateformes x86.

Pour configurer les fonctionnalités de pointe

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle SiteWise Edge pour laquelle vous souhaitez activer les fonctionnalités Edge.
4. Dans la section Fonctionnalités Edge, choisissez Modifier
5. Dans la section Fonctionnalités Edge, sélectionnez Activer le pack de traitement des données (entraîne des frais supplémentaires).
6. (Facultatif) Dans la section Connexion LDAP Edge, vous pouvez autoriser les groupes d'utilisateurs de votre annuaire d'entreprise à accéder à cette passerelle SiteWise Edge. Les groupes d'utilisateurs peuvent utiliser les informations d'identification LDAP (Lightweight Directory Access Protocol) pour accéder à la passerelle SiteWise Edge. Ils peuvent ensuite utiliser l' AWS IoT SiteWise application AWS OpsHub for, les opérations d' AWS IoT SiteWise API ou d'autres outils pour gérer la passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Gérer les passerelles SiteWise Edge](#).

 Note

Vous pouvez également utiliser Linux ou des Microsoft Windows informations d'identification pour accéder à la passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification du système d'exploitation Linux](#).

- a. Sélectionnez Activé.
- b. Dans Nom du fournisseur, entrez le nom de votre fournisseur LDAP.
- c. Dans Nom d'hôte ou adresse IP, entrez le nom d'hôte ou l'adresse IP de votre serveur LDAP.
- d. Pour Port, entrez un numéro de port.

- e. Pour Nom distinctif de base (DN), entrez un nom distinctif (DN) pour la base.

Les types d'attributs suivants sont pris en charge : CommonName (CN), LocalityName (L), Name (ST) stateOrProvince, OrganizationName (O), (OU), CountryName (C) organizationalUnitName , StreetAddress (STREET), DomainComponent (DC) et userid (UID).

- f. Pour le DN du groupe d'administrateurs, entrez un DN.
- g. Pour le DN du groupe d'utilisateurs, entrez un DN.

7. Choisissez Enregistrer.

Maintenant que vous avez activé les fonctionnalités Edge sur votre passerelle SiteWise Edge, vous devez configurer votre modèle d'actif pour la périphérie. La configuration périphérique de votre modèle d'actifs indique où les propriétés de vos actifs sont calculées. Vous pouvez calculer toutes les propriétés à la périphérie ou configurer les propriétés de votre modèle d'actifs séparément. Les propriétés du modèle d'actifs incluent [les métriques](#), les [transformations](#) et [les mesures](#).

Pour plus d'informations sur les propriétés des actifs, consultez [the section called “Définir les propriétés des données”](#).

Après avoir créé votre modèle d'actif, vous pouvez le configurer pour la périphérie. Pour plus d'informations sur la configuration de votre modèle d'actif pour la périphérie, consultez [the section called “Création d'un modèle d'actif \(console\)”](#).

Note

Les modèles d'actifs et les tableaux de bord sont automatiquement synchronisés entre le AWS Cloud et votre passerelle SiteWise Edge toutes les 10 minutes. Vous pouvez également effectuer une synchronisation manuelle à partir de l'application de passerelle SiteWise Edge locale.

Configuration du traitement des données de pointe pour les AWS IoT SiteWise modèles et les actifs

Vous pouvez utiliser AWS IoT SiteWise Edge pour collecter, stocker, organiser et surveiller les données des équipements localement. Vous pouvez utiliser SiteWise Edge pour modéliser vos données industrielles et SiteWise Monitor pour créer des tableaux de bord permettant à votre

personnel opérationnel de visualiser les données localement. Vous pouvez traiter vos données localement et les envoyer vers le AWS cloud, ou les traiter sur site à l'aide de l' AWS IoT SiteWise API.

Avec AWS IoT SiteWise Edge, vous pouvez traiter les données brutes localement et choisir de n'envoyer que des données agrégées vers le AWS cloud afin d'optimiser votre utilisation de la bande passante et les coûts de stockage dans le cloud.

 Note

- AWS IoT SiteWise conserve vos données Edge sur vos passerelles SiteWise Edge jusqu'à 30 jours. La durée de conservation de vos données dépend de l'espace disque disponible sur votre appareil.
- Si votre passerelle SiteWise Edge est déconnectée du AWS Cloud depuis 30 jours, elle [Configurer une source OPC UA dans Edge SiteWise](#) est automatiquement désactivée.

Configuration d'un modèle d'actif pour le traitement des données sur SiteWise Edge

Vous devez configurer votre modèle d'actif pour la périphérie avant de pouvoir traiter les données de votre passerelle SiteWise Edge à la périphérie. La configuration périphérique de votre modèle d'actifs indique où les propriétés de vos actifs sont calculées. Vous pouvez choisir de calculer toutes les propriétés à la périphérie et d'envoyer les résultats vers le AWS cloud, ou de personnaliser l'endroit où calculer chaque propriété d'actif séparément. Pour de plus amples informations, veuillez consulter [Configuration du traitement des données de pointe pour les AWS IoT SiteWise modèles et les actifs](#).

Les propriétés des actifs incluent les métriques, les transformations et les mesures :

- Les métriques sont les données agrégées de l'actif sur une période donnée. Vous pouvez calculer de nouvelles mesures en utilisant les données métriques existantes. AWS IoT SiteWise envoie toujours vos indicateurs vers le AWS Cloud pour un stockage à long terme. AWS IoT SiteWise calcule les métriques sur le AWS Cloud par défaut. Vous pouvez configurer votre modèle d'actifs pour calculer vos indicateurs à la périphérie. AWS IoT SiteWise envoie les résultats traités au AWS Cloud.
- Les transformations sont des expressions mathématiques qui mappent les points de données d'une propriété de ressource à partir d'un formulaire vers un autre. Les transformations peuvent utiliser des métriques comme données d'entrée et doivent être calculées et stockées au même

endroit que leurs entrées. Si vous configurez une entrée métrique pour qu'elle soit calculée à la périphérie, calcule AWS IoT SiteWise également la transformation associée à la périphérie.

- Les mesures sont formatées sous forme de données brutes que votre appareil collecte et envoie au AWS Cloud par défaut. Vous pouvez configurer votre modèle d'actif pour stocker ces données sur votre appareil local.

Pour plus d'informations sur les propriétés des actifs, consultez [the section called “Définir les propriétés des données”](#).

Après avoir créé votre modèle d'actif, vous pouvez le configurer pour la périphérie. Pour plus d'informations sur la configuration de votre modèle d'actif pour la périphérie, consultez [the section called “Création d'un modèle d'actif \(console\)”](#).

 Note

Les modèles d'actifs et les tableaux de bord sont automatiquement synchronisés entre le AWS Cloud et votre passerelle SiteWise Edge toutes les 10 minutes. Vous pouvez également effectuer une synchronisation manuelle à partir du [Gérer les passerelles SiteWise Edge](#).

Vous pouvez utiliser le AWS IoT SiteWise REST APIs et le AWS Command Line Interface (AWS CLI) pour interroger votre passerelle SiteWise Edge pour obtenir des données en périphérie. Avant d'interroger votre passerelle SiteWise Edge pour obtenir des données en périphérie, vous devez remplir les conditions préalables suivantes :

- Vos informations d'identification doivent être définies pour le REST APIs. Pour plus d'informations sur la définition des informations d'identification, consultez [the section called “Gérez les passerelles”](#).
- Le point de terminaison du SDK doit pointer vers l'adresse IP de votre passerelle SiteWise Edge. Vous trouverez de plus amples informations dans la documentation de votre SDK. Par exemple, consultez la section [Spécification de points de terminaison personnalisés](#) dans le guide du AWS SDK for Java 2.x développeur.
- Votre certificat de passerelle SiteWise Edge doit être enregistré. Vous trouverez plus d'informations sur l'enregistrement de votre certificat de passerelle SiteWise Edge dans la documentation de votre SDK. Par exemple, consultez [l'enregistrement des ensembles de certificats dans le fichier Node.js](#) du guide du AWS SDK for Java 2.x développeur.

Pour plus d'informations sur l'interrogation de données avec AWS IoT SiteWise, consultez [Interrogez les données de AWS IoT SiteWise](#).

Ajoutez des sources de données à votre passerelle AWS IoT SiteWise Edge

Après avoir configuré une passerelle AWS IoT SiteWise Edge, vous pouvez ajouter et configurer des sources de données pour ingérer les données des équipements industriels locaux vers AWS IoT SiteWise. SiteWise Edge prend en charge différents protocoles, notamment OPC UA, et de nombreux autres protocoles disponibles via des sources de données partenaires. Ces sources permettent à votre passerelle de se connecter à des serveurs locaux et de récupérer vos données industrielles. En configurant les sources de données, vous pouvez ingérer des données provenant de diverses sources de données, puis associer les flux de données aux propriétés des actifs, ce qui permet une modélisation complète des actifs industriels et un mappage des données. AWS IoT SiteWise

Rubriques

- [Sources de données OPC UA pour les passerelles AWS IoT SiteWise Edge](#)
- [Sources de données partenaires sur les passerelles SiteWise Edge](#)

Sources de données OPC UA pour les passerelles AWS IoT SiteWise Edge

Après avoir configuré une passerelle AWS IoT SiteWise Edge, vous pouvez configurer des sources de données afin que votre passerelle SiteWise Edge puisse ingérer des données provenant d'équipements industriels locaux vers AWS IoT SiteWise. Chaque source représente un serveur local, tel qu'un serveur OPC UA, auquel votre passerelle SiteWise Edge connecte et récupère les flux de données industriels. Pour plus d'informations sur la configuration d'une passerelle SiteWise Edge, consultez [Créez une passerelle SiteWise Edge auto-hébergée](#).

Le type de passerelle, compatible MQTT, passerelles V3 par rapport au flux classique, passerelles V2, influence la façon dont les données OPC UA sont traitées. Dans le stream classique, les passerelles V2 et les sources de données OPC UA sont ajoutées directement à la configuration de l' SiteWise éditeur IoT de la passerelle. Chaque source de données est couplée à la passerelle, et le routage des données est configuré individuellement pour chaque source. En revanche, à l'aide de passerelles V3 compatibles MQTT, les sources de données OPC UA sont converties en sujets MQTT et gérées via des destinations centralisées. Pour plus d'informations sur chaque type, reportez-vous

aux sections [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#) et [Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise](#).

 Note

AWS IoT SiteWise redémarre votre passerelle SiteWise Edge chaque fois que vous ajoutez ou modifiez une source. Votre passerelle SiteWise Edge n'ingère pas de données lors de la mise à jour de la configuration de la source. Le délai de redémarrage de votre passerelle SiteWise Edge dépend du nombre de balises figurant sur les sources de votre passerelle SiteWise Edge. Le temps de redémarrage peut aller de quelques secondes (pour une passerelle SiteWise Edge avec peu de balises) à plusieurs minutes (pour une passerelle SiteWise Edge avec de nombreuses balises).

Après avoir créé les sources, vous pouvez associer vos flux de données aux propriétés des actifs. Pour de plus amples informations sur la création et l'utilisation des ressources, veuillez consulter [Modéliser des actifs industriels](#).

Vous pouvez consulter CloudWatch les métriques pour vérifier qu'une source de données est connectée à AWS IoT SiteWise. Pour de plus amples informations, veuillez consulter [AWS IoT Greengrass Version 2 métriques de passerelle](#).

Actuellement, AWS IoT SiteWise prend en charge les protocoles de source de données suivants :

- [OPC UA](#) — Protocole de communication machine-to-machine (M2M) pour l'automatisation industrielle.

Support pour des protocoles industriels supplémentaires

SiteWise Edge prend en charge un large éventail de protocoles industriels grâce à l'intégration avec des partenaires de sources de données. Ces partenariats permettent la connectivité avec plus de 200 protocoles différents, adaptés à divers systèmes et appareils industriels.

Pour obtenir la liste des partenaires de sources de données disponibles, consultez [SiteWise Options de source de données pour les partenaires Edge Gateway](#).

Configurer une source OPC UA dans Edge SiteWise

Vous pouvez utiliser la AWS IoT SiteWise console ou une fonctionnalité de passerelle SiteWise Edge pour définir et ajouter une source OPC UA à votre passerelle SiteWise Edge afin de représenter un serveur OPC UA local.

Rubriques

- [Configuration d'une source OPC UA \(console\)](#)
- [Configurer une source OPC UA \(AWS CLI\)](#)

Configuration d'une source OPC UA (console)

Vous pouvez utiliser la console pour configurer la source OPC UA en suivant la procédure suivante.

Note

Avertissement : la duplication TQVs peut entraîner une double charge.

Pour configurer une source OPC UA à l'aide de la console AWS IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Sélectionnez la passerelle SiteWise Edge pour ajouter une source OPC UA.
4. Choisissez Add data source.
5. Entrez le nom de la source.
6. Entrez le point de terminaison local du serveur de source de données. Le point de terminaison peut être l'adresse IP ou le nom d'hôte. Vous pouvez également ajouter un numéro de port au point de terminaison local. Par exemple, votre point de terminaison local peut ressembler à ceci : **opc.tcp://203.0.113.0:49320**
7. (Facultatif) Pour l'ID de nœud à sélectionner, ajoutez des filtres de nœuds pour limiter les flux de données ingérés AWS dans le cloud. Par défaut, les passerelles SiteWise Edge utilisent le nœud racine d'un serveur pour ingérer tous les flux de données. Vous pouvez utiliser des filtres de nœuds pour réduire le temps de démarrage et l'utilisation du processeur de votre passerelle SiteWise Edge en incluant uniquement les chemins d'accès aux données que vous modélisez AWS IoT SiteWise. Par défaut, les passerelles SiteWise Edge téléchargent tous les chemins

OPC UA à l'exception de ceux qui commencent par. /Server/ Pour définir les filtres de nœuds OPC UA, vous pouvez utiliser les chemins des nœuds * et les ** caractères génériques. Pour de plus amples informations, veuillez consulter [Utiliser les filtres de nœuds OPC UA dans Edge SiteWise](#).

8. Les destinations varient entre les passerelles V3 compatibles MQTT et les flux classiques, les passerelles V2.
 - Streams classiques, les destinations de passerelle V2 entretiennent une relation 1:1 avec la source. Chaque source envoie des données vers une destination spécifique.
 - Les destinations de passerelle V3 compatibles MQTT sont configurées séparément car le modèle hub and spoke vous permet de centraliser la configuration et la gestion de plusieurs sources de données sur différentes passerelles. Pour configurer des destinations dans une passerelle V3, voir [Comprendre les destinations AWS IoT SiteWise Edge](#).

Classic streams, V2 gateway destinations

- AWS IoT SiteWise en temps réel — Choisissez cette option pour envoyer les données directement au AWS IoT SiteWise stockage. Ingérez et surveillez les données en temps réel à la périphérie.
- AWS IoT SiteWise Mise en mémoire tampon à l'aide d'Amazon S3 : envoyez les données au format Parquet vers Amazon S3, puis importez-les dans le AWS IoT SiteWise stockage. Choisissez cette option pour ingérer les données par lots et stocker les données historiques de manière rentable. Vous pouvez configurer l'emplacement de votre compartiment Amazon S3 préféré et la fréquence à laquelle vous souhaitez que les données soient chargées sur Amazon S3. Vous pouvez également choisir ce que vous souhaitez faire avec les données après leur ingestion dans AWS IoT SiteWise. Vous pouvez choisir que les données soient disponibles à la fois dans Amazon S3 AWS IoT SiteWise et dans Amazon S3 ou vous pouvez choisir de les supprimer automatiquement d'Amazon S3 après leur importation dans Amazon S3 AWS IoT SiteWise.
- Le compartiment Amazon S3 est un mécanisme de préparation et de mise en mémoire tampon qui prend en charge les fichiers au format Parquet.
- Si vous cochez la case Importer les données dans le AWS IoT SiteWise stockage, les données sont d'abord chargées dans Amazon S3, puis dans le AWS IoT SiteWise stockage.
 - Si vous cochez la case Supprimer les données d'Amazon S3, les données sont supprimées d'Amazon S3 après leur importation dans le SiteWise stockage.

- Si vous décochez la case Supprimer les données d'Amazon S3, les données sont stockées à la fois dans Amazon S3 et dans le SiteWise stockage.
- Si vous décochez la case Importer les données dans le AWS IoT SiteWise stockage, les données sont stockées uniquement dans Amazon S3. Il n'est pas importé dans le SiteWise stockage.

Visitez [Gérer le stockage des données](#) pour plus de détails sur les différentes options de stockage AWS IoT SiteWise proposées. Pour en savoir plus sur les options de tarification, consultez la section [AWS IoT SiteWise Tarification](#).

- AWS IoT Greengrass gestionnaire de flux — Utilisez le gestionnaire de AWS IoT Greengrass flux pour envoyer des données vers les destinations AWS cloud suivantes : canaux AWS IoT Analytics entrants, flux dans Amazon Kinesis Data Streams, propriétés des actifs ou objets AWS IoT SiteWise dans Amazon Simple Storage Service (Amazon S3). Pour plus d'informations, voir [Gérer les flux de données sur le AWS IoT Greengrass Core](#) dans le Guide AWS IoT Greengrass Version 2 du développeur.

Entrez un nom pour le AWS IoT Greengrass flux.

MQTT-enabled, V3 gateway destinations

1. Consultez [Passerelles V3 compatibles MQTT pour Edge AWS IoT SiteWise](#) pour plus d'informations sur l'ajout de vos destinations pertinentes.
 2. Revenez à cette procédure après avoir ajouté vos destinations sources.
9. Dans le volet Configuration avancée, vous pouvez effectuer les opérations suivantes :
- a. Choisissez un mode de sécurité des messages pour les connexions et les données en transit entre votre serveur source et votre passerelle SiteWise Edge. Ce champ est la combinaison de la politique de sécurité OPC UA et du mode de sécurité des messages. Choisissez la même politique de sécurité et le même mode de sécurité des messages que ceux que vous avez spécifiés pour votre serveur OPC UA.
 - b. Si votre source nécessite une authentification, choisissez un AWS Secrets Manager secret dans la liste de configuration de l'authentification. La passerelle SiteWise Edge utilise les informations d'authentification contenues dans ce secret lorsqu'elle se connecte à cette source de données. Vous devez associer des secrets au AWS IoT Greengrass composant de votre passerelle SiteWise Edge pour les utiliser pour l'authentification des

sources de données. Pour de plus amples informations, veuillez consulter [the section called “Configuration de l'authentification des sources de données”](#).

 Tip

Votre serveur de données peut avoir une option nommée Autoriser la connexion anonyme. Si cette option est Oui, votre source n'a pas besoin d'authentification.

- c. (Facultatif) Vous pouvez activer un préfixe de flux de données en sélectionnant Activer le préfixe de flux de données - facultatif.
 - Entrez un préfixe de flux de données. La passerelle SiteWise Edge ajoute ce préfixe à tous les flux de données provenant de cette source. Utilisez un préfixe de flux de données pour distinguer les flux de données portant le même nom mais provenant de sources différentes. Chaque flux de données doit avoir un nom unique dans votre compte.
- d. (Facultatif) Choisissez une option de conversion de type de données pour convertir les types de données OPC UA non pris en charge en chaînes avant de les ingérer. AWS IoT SiteWise Convertissez des valeurs de tableau contenant des types de données simples en chaînes JSON et DateTime des types de données en chaînes ISO 8601. Pour de plus amples informations, veuillez consulter [Conversion de types de données non pris en charge](#).
- e. (Facultatif) Pour les groupes de propriétés, choisissez Ajouter un nouveau groupe.
 - i. Entrez un nom pour le groupe de propriétés.
 - ii. Pour les propriétés :
 - 1. Pour les chemins de nœud, ajoutez des filtres de nœuds OPC UA pour limiter les chemins OPC UA vers lesquels le téléchargement est effectué. AWS IoT SiteWise Le format est similaire à celui de l'ID de nœud pour la sélection.
 - iii. Pour les paramètres de groupe, procédez comme suit :
 - 1. Pour le paramètre de qualité des données, choisissez le type de qualité de données que AWS IoT SiteWise Collector doit ingérer.
 - 2. Pour le réglage du mode de numérisation, configurez les propriétés d'abonnement standard à l'aide du mode de numérisation. Vous pouvez sélectionner S'abonner ou Sondage. Pour plus d'informations sur le mode de numérisation, consultez [the section called “Filtrer les plages d'ingestion de données”](#).

Subscribe

Pour envoyer chaque point de données

- i. Choisissez S'abonner et définissez les paramètres suivants :
 - A. [Déclencheur de modification des données](#) : condition qui déclenche une alerte de modification des données.
 - B. [Taille de la file d'attente d'abonnement](#) : profondeur de la file d'attente sur un serveur OPC-UA pour une métrique particulière dans laquelle les notifications relatives aux éléments surveillés sont mises en file d'attente.
 - C. [Intervalle de publication par abonnement](#) : intervalle (en millisecondes) du cycle de publication spécifié lors de la création de l'abonnement.
 - D. Intervalle entre les instantanés : facultatif : paramètre de délai d'expiration de la fréquence des instantanés pour garantir qu' AWS IoT SiteWise Edge ingère un flux constant de données.
 - E. Fréquence de numérisation : fréquence à laquelle vous souhaitez que la passerelle SiteWise Edge lise vos registres. AWS IoT SiteWise calcule automatiquement le taux de numérisation minimum autorisé pour votre passerelle SiteWise Edge.
 - F. Horodatage — L'horodatage à inclure dans vos points de données OPC UA. Vous pouvez utiliser l'horodatage du serveur ou celui de votre appareil.



Note

Utilisez la version 2.5.0 ou ultérieure du composant collecteur IoT SiteWise OPC UA. Si vous utilisez la fonction d'horodatage avec des versions antérieures, les mises à jour de configuration échouent. Pour de plus amples informations, veuillez consulter [Mettre à jour la version d'un AWS IoT SiteWise composant](#).

- ii. Dans les paramètres Deadband, configurez un type Deadband. Le type deadband contrôle les données que votre source vous AWS IoT SiteWise envoie et celles qu'elle supprime. Pour plus d'informations sur le paramètre Deadband, consultez [the section called “Filtrer les plages d'ingestion de données”](#).

- **Aucun** — Le serveur associé envoie tous les points de données pour ce groupe de propriétés.
 - **Pourcentage** : le serveur associé envoie uniquement les données qui se situent en dehors d'un pourcentage spécifié de la plage de données. Cette plage est calculée par le serveur en fonction des unités d'ingénierie minimales et maximales définies pour chaque nœud. Si le serveur ne prend pas en charge les pourcentages morts ou s'il n'a pas d'unités d'ingénierie définies, la passerelle calcule la plage en utilisant les valeurs minimale et maximale fournies ci-dessous.
 - **Absolu** — Le serveur associé envoie uniquement des données situées en dehors d'une plage spécifique.
- A. Définissez la valeur Deadband comme le pourcentage de la plage de données sur Deadband.
 - B. (Facultatif) Spécifiez un minimum et un maximum pour la plage de zone morte en utilisant Plage minimale - facultatif et Plage maximale - facultative.

Poll

Pour envoyer des points de données à un intervalle spécifique

- Choisissez Poll et définissez les paramètres suivants :
 - A. **Fréquence de numérisation** : fréquence à laquelle vous souhaitez que la passerelle SiteWise Edge lise vos registres. AWS IoT SiteWise calcule automatiquement le taux de numérisation minimum autorisé pour votre passerelle SiteWise Edge.
 - B. **Horodatage** — L'horodatage à inclure dans vos points de données OPC UA. Vous pouvez utiliser l'horodatage du serveur ou celui de votre appareil.



Note

Utilisez la version 2.5.0 ou ultérieure du composant collecteur IoT SiteWise OPC UA. Si vous utilisez la fonction d'horodatage avec des versions antérieures, les mises à jour de configuration

échouent. Pour de plus amples informations, veuillez consulter [Mettre à jour la version d'un AWS IoT SiteWise composant](#).

 Note

Les paramètres Deadband sont applicables lorsque vous avez sélectionné S'abonner dans les paramètres du mode Scan.

10. Choisissez Enregistrer.

Configurer une source OPC UA ()AWS CLI

Vous pouvez définir des sources de données OPC UA pour une passerelle SiteWise Edge à l'aide du AWS CLI. Pour ce faire, créez un fichier JSON de configuration des fonctionnalités OPC UA et utilisez la [update-gateway-capability-configuration](#) commande pour mettre à jour la configuration de la passerelle SiteWise Edge. Vous devez définir toutes vos sources OPC UA dans une configuration de fonctionnalité unique.

MQTT-enabled, V3 gateway

Cette fonctionnalité possède l'espace de noms suivant.

- `iotsitewise:opcuacollector:3`

```
{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
        | "AES128_SHA256_RSA0AEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
```

```

    "type": "Anonymous" | "Username",
    "usernameSecretArn": "string"
  },
  "nodeFilterRules": [
    {
      "action": "INCLUDE",
      "definition": {
        "type": "OpcUaRootPath",
        "rootPath": "string"
      }
    }
  ],
  "measurementDataStreamPrefix": "string",
  "typeConversions": {
    "array": "JsonArray",
    "datetime": "ISO8601String"
  },
  "destination": {
    {
      "type": "MQTT"
    }
  },
  "propertyGroups": [
    {
      "name": "string",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "string"
        }
      ],
      "deadband": {
        "type": "PERCENT" | "ABSOLUTE",
        "value": double,
        "eguMin": double,
        "eguMax": double,
        "timeoutMilliseconds": integer
      },
      "scanMode": {
        "type": "EXCEPTION" | "POLL",
        "rate": integer,
        "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
      }
    }
  ],

```

```

    "dataQuality": {
      "allowGoodQuality": true | false,
      "allowBadQuality": true | false,
      "allowUncertainQuality": true | false
    },
    "subscription": {
      "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
      "queueSize": integer,
      "publishingIntervalMilliseconds": integer,
      "snapshotFrequencyMilliseconds": integer
    }
  }
]
}
]
}

```

Classic streams, V2 gateway

Cette fonctionnalité possède l'espace de noms suivant.

- `iotsitewise:opcuacollector:2`

Syntaxe de demande

```

{
  "sources": [
    {
      "name": "string",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny" | "X509",
          "certificateBody": "string",
          "certificateChain": "string",
        },
        "endpointUri": "string",
        "securityPolicy": "NONE" | "BASIC128_RSA15" | "BASIC256" | "BASIC256_SHA256"
| "AES128_SHA256_RSA0AEP" | "AES256_SHA256_RSAPSS",
        "messageSecurityMode": "NONE" | "SIGN" | "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Anonymous" | "Username",
          "usernameSecretArn": "string"
        }
      }
    }
  ]
}

```

```

    },
    "nodeFilterRules": [
      {
        "action": "INCLUDE",
        "definition": {
          "type": "OpcUaRootPath",
          "rootPath": "string"
        }
      }
    ],
    "measurementDataStreamPrefix": "string",
    "typeConversions": {
      "array": "JsonArray",
      "datetime": "ISO8601String"
    },
    "destination": {
      "type": "StreamManager",
      "streamName": "string",
      "streamBufferSize": integer,
    },
    "propertyGroups": [
      {
        "name": "string",
        "nodeFilterRuleDefinitions": [
          {
            "type": "OpcUaRootPath",
            "rootPath": "string"
          }
        ],
      },
    ],
    "deadband": {
      "type": "PERCENT" | "ABSOLUTE",
      "value": double,
      "eguMin": double,
      "eguMax": double,
      "timeoutMilliseconds": integer
    },
    "scanMode": {
      "type": "EXCEPTION" | "POLL",
      "rate": integer,
      "timestampToReturn": "SOURCE_TIME" | "SERVER_TIME"
    },
    "dataQuality": {
      "allowGoodQuality": true | false,

```

```

        "allowBadQuality": true | false,
        "allowUncertainQuality": true | false
    },
    "subscription": {
        "dataChangeTrigger": "STATUS" | "STATUS_VALUE" |
"STATUS_VALUE_TIMESTAMP",
        "queueSize": integer,
        "publishingIntervalMilliseconds": integer,
        "snapshotFrequencyMilliseconds": integer
    }
}
]
}
]
}

```

Corps de la demande

sources

Liste des structures de définition de source OPC UA contenant chacune les informations suivantes :

name

Nom unique et convivial pour la source.

endpoint

Structure de point de terminaison contenant les informations suivantes :

certificateTrust

Structure de stratégie d'approbation de certificat contenant les informations suivantes :

type

Mode d'approbation de certificat pour la source. Sélectionnez l'une des méthodes suivantes :

- **TrustAny**— La passerelle SiteWise Edge fait confiance à n'importe quel certificat lorsqu'elle se connecte à la source OPC UA.
- **X509**— La passerelle SiteWise Edge fait confiance à un certificat X.509 lorsqu'elle se connecte à la source OPC UA. Si vous choisissez cette option, vous devez définir

`certificateBody` dans `certificateTrust`. Vous pouvez également définir `certificateChain` dans `certificateTrust`.

`certificateBody`

(Facultatif) Corps d'un certificat X.509.

Ce champ est obligatoire si vous choisissez X509 pour type dans `certificateTrust`.

`certificateChain`

(Facultatif) Chaîne de confiance pour un certificat X.509.

Ce champ n'est utilisé que si vous choisissez X509 pour type dans `certificateTrust`.

`endpointUri`

Point de terminaison local de la source OPC UA. Par exemple, votre point de terminaison local peut ressembler à `opc.tcp://203.0.113.0:49320`.

`securityPolicy`

La politique de sécurité à utiliser pour sécuriser les messages lus depuis la source OPC UA. Sélectionnez l'une des méthodes suivantes :

- `NONE`— La passerelle SiteWise Edge ne sécurise pas les messages provenant de la source OPC UA. Nous vous recommandons de choisir une autre politique de sécurité. Si vous choisissez cette option, vous devez également choisir `NONE` pour `messageSecurityMode`.
- `BASIC256_SHA256`— La politique `Basic256Sha256` de sécurité.
- `AES128_SHA256_RSA0AEP`— La politique `Aes128_Sha256_Rsa0aep` de sécurité.
- `AES256_SHA256_RSAPSS`— La politique `Aes256_Sha256_RsaPss` de sécurité.
- `BASIC128_RSA15`— (Obsolète) La politique de `Basic128Rsa15` sécurité est déconseillée dans la spécification OPC UA car elle n'est plus considérée comme sécurisée. Nous vous recommandons de choisir une autre politique de sécurité. Pour plus d'informations, consultez [Profile SecurityPolicy — Basic128Rsa15](#).
- `BASIC256`— (Obsolète) La politique de `Basic256` sécurité est déconseillée dans la spécification OPC UA car elle n'est plus considérée comme sécurisée. Nous vous

recommandons de choisir une autre politique de sécurité. Pour plus d'informations, voir [SecurityPolicy — Basic256](#).

⚠ Important

Si vous choisissez une politique de sécurité autre que `NONE`, vous devez choisir `SIGN` ou `SIGN_AND_ENCRYPT` pour `messageSecurityMode`. Vous devez également configurer votre serveur source pour qu'il fasse confiance à la passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Configurer les serveurs OPC UA pour qu'ils fassent confiance à la passerelle AWS IoT SiteWise Edge](#).

`messageSecurityMode`

Mode de sécurité des messages à utiliser pour sécuriser les connexions à la source OPC UA. Sélectionnez l'une des méthodes suivantes :

- `NONE`— La passerelle SiteWise Edge ne sécurise pas les connexions à la source OPC UA. Nous vous recommandons de choisir un autre mode de sécurité des messages. Si vous choisissez cette option, vous devez également choisir `NONE` pour `securityPolicy`.
- `SIGN`— Les données en transit entre la passerelle SiteWise Edge et la source OPC UA sont signées mais pas chiffrées.
- `SIGN_AND_ENCRYPT`— Les données en transit entre la passerelle et la source OPC UA sont signées et cryptées.

⚠ Important

Si vous choisissez un mode de sécurité des messages autre que `NONE`, vous devez en choisir un `securityPolicy` autre que `NONE`. Vous devez également configurer votre serveur source pour qu'il fasse confiance à la passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Configurer les serveurs OPC UA pour qu'ils fassent confiance à la passerelle AWS IoT SiteWise Edge](#).

`identityProvider`

Structure de fournisseur d'identité contenant les informations suivantes :

type

Type d'informations d'identification d'authentification requises par la source.

Sélectionnez l'une des méthodes suivantes :

- Anonymous— La source n'a pas besoin d'authentification pour se connecter.
- Username— La source a besoin d'un nom d'utilisateur et d'un mot de passe pour se connecter. Si vous choisissez cette option, vous devez définir `usernameSecretArn` dans `identityProvider`.

usernameSecretArn

(Facultatif) L'ARN d'un AWS Secrets Manager secret. La passerelle SiteWise Edge utilise les informations d'authentification contenues dans ce secret lorsqu'elle se connecte à cette source. Vous devez associer des secrets au SiteWise connecteur IoT de votre passerelle SiteWise Edge pour les utiliser à des fins d'authentification à la source. Pour de plus amples informations, veuillez consulter [Configurer l'authentification des sources de données pour SiteWise Edge](#).

Ce champ est obligatoire si vous choisissez Username pour type dans `identityProvider`.

nodeFilterRules

Liste des structures de règles de filtrage des nœuds qui définissent les chemins de flux de données OPC UA à envoyer vers le AWS cloud. Vous pouvez utiliser des filtres de nœuds pour réduire le temps de démarrage et l'utilisation du processeur de votre passerelle SiteWise Edge en incluant uniquement les chemins d'accès aux données que vous modélisez AWS IoT SiteWise. Par défaut, les passerelles SiteWise Edge téléchargent tous les chemins OPC UA à l'exception de ceux qui commencent par. `/Server/` Pour définir les filtres de nœuds OPC UA, vous pouvez utiliser les chemins des nœuds `*` et les `**` caractères génériques. Pour de plus amples informations, veuillez consulter [Utiliser les filtres de nœuds OPC UA dans Edge SiteWise](#).

Chaque structure de la liste doit contenir les informations suivantes :

action

Action pour cette règle de filtrage de nœud. Vous pouvez choisir les options suivantes :

- INCLUDE— La passerelle SiteWise Edge inclut uniquement les flux de données qui répondent à cette règle.

definition

Structure de règles de filtrage de nœud contenant les informations suivantes :

type

Type de chemin de filtre de nœud pour cette règle. Vous pouvez choisir les options suivantes :

- `OpcUaRootPath`— La passerelle SiteWise Edge évalue ce chemin de filtre de nœuds par rapport à la racine de la hiérarchie des chemins OPC UA.

rootPath

Le chemin du filtre de nœuds à évaluer par rapport à la racine de la hiérarchie des chemins OPC UA. Ce chemin doit commencer par /.

measurementDataStreamPrefix

Chaîne à ajouter à tous les flux de données provenant de la source. La passerelle SiteWise Edge ajoute ce préfixe à tous les flux de données provenant de cette source. Utilisez un préfixe de flux de données pour distinguer les flux de données portant le même nom mais provenant de sources différentes. Chaque flux de données doit avoir un nom unique dans votre compte.

typeConversions

Les types de conversions disponibles pour les types de données OPC UA non pris en charge. Chaque type de données est converti en chaînes. Pour de plus amples informations, veuillez consulter [Conversion de types de données non pris en charge](#).

array

Type de données de tableau simple converti en chaînes. Vous pouvez choisir les options suivantes :

- `JsonArray`— Indique que vous choisissez de convertir les types de données de vos tableaux simples en chaînes.

datetime

Type de DateTime données converti en chaînes. Vous pouvez choisir les options suivantes :

- `ISO8601String`— Indique que vous choisissez de convertir les types de données ISO 8601 en chaînes.

destination

Configuration pour la destination des balises OPC UA. Les passerelles V3 classiques avec flux, v2 et MQTT ont des configurations différentes pour les destinations.

type

Type de destination.

`streamName`— uniquement pour les streams classiques, les passerelles V2

Nom du flux . Le nom du flux doit être unique.

`streamBufferSize`— uniquement pour les streams classiques, les passerelles V2

Taille de la mémoire tampon du flux. Cela est important pour gérer le flux de données provenant des sources OPC UA.

propertyGroups

(Facultatif) La liste des groupes de propriétés qui définissent deadband et scanMode demandés par le protocole.

name

Nom du groupe de propriétés. Il doit s'agir d'un identifiant unique.

deadband

La deadband valeur définit la modification minimale de la valeur d'un point de données qui doit se produire avant que les données ne soient envoyées au cloud. Elle contient les informations suivantes :

type

Les types de deadband pris en charge. Vous pouvez choisir les options suivantes :

- **ABSOLUTE**— Une valeur fixe qui indique le changement absolu minimum requis pour considérer qu'un point de données est suffisamment significatif pour être envoyé vers le cloud.
- **PERCENT**— Valeur dynamique qui indique la modification minimale requise sous forme de pourcentage de la valeur du dernier point de données envoyé. Ce type de zone morte est utile lorsque les valeurs des données varient considérablement au fil du temps.

value

La valeur du deadband. Dans ce cas `typeABSOLUTE`, cette valeur est un double sans unité. Dans ce cas `typePERCENT`, cette valeur est un double compris entre 1 et 100.

eguMin

(Facultatif) L'unité d'ingénierie minimale lors de l'utilisation d'une `PERCENT` zone morte. Vous définissez ce paramètre si aucune unité d'ingénierie n'est configurée sur le serveur OPC UA.

eguMax

(Facultatif) Le maximum de l'unité d'ingénierie lors de l'utilisation d'une `PERCENT` zone morte. Vous définissez ce paramètre si aucune unité d'ingénierie n'est configurée sur le serveur OPC UA.

timeoutMilliseconds

Durée en millisecondes avant le délai d'expiration. Le minimum est 100.

scanMode

`scanModeStructure` qui contient les informations suivantes :

type

Les types de `scanMode`. Les valeurs acceptées sont `POLL` et `EXCEPTION`.

rate

Intervalle d'échantillonnage pour le mode de numérisation.

timestampToReturn

Source de l'horodatage. Vous pouvez choisir les options suivantes :

- `SOURCE_TIME`— Utilise l'horodatage de votre appareil.
- `SERVER_TIME`— Utilise l'horodatage de votre serveur.

Note

`TimestampToReturn` À utiliser avec la version 2.5.0 ou ultérieure du composant collecteur IoT SiteWise OPC UA. Si vous utilisez cette fonctionnalité avec des versions antérieures, les mises à jour de configuration échouent. Pour

de plus amples informations, veuillez consulter [Mettre à jour la version d'un AWS IoT SiteWise composant](#).

nodeFilterRuleDefinitions

(Facultatif) Liste des chemins de nœuds à inclure dans le groupe de propriétés. Les groupes de propriétés ne peuvent pas se chevaucher. Si vous ne spécifiez aucune valeur pour ce champ, le groupe contient tous les chemins situés sous la racine et vous ne pouvez pas créer de groupes de propriétés supplémentaires. La structure `nodeFilterRuleDefinitions` contient les informations suivantes :

type

`OpcUaRootPath` est le seul type pris en charge. Cela indique que la valeur de `rootPath` est un chemin relatif à la racine de l'espace de navigation OPC UA.

rootPath

Liste séparée par des virgules qui indique les chemins (relatifs à la racine) à inclure dans le groupe de propriétés.

Exemples de configuration de fonctionnalités supplémentaires pour les flux classiques, les passerelles V2 ()AWS CLI

L'exemple suivant définit une configuration de fonctionnalité de passerelle OPC UA SiteWise Edge à partir d'une charge utile stockée dans un fichier JSON.

```
aws iotsitewise update-gateway-capability-configuration \
--capability-namespace "iotsitewise:opcuacollector:2" \
--capability-configuration file://opc-ua-configuration.json
```

Exemple : configuration de la source OPC UA

Le `opc-ua-configuration.json` fichier suivant définit une configuration de source OPC UA de base non sécurisée.

```
{
  "sources": [
    {
      "name": "Wind Farm #1",
      "endpoint": {
```

```

        "certificateTrust": {
            "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.0:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
            "type": "Anonymous"
        },
        "nodeFilterRules": []
    },
    "measurementDataStreamPrefix": ""
}
]
}

```

Exemple : configuration de source OPC UA avec groupes de propriétés définis

Le `opc-ua-configuration.json` fichier suivant définit une configuration de source OPC UA de base non sécurisée avec des groupes de propriétés définis.

```

{
  "sources": [
    {
      "name": "source1",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://10.0.0.9:49320",
        "securityPolicy": "NONE",
        "messageSecurityMode": "NONE",
        "identityProvider": {
          "type": "Anonymous"
        },
        "nodeFilterRules": [
          {
            "action": "INCLUDE",
            "definition": {
              "type": "OpcUaRootPath",
              "rootPath": "/Utilities/Tank"
            }
          }
        ]
      }
    }
  ]
}

```

```

    ]
  },
  "measurementDataStreamPrefix": "propertyGroups",
  "propertyGroups": [
    {
      "name": "Deadband_Abs_5",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Temperature/TT-001"
        },
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Temperature/TT-002"
        }
      ],
      "deadband": {
        "type": "ABSOLUTE",
        "value": 5.0,
        "timeoutMilliseconds": 120000
      }
    },
    {
      "name": "Polling_10s",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Pressure/PT-001"
        }
      ],
      "scanMode": {
        "type": "POLL",
        "rate": 10000
      }
    },
    {
      "name": "Percent_Deadband_Timeout_90s",
      "nodeFilterRuleDefinitions": [
        {
          "type": "OpcUaRootPath",
          "rootPath": "/Utilities/Tank/Flow/FT-*"
        }
      ],
      "deadband": {

```



```

        "type": "PERCENT",
        "value": 5.0,
        "eguMin": -100,
        "eguMax": 100,
        "timeoutMilliseconds": 90000
      }
    ]
  }
}

```

Exemple : configuration de la source OPC UA avec propriétés

L'exemple JSON suivant `opc-ua-configuration.json` définit une configuration de source OPC UA avec les propriétés suivantes :

- Faire confiance à n'importe quel certificat.
- Utilise la politique de BASIC256 sécurité pour sécuriser les messages.
- Utilise le mode SIGN_AND_ENCRYPT pour sécuriser les connexions.
- Utilise les informations d'authentification stockées dans un secret de Secrets Manager.
- Filtre les flux de données sauf ceux dont le chemin commence par `/WindFarm/2/WindTurbine/`.
- Ajoute `/Washington` au début de chaque chemin de flux de données pour distinguer ce « parc éolien #2 » et un « parc éolien #2 » dans une autre zone.

```

{
  "sources": [
    {
      "name": "Wind Farm #2",
      "endpoint": {
        "certificateTrust": {
          "type": "TrustAny"
        },
        "endpointUri": "opc.tcp://203.0.113.1:49320",
        "securityPolicy": "BASIC256",
        "messageSecurityMode": "SIGN_AND_ENCRYPT",
        "identityProvider": {
          "type": "Username",

```

```

        "usernameSecretArn":
        "arn:aws:secretsmanager:region:123456789012:secret:greenrass-windfarm2-auth-1ABCDE"
    },
    "nodeFilterRules": [
    {
        "action": "INCLUDE",
        "definition": {
            "type": "OpcUaRootPath",
            "rootPath": "/WindFarm/2/WindTurbine/"
        }
    }
    ],
    "measurementDataStreamPrefix": "/Washington"
}
]
}

```

Exemple : Configuration de la source OPC UA avec certificat de confiance

L'exemple JSON suivant `opc-ua-configuration.json` définit une configuration de source OPC UA avec les propriétés suivantes :

- Approuve un certificat X.509 donné.
- Utilise la politique de BASIC256 sécurité pour sécuriser les messages.
- Utilise le mode SIGN_AND_ENCRYPT pour sécuriser les connexions.

```

{
  "sources": [
    {
      "name": "Wind Farm #3",
      "endpoint": {
        "certificateTrust": {
          "type": "X509",
          "certificateBody": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAwTC0lBTsBDb25zb2x1MRIw
EAYDVQQDEw1UZXR0Q21sYWVxHmAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpva25l
jb20wHhcNMTEwNDI1MjA0NTIwHhcnMTIwNDI1MjA0NTIwHjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAgTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb

```

```

WF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----",
    "certificateChain": "-----BEGIN CERTIFICATE-----
MIICiTCCAfICCCQD6m7oRw0uX0jANBgkqhkiG9w
0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZ
WF0dGx1MQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIw
EAYDVQQDEw1UZXRhbnQ21sYWMxHzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5
jb20wHhcNMTEwNDI1MjA0NTIxWhcNMTEwNDI1MjA0NTIxWjCBiDELMAkGA1UEBh
MCVVMxCzAJBgNVBAGTAldBMRAwDgYDVQQHEwdTZWF0dGx1MQ8wDQYDVQQKEwZBb
WF6b24xFDASBgNVBAsTC0lBTSBDb25zb2x1MRIwEAYDVQQDEw1UZXRhbnQ21sYWMx
HzAdBgkqhkiG9w0BCQEWEG5vb25lQGFTYXpvi5jb20wgZ8wDQYJKoZIhvcNAQE
BBQADgY0AMIGJAoGBAMaK0dn+a4GmWIWJ21uUSfwfEvySWtC2XADZ4nB+BLyGVI
k60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9TrDHudUZg3qX4waLG5M43q7Wgc/MbQ
ITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAaRHhd1QWIMm2nr
AgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9q6q+auN
KyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0FkbFFBjvSfpJI1J00zbhNYS5f6Guo
EDmFJl0ZxBHjJnyp3780D8uTs7fLvJx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw
3rrszlaEXAMPLE=
    -----END CERTIFICATE-----"
    },
    "endpointUri": "opc.tcp://203.0.113.2:49320",
    "securityPolicy": "BASIC256",
    "messageSecurityMode": "SIGN_AND_ENCRYPT",
    "identityProvider": {
        "type": "Anonymous"
    },
    "nodeFilterRules": []
  },
  "measurementDataStreamPrefix": ""
}
]
}

```

Configurer les serveurs OPC UA pour qu'ils fassent confiance à la passerelle AWS IoT SiteWise Edge

Si vous choisissez une valeur `messageSecurityMode` autre que `None` lors de la configuration de votre source OPC UA, vous devez permettre à vos serveurs source de faire confiance à la passerelle AWS IoT SiteWise Edge. La passerelle SiteWise Edge génère un certificat dont votre serveur source peut avoir besoin. Le processus varie en fonction de vos serveurs sources. Pour plus d'informations, consultez la documentation de vos serveurs.

La procédure suivante décrit les étapes de base.

Pour permettre à un serveur OPC UA de faire confiance à la passerelle SiteWise Edge

1. Ouvrez l'interface de configuration de votre serveur OPC UA.
2. Entrez le nom d'utilisateur et le mot de passe de l'administrateur du serveur OPC UA.
3. Localisez Clients de confiance dans l'interface, puis choisissez Client de passerelle AWS IoT SiteWise .
4. Choisissez Trust (Approuver).

Exportation du certificat client OPC UA

Certains serveurs OPC UA ont besoin d'accéder au fichier de certificat client OPC UA pour faire confiance à la passerelle SiteWise Edge. Si cela s'applique à vos serveurs OPC UA, vous pouvez utiliser la procédure suivante pour exporter le certificat client OPC UA depuis la passerelle SiteWise Edge. Vous pouvez ensuite importer le certificat sur votre serveur OPC UA.

Pour exporter le fichier de certificat client OPC UA pour une source

1. Exécutez la commande suivante pour passer au répertoire contenant le fichier de certificat.
`sitewise-work` Remplacez-le par le chemin de stockage local du dossier de travail `aws.iot.SiteWiseEdgeCollectorOpcua` Greengrass et remplacez-le `source-name` par le nom de la source de données.

Par défaut, le dossier de travail Greengrass se trouve sous Linux et `/greengrass/v2/work/aws.iot.SiteWiseEdgeCollectorOpcua` C:/greengrass/v2/work/`aws.iot.SiteWiseEdgeCollectorOpcua` activé. Microsoft Windows

```
cd /sitewise-work/source-name/opcua-certificate-store
```

2. Le certificat client OPC UA de la passerelle SiteWise Edge pour cette source se trouve dans le `aws-iot-opcua-client.pfx` fichier.

Exécutez la commande suivante pour exporter le certificat vers un fichier `.pem` appelé `aws-iot-opcua-client-certificate.pem`.

```
keytool -exportcert -v -alias aws-iot-opcua-client -keystore aws-iot-opcua-client.pfx -storepass amazon -storetype PKCS12 -rfc > aws-iot-opcua-client-certificate.pem
```

3. Transférez le fichier de certificat de la passerelle SiteWise Edge vers le serveur OPC UA. `aws-iot-opcua-client-certificate.pem`

Pour ce faire, vous pouvez utiliser un logiciel commun tel que le programme `scp` pour transférer le fichier en utilisant le protocole SSH. Pour de plus amples informations, veuillez consulter [Copie sécurisée](#) sur Wikipédia.

Note

Si votre passerelle SiteWise Edge s'exécute sur Amazon Elastic Compute Cloud (Amazon EC2) et que vous vous y connectez pour la première fois, vous devez configurer les conditions requises pour vous connecter. Pour plus d'informations, consultez [Connect to your Linux instance using SSH](#) dans le Amazon EC2 User Guide.

4. Importez le fichier de certificat sur le serveur OPC UA pour faire confiance à la passerelle SiteWise Edge. `aws-iot-opcua-client-certificate.pem` Les étapes varient en fonction du serveur source que vous utilisez. Consultez la documentation de votre serveur.

Filtrez les plages d'ingestion de données avec OPC UA

Vous pouvez contrôler la façon dont vous ingérez les données avec une source OPC UA en utilisant le mode scan et les plages de zones mortes. Ces fonctionnalités vous permettent de contrôler le type de données à ingérer, ainsi que la manière et le moment où votre serveur et la passerelle SiteWise Edge échangent ces informations.

Collectez ou filtrez les données en fonction de leur qualité

Vous pouvez configurer vos paramètres de qualité des données pour contrôler les données collectées à partir de la source OPC UA. La source de données inclut l'évaluation de qualité sous

forme de métadonnées lorsqu'elle l'envoie. Vous pouvez sélectionner l'une ou l'ensemble des options suivantes :

- Good
- Bad
- Uncertain

Gérer les valeurs NaN ou nulles

SiteWise Edge prend en charge la collecte et le traitement des valeurs NaN et nulles.

- NaN (pas un nombre) : représente des résultats numériques non définis ou non représentables.
- Null : indique des données manquantes.

Le collecteur IoT SiteWise OPC UA capture les valeurs NaN et Null avec une qualité MAUVAISE ou INCERTAINE. Ces valeurs spéciales sont écrites dans le flux local, ce qui permet une collecte de données plus complète.

Contrôlez la fréquence de collecte des données avec le mode Scan

Vous pouvez configurer le mode de numérisation de votre OPC UA pour contrôler la manière dont vous collectez les données à partir de votre source OPC UA. Vous pouvez choisir le mode abonnement ou le mode sondage.

- Mode d'abonnement — La source OPC UA collecte des données à envoyer à votre passerelle SiteWise Edge à la fréquence définie par votre taux de numérisation. Le serveur envoie des données uniquement lorsque la valeur a changé. Il s'agit donc de la fréquence maximale à laquelle votre passerelle SiteWise Edge reçoit des données.
- Mode d'interrogation : votre passerelle SiteWise Edge interroge la source OPC UA à une fréquence définie en fonction de votre fréquence de numérisation. Le serveur envoie des données, que la valeur ait changé ou non, de sorte que votre passerelle SiteWise Edge reçoit toujours les données à cet intervalle.

Note

L'option du mode de sondage remplace vos paramètres de zone morte pour cette source.

Filtrez l'ingestion de données OPC UA avec des plages mortes

Vous pouvez appliquer une zone morte à vos groupes de propriétés sources OPC UA afin de filtrer et de supprimer certaines données au lieu de les envoyer vers le Cloud. AWS Une zone morte indique une fenêtre de fluctuations attendues des valeurs de données entrantes provenant de votre source OPC UA. Si les valeurs se trouvent dans cette fenêtre, votre serveur OPC UA ne les enverra pas au AWS Cloud. Vous pouvez utiliser le filtrage en zone morte pour réduire la quantité de données que vous traitez et envoyez vers le AWS cloud. Pour savoir comment configurer des sources OPC UA pour votre passerelle SiteWise Edge, consultez [Sources de données OPC UA pour les passerelles AWS IoT SiteWise Edge](#).

Note

Votre serveur supprime toutes les données qui se trouvent dans la fenêtre spécifiée par votre deadband. Vous ne pouvez pas récupérer ces données supprimées.

Types de deadbands

Vous pouvez spécifier deux types de zones mortes pour le groupe de propriétés de votre serveur OPC UA. Ils vous permettent de choisir la quantité de données à envoyer AWS vers le cloud et celle à supprimer.

- **Pourcentage** : vous spécifiez une fenêtre en utilisant un pourcentage de fluctuation attendue de la valeur de mesure. Le serveur calcule la fenêtre exacte à partir de ce pourcentage et envoie au AWS Cloud les données qui dépassent les limites de la fenêtre. Par exemple, la spécification d'une valeur de zone morte de 2 % sur un capteur dans une plage comprise entre -100 degrés Fahrenheit et +100 degrés Fahrenheit indique au serveur d'envoyer des données vers le AWS cloud lorsque la valeur change de 4 degrés Fahrenheit ou plus.

Note

Vous pouvez éventuellement spécifier une valeur de bande morte minimale et maximale pour cette fenêtre si votre serveur source ne définit pas d'unités d'ingénierie. Si aucune plage d'unités d'ingénierie n'est fournie, le serveur OPC UA utilise par défaut la plage complète du type de données de mesure.

- **Absolu** — Vous spécifiez une fenêtre en utilisant des unités exactes. Par exemple, la spécification d'une valeur de zone morte de 2 sur un capteur indique au serveur d'envoyer des données au

AWS Cloud lorsque sa valeur change d'au moins 2 unités. Vous pouvez utiliser l'impasse absolue pour les environnements dynamiques dans lesquels des fluctuations sont régulièrement attendues pendant les opérations normales.

Délais d'expiration de la bande morte

Vous pouvez éventuellement configurer un paramètre de délai d'expiration en zone morte. Après ce délai, le serveur OPC UA envoie la valeur de mesure actuelle même si elle se situe dans les limites de la fluctuation de bande morte attendue. Vous pouvez utiliser le paramètre de temporisation pour garantir AWS IoT SiteWise l'ingestion d'un flux constant de données à tout moment, même lorsque les valeurs ne dépassent pas la fenêtre de zone morte définie.

Utiliser les filtres de nœuds OPC UA dans Edge SiteWise

Lorsque vous définissez des sources de données OPC UA pour une passerelle SiteWise Edge, vous pouvez définir des filtres de nœuds. Les filtres de nœuds vous permettent de limiter les chemins de flux de données que la passerelle SiteWise Edge envoie vers le cloud. Vous pouvez utiliser des filtres de nœuds pour réduire le temps de démarrage et l'utilisation du processeur de votre passerelle SiteWise Edge en incluant uniquement les chemins d'accès aux données que vous modélisez AWS IoT SiteWise. Par défaut, les passerelles SiteWise Edge téléchargent tous les chemins OPC UA à l'exception de ceux qui commencent par `/Server/`. Vous pouvez utiliser les caractères génériques `*` et `**` dans vos filtres de nœuds pour inclure plusieurs chemins de flux de données avec un seul filtre. Pour savoir comment configurer des sources OPC UA pour votre passerelle SiteWise Edge, consultez [Sources de données OPC UA pour les passerelles AWS IoT SiteWise Edge](#).

Note

AWS IoT SiteWise redémarre votre passerelle SiteWise Edge chaque fois que vous ajoutez ou modifiez une source. Votre passerelle SiteWise Edge n'ingère pas de données lors de la mise à jour de la configuration de la source. Le délai de redémarrage de votre passerelle SiteWise Edge dépend du nombre de balises figurant sur les sources de votre passerelle SiteWise Edge. Le temps de redémarrage peut aller de quelques secondes (pour une passerelle SiteWise Edge avec peu de balises) à plusieurs minutes (pour une passerelle SiteWise Edge avec de nombreuses balises).

Le tableau suivant répertorie les caractères génériques que vous pouvez utiliser pour filtrer les sources de données OPC UA.

Caractères génériques du filtre de nœuds OPC UA

Caractère générique	Description
*	Correspond à un niveau unique dans un chemin de flux de données.
**	Correspond à plusieurs niveaux dans un chemin de flux de données.

Note

Si vous configurez une source avec un filtre large, puis que vous modifiez ultérieurement la source pour utiliser un filtre plus restrictif, AWS IoT SiteWise arrête de stocker les données qui ne correspondent pas au nouveau filtre.

Exemple : scénario utilisant des filtres de nœuds

Considérons les flux de données hypothétiques suivants :

- /WA/Factory 1/Line 1/PLC1
- /WA/Factory 1/Line 1/PLC2
- /WA/Factory 1/Line 2/Counter1
- /WA/Factory 1/Line 2/PLC1
- /OR/Factory 1/Line 1/PLC1
- /OR/Factory 1/Line 2/Counter2

À l'aide des flux de données précédents, vous pouvez définir des filtres de nœuds pour limiter les données à inclure à partir de votre source OPC UA.

- Pour sélectionner tous les nœuds dans cet exemple, utilisez / ou /**/. Vous pouvez inclure plusieurs répertoires ou dossiers avec les caractères génériques « ** ».
- Pour sélectionner tous les flux de données PLC, utilisez /*/*/*/PLC* ou /**/PLC*.
- Pour sélectionner tous les compteurs dans cet exemple, utilisez /**/Counter* ou /*/*/*/*/Counter*.

- Pour sélectionner tous les compteurs à partir de Line 2, utilisez `/**/Line 2/Counter*`.

Conversion de types de données non pris en charge

Activez éventuellement la conversion des types de données AWS IoT SiteWise pour les tableaux et types de DateTime données simples. AWS IoT SiteWise ne prend pas en charge tous les types de données OPC UA. Lorsque vous envoyez des données non prises en charge à votre flux de AWS IoT Greengrass données, ces données sont perdues. Toutefois, en convertissant les types de données natifs non pris en charge en chaînes, vous pouvez ingérer les données au AWS IoT SiteWise lieu de les supprimer. AWS IoT SiteWise sérialise vos données converties afin que vous puissiez ultérieurement utiliser vos propres fonctions pour reconverter les chaînes dans leur type de données d'origine en aval, si nécessaire.

Vous pouvez mettre à jour les paramètres de conversion des types de données pour une source de données à tout moment et chaque source de données peut avoir ses propres paramètres.

Lorsque vous ajoutez des sources de données dans le Console AWS IoT SiteWise, deux cases à cocher se trouvent sous Conversion de type de données dans Configuration avancée. Vous pouvez indiquer les types de données à convertir en chaînes.

De plus, le collecteur IoT SiteWise OPC UA peut accepter des valeurs NaN ou nulles sur le bord.

- Convertissez des valeurs de tableau contenant des types de données simples en chaînes JSON
- Convertir DateTime des valeurs en chaînes ISO 8601

Prérequis

- Utilisez la version 2.5.0 ou ultérieure du collecteur [IoT SiteWise OPC UA](#).

Limites

Voici les limites de la conversion des types de données OPC UA en AWS IoT SiteWise chaînes.

- La conversion de types de données complexes n'est pas prise en charge.
- Les limites de chaînes après la conversion sont de 1 024 octets. Si la chaîne est supérieure à 1024 octets, elle est rejetée par AWS IoT SiteWise.

Configurer l'authentification des sources de données pour SiteWise Edge

Si votre serveur OPC UA nécessite des informations d'authentification pour se connecter, vous pouvez les utiliser AWS Secrets Manager pour créer et déployer un secret sur votre passerelle SiteWise Edge. AWS Secrets Manager chiffre les secrets sur l'appareil pour protéger votre nom d'utilisateur et votre mot de passe jusqu'à ce que vous ayez besoin de les utiliser. Pour plus d'informations sur le composant du gestionnaire de AWS IoT Greengrass secrets, consultez la section [Gestionnaire de secrets](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

Pour plus d'informations sur la gestion de l'accès aux secrets de Secrets Manager, voir :

- [Qui est autorisé à accéder à vos AWS Secrets Manager secrets.](#)
- [Déterminer si une demande est autorisée ou refusée dans un compte.](#)

Étape 1 : créer des secrets d'authentification à la source

Vous pouvez l'utiliser AWS Secrets Manager pour créer un secret d'authentification pour votre source de données. Dans le secret, définissez **username** des paires **password** clé-valeur contenant les détails d'authentification de votre source de données.

Pour créer des secrets (console)

1. Accédez à la [console AWS Secrets Manager](#).
2. Choisissez Store a new secret (Stocker un nouveau secret).
3. Sous Type de secret, sélectionnez Autre type de secret.
4. Sous Paires clé/valeur, procédez comme suit :
 1. Dans la première zone de saisie, entrez **username** le nom d'utilisateur dans la deuxième zone de saisie.
 2. Choisissez Add row (Ajouter une ligne).
 3. Dans la première zone de saisie, entrez **password** le mot de passe dans la deuxième zone de saisie.
5. Pour Clé de chiffrement, sélectionnez aws/secretsmanager, puis Next.
6. Sur la page Enregistrer un nouveau secret, entrez un nom secret.
7. (Facultatif) Entrez une description qui vous aidera à identifier ce secret, puis choisissez Next.

8. (Facultatif) Sur la page Enregistrer une nouvelle page secrète, activez la rotation automatique. Pour plus d'informations, voir [Rotation des secrets](#) dans le guide de AWS Secrets Manager l'utilisateur.
9. Spécifiez un calendrier de rotation.
10. Choisissez une fonction Lambda capable de faire pivoter ce secret, puis choisissez Next.
11. Passez en revue vos configurations secrètes, puis choisissez Store.

Pour autoriser l'interaction avec votre passerelle SiteWise Edge AWS Secrets Manager, le rôle IAM de votre passerelle SiteWise Edge doit autoriser l'`secretsmanager:GetSecretValue` action. Vous pouvez utiliser le périphérique principal de Greengrass pour rechercher la politique IAM. Pour plus d'informations sur la mise à jour d'une stratégie IAM, consultez la section [Modification des politiques IAM](#) dans le Guide de l'AWS Identity and Access Management utilisateur.

Exemple politique

secret-arn Remplacez-le par le Amazon Resource Name (ARN) du secret que vous avez créé à l'étape précédente. Pour plus d'informations sur la façon d'obtenir l'ARN d'un secret, consultez la section [Rechercher des secrets AWS Secrets Manager dans](#) le guide de AWS Secrets Manager l'utilisateur.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": [
        "secret-arn"
      ]
    }
  ]
}
```

Étape 2 : Déployer des secrets sur votre périphérique de passerelle SiteWise Edge

Vous pouvez utiliser la AWS IoT SiteWise console pour déployer des secrets sur votre passerelle SiteWise Edge.

Pour déployer un secret (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Passerelles.
3. Dans la liste des passerelles, choisissez la passerelle SiteWise Edge cible.
4. Dans la section Configuration de la passerelle, choisissez le lien du périphérique principal Greengrass pour ouvrir le AWS IoT Greengrass cœur associé à la passerelle SiteWise Edge.
5. Dans le volet de navigation, choisissez Deployments.
6. Choisissez le déploiement cible, puis sélectionnez Revise.
7. Sur la page Spécifier la cible, choisissez Next.
8. Sur la page Sélectionner les composants, dans la section Composants publics, désactivez Afficher uniquement les composants sélectionnés.
9. Recherchez et choisissez le fichier aws.greengrass.SecretManagercomposant, puis choisissez Next.
10. Dans la liste des composants sélectionnés, sélectionnez aws.greengrass.SecretManager, puis choisissez Configurer le composant.
11. Dans le champ Configuration à fusionner, ajoutez l'objet JSON suivant.

Note

secret-arn Remplacez-le par l'ARN du secret que vous avez créé à l'étape précédente. Pour plus d'informations sur la façon d'obtenir l'ARN d'un secret, consultez la section [Rechercher des secrets AWS Secrets Manager dans](#) le guide de AWS Secrets Manager l'utilisateur.

```
{
  "cloudSecrets": [
    {
      "arn": "secret-arn"
    }
  ]
}
```

12. Choisissez Confirmer.
13. Choisissez Suivant.

14. Sur la page Configurer les paramètres avancés, choisissez Next.
15. Passez en revue vos configurations de déploiement, puis choisissez Deploy.

Étape 3 : ajouter des configurations d'authentification

Vous pouvez utiliser la AWS IoT SiteWise console pour ajouter des configurations d'authentification à votre passerelle SiteWise Edge.

Pour ajouter des configurations d'authentification (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans la liste des passerelles, choisissez la passerelle SiteWise Edge cible.
3. Dans la liste des sources de données, choisissez la source de données cible, puis sélectionnez Modifier.
4. Sur la page Ajouter une source de données, choisissez Configuration avancée.
5. Pour la configuration de l'authentification, choisissez le secret que vous avez déployé à l'étape précédente.
6. Choisissez Save (Enregistrer).

Sources de données partenaires sur les passerelles SiteWise Edge

Lorsque vous utilisez une passerelle AWS IoT SiteWise Edge, vous pouvez connecter une source de données partenaire à votre passerelle SiteWise Edge et recevoir des données du partenaire sur votre passerelle SiteWise Edge et dans le AWS cloud. Ces sources de données partenaires sont AWS IoT Greengrass des composants développés en partenariat entre AWS et le partenaire. Lorsque vous ajoutez une source de données partenaire, ce composant AWS IoT SiteWise sera créé et déployé sur votre passerelle SiteWise Edge.

Note

Vous pouvez ajouter une source de données pour chaque partenaire dans chaque passerelle.

Pour ajouter une source de données partenaire, procédez comme suit :

1. [Ajouter une source de données partenaire dans SiteWise Edge](#)

2. Accédez au portail Web du partenaire, le cas échéant, et configurez la source de données du partenaire afin qu'elle se connecte à la passerelle SiteWise Edge.

Rubriques

- [Sécurité](#)
- [Configuration Docker sur votre passerelle SiteWise Edge](#)
- [Ajouter une source de données partenaire dans SiteWise Edge](#)
- [SiteWise Options de source de données pour les partenaires Edge Gateway](#)

Sécurité

Dans le cadre du [modèle de responsabilité partagée](#) entre AWS nos clients et nos partenaires, les informations suivantes décrivent qui est responsable des différents aspects de la sécurité :

Responsabilité du client

- Sélection du partenaire.
- Configuration de l'accès réseau accordé au partenaire.
- Surveillance de l'utilisation raisonnable des ressources de la machine de passerelle SiteWise Edge (processeur, mémoire et système de fichiers).

AWS responsabilité

- Isoler le partenaire des ressources AWS cloud du client, à l'exception de celles dont le partenaire a besoin. Dans ce cas, AWS IoT SiteWise ingestion.
- Restreindre la solution partenaire à une utilisation raisonnable des ressources de la machine de passerelle SiteWise Edge (processeur et mémoire).

Responsabilité du partenaire

- Utilisation de valeurs par défaut sécurisées.
- Garantir la sécurité de la solution au fil du temps grâce à des correctifs et à d'autres mises à jour appropriées.
- Préserver la confidentialité des données des clients.

Configuration Docker sur votre passerelle SiteWise Edge

AWS IoT SiteWise fournit une Docker image qui vous permet d'exécuter l'application SiteWise Edge sur différents environnements et plateformes. Cette Docker image encapsule tous les composants

et dépendances nécessaires pour collecter, traiter et envoyer les données de votre équipement industriel vers le AWS cloud. En utilisant l'image Docker, vous pouvez déployer et exécuter l'application SiteWise Edge sur des hôtes compatibles Docker, tels que des serveurs, des appareils Edge ou des services de conteneur basés sur le cloud.

Pour ajouter une source de données partenaire, [Docker Engine](#) 1.9.1 ou version ultérieure doit être installé sur votre appareil local.

Note

La version 20.10 est la dernière version vérifiée pour fonctionner avec le logiciel de passerelle SiteWise Edge.

Vérifier Docker qu'il est installé

Pour vérifier Docker qu'il est installé, exécutez la commande suivante depuis un terminal connecté à votre passerelle SiteWise Edge :

```
docker info
```

Si la commande renvoie un `docker is not recognized` résultat ou si une ancienne version de Docker est installée, [installez Docker Engine](#) avant de continuer.

Configurer Docker

L'utilisateur du système qui exécute un composant de Docker conteneur doit disposer des autorisations root ou administrateur, ou vous devez le configurer Docker pour l'exécuter en tant qu'utilisateur non root ou non administrateur.

Sur les appareils Linux, vous devez ajouter un `ggc_user` utilisateur au `docker` groupe sans lequel vous pouvez appeler Docker des commandes `sudo`.

Pour ajouter `ggc_user` au `docker` groupe l'utilisateur non root que vous utilisez pour exécuter les composants du Docker conteneur, ou l'utilisateur non root que vous utilisez pour exécuter les composants du conteneur, exécutez la commande suivante :

```
sudo usermod -aG docker ggc_user
```

Pour plus d'informations, consultez les [étapes de post-installation de Linux pour Docker Engine](#).

Ajouter une source de données partenaire dans SiteWise Edge

Pour connecter une source de données partenaire à votre passerelle SiteWise Edge, ajoutez-la en tant que source de données. Lorsque vous l'ajoutez en tant que source de données, un AWS IoT Greengrass composant privé AWS IoT SiteWise sera déployé sur votre passerelle SiteWise Edge.

Prérequis

Pour ajouter une source de données partenaire, vous devez effectuer les opérations suivantes :

- Pour EasyEdge et CloudRail, créez un compte auprès du partenaire, puis liez les comptes.
- [Configuration Docker sur votre passerelle SiteWise Edge](#)

Création d'une passerelle SiteWise Edge avec une source de données partenaire

Si vous souhaitez créer une nouvelle passerelle SiteWise Edge, suivez les étapes décrites dans [Créez une passerelle SiteWise Edge auto-hébergée](#). Après avoir créé la passerelle SiteWise Edge, suivez les étapes ci-dessous [Ajouter une source de données partenaire à une passerelle SiteWise Edge existante](#) pour ajouter une source de données partenaire.

Ajouter une source de données partenaire à une passerelle SiteWise Edge existante

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Choisissez la passerelle SiteWise Edge à laquelle vous souhaitez connecter la source de données partenaire.
4. Sous Sources de données, choisissez Ajouter une source de données.
5. Sur l'écran Ajouter une source de données, choisissez un type de source pour sélectionner le partenaire qui connecte votre passerelle SiteWise Edge. Chaque source de données possède ses propres options de configuration. Il existe deux catégories de sources de données : les AWS sources et les sources partenaires.

À l'aide d'une source de données partenaire, vous pouvez sélectionner une source par passerelle. Pour obtenir la liste des options d'intégration des partenaires de sources de données, consultez [SiteWise Options de source de données pour les partenaires Edge Gateway](#). Notez que vous pouvez ajouter jusqu'à 100 sources de données OPC UA (AWS sources). Pour commencer à utiliser les sources de données OPC UA, voir [Sources de données OPC UA pour les passerelles AWS IoT SiteWise Edge](#).

6. Entrez le nom de la source.
7. Sélectionnez l'onglet de votre source de données ci-dessous et suivez la procédure de configuration.

CloudRail

Une grande partie de la CloudRail configuration est effectuée dans le CloudRail portail après avoir enregistré la source de données pour votre passerelle SiteWise Edge. Il est toutefois nécessaire d'autoriser la connexion.

Note

La CloudRail connexion n'est disponible que sous Linux.

1. [Créez un CloudRail compte](#) pour commencer à vous connecter à AWS IoT SiteWise.
2. Assurez-vous que Docker est installé sur votre passerelle. Pour de plus amples informations, veuillez consulter [Configuration Docker sur votre passerelle SiteWise Edge](#).
3. Lisez le contrat d'autorisation d'accès et de déploiement, puis choisissez Autoriser. Le fait de cocher cette case permet au AWS partenaire d'accéder à votre source de données et AWS de le déployer sur le composant du partenaire.

Note

Le préfixe de mesure (facultatif) est défini dans votre CloudRail portail.

Note

Le logiciel du partenaire est développé, maintenu et pris en charge par le AWS partenaire. AWS n'est pas responsable de l'interface, de la configuration ou du logiciel.

Pour de plus amples informations, veuillez consulter [CloudRail](#).

EasyEdge

Une grande partie de la EasyEdge configuration est effectuée dans le EasyEdge portail après avoir enregistré la source de données pour votre passerelle SiteWise Edge. Il est toutefois nécessaire d'autoriser la connexion.

Note

La EasyEdge connexion n'est disponible que sous Linux.

1. [Créez un EasyEdge compte](#) pour commencer à vous connecter à AWS IoT SiteWise.
2. Assurez-vous que Docker est installé sur votre passerelle. Pour de plus amples informations, veuillez consulter [Configuration Docker sur votre passerelle SiteWise Edge](#).
3. Lisez le contrat d'autorisation d'accès et de déploiement, puis choisissez Autoriser. Le fait de cocher cette case permet au AWS partenaire d'accéder à votre source de données et AWS de le déployer sur le composant du partenaire.

Note

Le préfixe de mesure (facultatif) est défini dans votre EasyEdge portail.

Note

Le logiciel du partenaire est développé, maintenu et pris en charge par le AWS partenaire. AWS n'est pas responsable de l'interface, de la configuration ou du logiciel.

Pour de plus amples informations, veuillez consulter [EasyEdge](#).

Litmus Edge

Vous pouvez activer la Litmus configuration de deux manières. Activez Litmus Edge directement AWS IoT SiteWise en utilisant les informations du Litmus Edge Manager

portail. Vous pouvez également activer manuellement Litmus Edge pour AWS IoT SiteWise via Litmus Edge Manager.

 Note

La Litmus Edge connexion n'est disponible que sous Linux.

Pour l'activer à l'aide d'un code Litmus Edge d'activation sur AWS IoT SiteWise

Utilisez cette procédure lorsque vous ajoutez une source de Litmus Edge données avec un code Litmus Edge d'activation sur le Console AWS IoT SiteWise.

1. Sélectionnez Activer maintenant à l'aide d'un code. Des options de configuration supplémentaires apparaissent.
2. Entrez le Litmus Edge Manager pour vous connecter Litmus Edge à votre passerelle SiteWise Edge. Pour plus d'informations, voir [Étape 3a : Définir le point de terminaison de gestion des données et des appareils](#) dans la Litmus Edge Manager documentation.
3. Entrez le code Litmus Edge Manager d'activation pour l'activer Litmus Edge sur AWS IoT SiteWise
4. Fournissez éventuellement AWS IoT SiteWise le certificat Litmus Edge Manager CA. Le certificat Litmus Edge empêche toute activation sur un site non autorisé Litmus Edge Manager.
5. Assurez-vous que Docker est installé sur votre passerelle. Pour de plus amples informations, veuillez consulter [Configuration Docker sur votre passerelle SiteWise Edge](#).

 Note

AWS IoT SiteWise déploie l'application partenaire en tant que conteneur Docker. L'application est déployée avec des NET_ADMIN fonctionnalités permettant de gérer le conteneur Litmus Edge Docker. Litmus Edge Manager Litmus Edgenécessite cet accès privilégié pour fonctionner sur vos appareils. Pour plus d'informations sur les exigences de Litmus Edge Docker, consultez la section [Installation de Docker](#) dans le QuickStart guide de la documentation Litmus Edge.

6. Lisez le contrat d'autorisation d'accès et de déploiement, puis choisissez Autoriser. Le fait de cocher cette case permet au AWS partenaire d'accéder à votre source de données et AWS de le déployer sur le composant du partenaire.

Pour activer manuellement via Litmus Edge

1. Sélectionnez Activer ultérieurement Litmus Edge.
2. Assurez-vous que Docker est installé sur votre passerelle. Pour de plus amples informations, veuillez consulter [Configuration Docker sur votre passerelle SiteWise Edge](#).

 Note

AWS IoT SiteWise déploie l'application partenaire en tant que conteneur Docker. L'application est déployée avec des NET_ADMIN fonctionnalités permettant de gérer le conteneur Litmus Edge Docker. Litmus Edge Manager Litmus Edgenécessite cet accès privilégié pour fonctionner sur vos appareils. Pour plus d'informations sur les exigences de Litmus Edge Docker, consultez la section [Installation de Docker](#) dans le QuickStart guide de la documentation Litmus Edge.

3. Lisez le contrat d'autorisation d'accès et de déploiement, puis choisissez Autoriser. Le fait de cocher cette case permet au AWS partenaire d'accéder à votre source de données et AWS de le déployer sur le composant du partenaire.
4. Une fois le déploiement terminé, suivez les instructions [d'accès à l'interface utilisateur Web de Litmus Edge](#) figurant dans la Litmus Edge QuickStart Guidedocumentation.

 Note

Le logiciel du partenaire est développé, maintenu et pris en charge par le AWS partenaire. AWS n'est pas responsable de l'interface, de la configuration ou du logiciel.

Pour de plus amples informations, veuillez consulter [Litmus Edge](#).

8. Choisissez Enregistrer.

SiteWise Options de source de données pour les partenaires Edge Gateway

AWS IoT SiteWise vous permet de connecter et d'ingérer des données provenant de diverses sources de données partenaires, telles que des équipements industriels, des capteurs et d'autres systèmes tiers. Pour connecter une source de données partenaire, vous devez suivre quelques étapes, notamment configurer la source de données à laquelle envoyer les données AWS IoT SiteWise, configurer les autorisations et l'authentification nécessaires, et mapper les données à vos modèles d'actifs. Ce processus garantit que les données de vos partenaires sont parfaitement intégrées dans votre AWS IoT SiteWise environnement, ce qui vous permet de les surveiller et de les analyser en même temps que vos autres sources de données.

Cette section répertorie les partenaires disponibles pour l'intégration de sources de données tierces sur les passerelles SiteWise Edge. Utilisez les informations ci-dessous pour configurer une source de données partenaire.

Note

Vous pouvez ajouter une source de données pour chaque partenaire dans chaque passerelle

CloudRail

Portail :

<https://devices.cloudrail.com/>

Prérequis

Pour plus d'informations sur CloudRail les exigences, consultez les [FAQ](#) sur le CloudRail site Web.

CloudRaildocumentation :

[Informatique de pointe : SiteWise Edge](#)

EasyEdge

Portail :

<https://studio.easyedge.io/>

Prérequis

[EasyEdgeexigences](#) — Informations sur les EasyEdge exigences, y compris les points de terminaison et les ports requis pour configurer le pare-feu. Remarque : vous aurez besoin d'un EasyEdge compte pour accéder à cette documentation.

EasyEdgedocumentation :

[EasyEdgepour AWS](#)

Litmus Edge

Accès à Litmus Edge Manager :

Pour accéder à Litmus Edge, configurez un compte [Litmus Edge Manager](#).

Prérequis

[Litmus EdgeExigences](#) : configurations recommandées et configuration système requise pour le déploiementLitmus Edge.

Litmusdocumentation :

- [Intégration à AWS IoT SiteWise](#)
- [Documentation Litmus Edge](#)

AWS IoT Greengrass composants pour AWS IoT SiteWise Edge

SiteWise Edge utilise des AWS IoT Greengrass composants pour collecter, traiter et transmettre des données industrielles à la périphérie. Ces composants fonctionnent ensemble pour permettre le traitement local des données et une intégration parfaite avec le service AWS IoT SiteWise cloud.

SiteWise Éditeur IoT

Le composant SiteWise éditeur IoT (`aws.iot.SiteWiseEdgePublisher`) est chargé de :

- Transmission sécurisée des données collectées vers le service AWS IoT SiteWise cloud

- Gestion de la mise en mémoire tampon des données et des nouvelles tentatives en cas de problèmes de connectivité

Pour plus d'informations sur la configuration de l'éditeur pour SiteWise Edge, consultez [Configuration du composant AWS IoT SiteWise éditeur](#). Et, pour plus d'informations sur le composant éditeur, consultez la section [IoT SiteWise Publisher](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

SiteWise Processeur IoT

Le composant SiteWise processeur IoT (`aws.iot.SiteWiseEdgeProcessor`) exécute les tâches suivantes :

- Exécution de transformations de données et de calculs à la périphérie
- Implémentation des définitions et des calculs des propriétés des actifs localement
- Réduction du volume de données en agrégeant ou en filtrant les données avant leur transmission

Pour plus d'informations sur le composant processeur, consultez la section [SiteWise Processeur IoT](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Collecteur IoT SiteWise OPC UA

Le composant collecteur IoT SiteWise OPC UA (`aws.iot.SiteWiseEdgeCollectorOpcua`) est conçu pour :

- Connectez-vous aux serveurs OPC UA dans les environnements industriels
- Collectez efficacement des données à partir de sources de données OPC UA
- Transformez les données OPC UA dans un format compatible avec AWS IoT SiteWise

Pour plus d'informations sur le composant collecteur OPC UA, consultez la section [Collecteur IoT SiteWise OPC UA](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

Simulateur de source de données IoT SiteWise OPC UA

Le composant simulateur de source de données IoT SiteWise OPC UA (`aws.iot.SiteWiseEdgeOpcuaDataSourceSimulator`) fournit les fonctionnalités suivantes :

- Démarre un serveur OPC UA local qui génère des exemples de données
- Simule une source de données qui peut être lue par le composant collecteur AWS IoT SiteWise OPC UA sur une passerelle AWS IoT SiteWise

- Permet d'explorer les AWS IoT SiteWise fonctionnalités à l'aide des exemples de données générés

Ce composant est particulièrement utile à des fins de test et de développement, car il vous permet de simuler des sources de données industrielles sans avoir besoin d'équipement physique.

Pour plus d'informations sur le composant de simulation de source de données, voir le [simulateur de source de données IoT SiteWise OPC UA](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

Ces AWS IoT Greengrass composants permettent d'activer les fonctionnalités SiteWise Edge. L' SiteWise éditeur IoT veille à ce que les données soient envoyées de manière fiable vers le cloud, le SiteWise processeur IoT gère les calculs locaux et l'optimisation des données, et le collecteur IoT SiteWise OPC UA facilite l'intégration aux protocoles industriels courants.

Note

Pour utiliser ces composants, vous devez les avoir installés AWS IoT Greengrass V2 ou les installer ultérieurement sur vos appareils Edge. Il est important de configurer correctement chaque composant pour optimiser les performances d' SiteWise Edge.

Filtrer les actifs sur une passerelle SiteWise Edge

Vous pouvez utiliser le filtrage périphérique pour gérer plus efficacement vos actifs en n'envoyant qu'un sous-ensemble d'actifs à une passerelle SiteWise Edge spécifique pour une utilisation dans le traitement des données. Si vos actifs sont organisés dans une structure arborescente, ou parent-enfant, vous pouvez configurer une politique IAM attachée au rôle IAM d'une passerelle SiteWise Edge qui autorise uniquement l'envoi de la racine de l'arborescence, ou parent, et de ses enfants à une passerelle Edge spécifique. SiteWise

Note

Si vous organisez des actifs existants dans une arborescence, après avoir créé la structure, accédez à chaque actif existant que vous avez ajouté à la structure et choisissez Modifier, puis cliquez sur Enregistrer pour vous assurer que la AWS IoT SiteWise nouvelle structure est reconnue.

Configurer le filtrage des bords

Configurez le filtrage Edge sur votre passerelle SiteWise Edge en ajoutant la politique IAM suivante au rôle IAM de la passerelle SiteWise Edge, en la *<root-asset-id>* remplaçant par l'ID de la ressource racine que vous souhaitez envoyer à la passerelle SiteWise Edge.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets"
      ],
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringNotLike": {
          "iotsitewise:assetHierarchyPath": "/*<root-asset-id>*"
        }
      }
    }
  ]
}
```

Si vous souhaitez supprimer des actifs se trouvant actuellement sur votre passerelle SiteWise Edge, connectez-vous à votre passerelle SiteWise Edge et exécutez la commande suivante pour forcer la passerelle SiteWise Edge à se synchroniser avec celle-ci AWS IoT SiteWise en supprimant le cache.

```
sudo rm /greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/sync-app/
sync_resource_bundles/edge.json
```

Configuration de la prise en charge des proxys et gestion des magasins de confiance pour AWS IoT SiteWise Edge

Dans AWS IoT SiteWise Edge, configurez et gérez les magasins de confiance afin de configurer la prise en charge de proxy pour vos appareils Edge. Configurez d'abord la configuration du proxy, puis configurez les magasins de confiance. Vous pouvez configurer les magasins de confiance pendant l'installation de la passerelle ou manuellement une fois celle-ci établie.

- Proxies — Facilitez la connectivité entre vos appareils et AWS services périphériques dans divers environnements réseau.
- Trust Stores : sécurisez les connexions en gérant les certificats fiables. Les configurations appropriées vous aident à respecter les politiques de sécurité de votre réseau, à permettre la communication dans des environnements réseau restreints et à optimiser le transfert de données entre les appareils de périphérie et les services cloud.

SiteWise Edge utilise plusieurs magasins de confiance pour différents types de composants, garantissant ainsi un flux de données sécurisé et efficace entre vos appareils Edge et le cloud. Vous pouvez configurer des magasins de confiance et des proxys sur une passerelle existante ou lors du processus d'installation lors de la création d'une nouvelle passerelle.

Exigences relatives aux configurations de Trust Store et de proxy

Avant de configurer un trust store ou d'installer SiteWise Edge avec des paramètres de proxy, assurez-vous que vous remplissez les conditions requises. Les exigences de mise en œuvre varient en fonction de l'utilisation de vos composants et de vos exigences en matière de fonctionnalités.

Exigences relatives à l'assistance par proxy

- URL de votre serveur proxy. L'URL doit inclure les informations utilisateur, le numéro de port de l'hôte. Par exemple, `scheme://[userinfo@]host[:port]`.
 - `scheme`— Doit être HTTP ou HTTPS
 - (Facultatif) `userinfo` — Informations sur le nom d'utilisateur et le mot de passe
 - `host`— Le nom d'hôte ou l'adresse IP du serveur proxy
 - `port`— Le numéro de port
- Liste d'adresses permettant de contourner le proxy.
- (Facultatif) Le fichier de certificat CA du proxy si vous utilisez un proxy HTTPS avec un certificat auto-signé.

Exigences du Trust Store

- Pour bénéficier de toutes les fonctionnalités du pack de traitement des données avec le proxy HTTPS, vous devez mettre à jour les trois magasins de confiance.
- Si vous utilisez uniquement le collecteur IoT SiteWise OPC UA et l' SiteWise éditeur IoT, mettez à jour les certificats AWS IoT Greengrass Core et Java Trust Stores vers la dernière version.

Bonnes pratiques pour les configurations Edge de Trust Store et de serveur proxy

Pour une maintenance continue et pour maintenir le plus haut niveau de sécurité dans votre environnement périphérique :

- Vérifiez et mettez à jour régulièrement les paramètres du proxy afin de les aligner sur les exigences de sécurité de votre réseau.
- Surveillez la connectivité de la passerelle et le flux de données pour garantir une communication proxy appropriée
- Gérez et mettez à jour les magasins de confiance conformément aux politiques de gestion des certificats de votre organisation
- Vous pouvez mettre en œuvre et suivre nos meilleures pratiques recommandées pour sécuriser les communications dans les environnements périphériques, telles que :
- Documentez les configurations de votre proxy et de votre magasin de confiance pour une visibilité opérationnelle
- Suivez les pratiques de sécurité de votre organisation pour la gestion des informations d'identification

Ces pratiques permettent de garantir la sécurité et la fiabilité des opérations de vos passerelles SiteWise Edge tout en restant conformes à vos politiques de sécurité générales.

Configurer les paramètres du proxy lors de l'installation de la passerelle AWS IoT SiteWise Edge

Vous pouvez configurer AWS IoT SiteWise Edge pour qu'il fonctionne avec un serveur proxy lors de l'installation de la passerelle. Le script d'installation prend en charge les proxys HTTP et HTTPS et peut configurer automatiquement des magasins de confiance pour des connexions proxy sécurisées.

Lorsque vous exécutez le script d'installation avec les paramètres du proxy, celui-ci exécute plusieurs tâches importantes :

- Valide le format et les paramètres de l'URL du proxy pour s'assurer qu'ils sont correctement spécifiés.
- Télécharge et installe les dépendances requises via le proxy configuré.
- Si un certificat d'autorité de certification proxy est fourni, il est ajouté au certificat d'autorité de certification AWS IoT Greengrass racine et importé dans Java KeyStore.

- Configure AWS IoT Greengrass (utilisé par SiteWise Edge) pour utiliser le proxy pour toutes les connexions sortantes.
- Termine l'installation d' SiteWise Edge avec les configurations de proxy et de trust store appropriées.

Pour configurer les paramètres du proxy lors de l'installation du logiciel de passerelle

1. Créez une passerelle SiteWise Edge. Pour plus d'informations, consultez [Créez une passerelle SiteWise Edge auto-hébergée](#) et [Installez le logiciel de passerelle AWS IoT SiteWise Edge sur votre appareil local](#).
2. Exécutez le script d'installation avec les paramètres de proxy appropriés à votre environnement. Remplacez les espaces réservés par vos informations de proxy spécifiques

Remplacez chacun des éléments suivants :

- -p, --proxy-url — URL du serveur proxy. L'URL doit être http soit https.
- -n, --no-proxy — Une liste d'adresses séparées par des virgules pour contourner le proxy.
- (Facultatif)-c, --proxy-ca-cert — Chemin d'accès au fichier de certificat CA du proxy.
- (Facultatif)-j, --javastorepass — Le KeyStore mot de passe Java. Le mot de passe par défaut est changeit.

Linux

Pour les systèmes Linux, utilisez la structure de commande suivante :

```
sudo ./install.sh -p proxy-url -n no-proxy-addresses [-c proxy-ca-cert-path] [-j javastorepass]
```

Windows

Pour les Microsoft Windows systèmes utilisant PowerShell, utilisez cette structure de commande :

```
.\install.ps1 -ProxyUrl proxy-url -NoProxyAddresses no-proxy-addresses [-ProxyCaCertPath proxy-ca-cert-path] [-JavaStorePass javastorepass]
```

Résolution des problèmes lors de l'installation activée par proxy

Pour plus d'informations sur la résolution des problèmes liés au trust store liés à une passerelle SiteWise Edge, consultez [Problèmes d'installation avec proxy](#).

Configurer manuellement les magasins de confiance pour la prise en charge du proxy HTTPS dans AWS IoT SiteWise Edge

Lorsque vous configurez des composants AWS IoT SiteWise Edge pour qu'ils se connectent via un proxy HTTPS, ajoutez le certificat du serveur proxy aux magasins de confiance appropriés. SiteWise Edge utilise plusieurs magasins de confiance pour sécuriser les communications. Il existe trois magasins de confiance et leur utilisation dépend du type de composant SiteWise Edge utilisé dans l'implémentation de votre passerelle.

Les magasins de confiance sont automatiquement mis à jour pendant le processus d'installation lorsque les paramètres de proxy sont fournis.

- [Configuration d'un magasin de confiance pour les composants AWS IoT Greengrass principaux](#)— Le certificat CA AWS IoT Greengrass racine est inclus dans les magasins de confiance pour vérifier l'authenticité des AWS services.

Ce trust store permet aux AWS IoT Greengrass composants de communiquer en toute sécurité avec les AWS services via le proxy tout en vérifiant l'authenticité de ces services.

- [Configuration d'un magasin de confiance pour les composants basé sur Java](#)— Le Java KeyStore (JKS) est le principal magasin de confiance utilisé par les composants Java pour SSL/TLS les connexions.

Les applications Java s'appuient sur le JKS pour établir des connexions sécurisées. Par exemple, si vous utilisez l' SiteWise éditeur IoT ou le collecteur IoT SiteWise OPC UA, qui sont basés sur Java, vous devez configurer ce trust store. Cela garantit que ces composants peuvent communiquer en toute sécurité via le proxy HTTPS lors de l'envoi de données vers le cloud ou lors de la collecte de données à partir de serveurs OPC UA.

- [Configuration du magasin de confiance des composants au niveau du système](#)— Lorsque vous utilisez des proxys HTTPS, leurs certificats doivent être ajoutés aux magasins de confiance appropriés pour permettre des connexions sécurisées.

Lorsque vous utilisez des proxys HTTPS, leurs certificats doivent être ajoutés aux magasins de confiance appropriés pour permettre des connexions sécurisées. Cela est nécessaire car les composants au niveau du système, souvent écrits dans des langages tels que Rust ou Go,

s'appuient sur le trust store du système plutôt que sur le JKS de Java. Par exemple, si vous utilisez des utilitaires système qui doivent communiquer via le proxy (pour les mises à jour logicielles ou la synchronisation de l'heure, par exemple), vous devez configurer le trust store au niveau du système. Cela garantit que ces composants et utilitaires peuvent établir des connexions sécurisées via le proxy.

Configuration d'un magasin de confiance pour les composants AWS IoT Greengrass principaux

Pour les fonctions AWS IoT Greengrass principales qui utilisent l'autorité de certification racine d'Amazon :

1. Localisez le fichier de certificat à l'adresse `/greengrass/v2/AmazonRootCA1.pem`
2. Ajoutez le certificat racine du proxy HTTPS (auto-signé) à ce fichier.

```
-----BEGIN CERTIFICATE-----
MIIETCCA v2gAwIQWgIVAMHSAzWG/5YVRYtRQ0xXUTEpHuEmApzGCSqGS Ib3DQEK
\ncwUAhUL9MQswCQwJVUzEPMAVUzEYMBYGA1UECgwP1hem9uLmNvbSBjbMUMRww
... content of proxy CA certificate ...
+vHIR1t0e5JAm5\noTIZGoFbK82A0/n07f/t5PSIDAim9V3Gc3pSXxCCAQoFYnui
GaPULGk1gCE84a0X\n7Rp/1ND/PuMZ/s8Yj1kY2NmYmNjMCAXDTE5MTEyN2cM216
gJMIADggEPADf2/m45hzEXAMPLE=
-----END CERTIFICATE-----

-----BEGIN CERTIFICATE-----
MIIDQTCCAimgF6AwIBAgITBmyfz/5mjAo54vB4ikPm1jZKYjANJmApzyMZFo6qBg
ADA5MQswCQYDVQQGEwJVUzEPMA0tMVT8QTPHRh8jrdkGA1UEChMGDV3QQDExBBKW
... content of root CA certificate ...
o/ufQJQWUCyziar1hem9uMRkwFwYVPSHCb2XV4cdFyQzR1KldZwgJcIQ6XUDgHaa
5MsI+yMRQ+hDaXJioblXgUka642M4UwtBV8oK2xJNDd2ZhwLnoQdeXeGADKkpy
rqXRfKoQnoZsG4q5WTP46EXAMPLE
-----END CERTIFICATE-----
```

Configuration du proxy HTTPS sur une passerelle établie

Vous pouvez ajouter la prise en charge du proxy à une passerelle établie en vous connectant au port 443 au lieu du port 8883. Pour plus d'informations sur l'utilisation d'un serveur proxy, voir [Connect sur le port 443 ou via un proxy réseau](#) dans le Guide du AWS IoT Greengrass Version 2 développeur. Si vous créez une nouvelle passerelle, vous pouvez définir la configuration du proxy lors de l'installation

de la passerelle. Pour de plus amples informations, veuillez consulter [Configurer les paramètres du proxy lors de l'installation de la passerelle AWS IoT SiteWise Edge](#).

Lorsque vous utilisez un proxy HTTPS avec AWS IoT Greengrass on SiteWise Edge, le logiciel choisit automatiquement entre HTTP et HTTPS pour les connexions proxy en fonction de l'URL fournie.

 Important

Mettez à jour tous les magasins de confiance requis avant de tenter de vous connecter via un proxy HTTPS.

Configuration d'un magasin de confiance pour les composants basé sur Java

Pour l' SiteWise éditeur IoT, le collecteur IoT SiteWise OPC UA et les services Java du pack de traitement de données, l'emplacement du magasin de confiance Java par défaut est `$JAVA_HOME/jre/lib/security/cacerts`

Pour ajouter un certificat

1. Créez un fichier pour stocker le certificat du serveur proxy, tel que `proxy.crt`.

 Note

Créez le fichier à l'avance à l'aide du certificat du serveur proxy.

2. Ajoutez le fichier au trust store de Java à l'aide de la commande suivante :

```
sudo keytool -import -alias proxyCert -keystore /usr/lib/jvm/java-11-openjdk-amd64/lib/security/cacerts -file proxy.crt
```

3. Lorsque vous y êtes invité, utilisez le mot de passe par défaut : `changeit`

Configuration du magasin de confiance des composants au niveau du système

Pour les composants écrits en Rust, Go et dans d'autres langages utilisant le System Trust Store :

Linux

Systèmes Linux : ajoutez des certificats à `/etc/ssl/certs/ca-certificates.crt`

Windows

Microsoft Windows systèmes : pour configurer le magasin de confiance, suivez la procédure du [magasin de certificats](#) décrite dans la documentation Microsoft Ignite.

Windows propose plusieurs magasins de certificats, y compris des magasins distincts pour les zones utilisateur et ordinateur, chacune comportant plusieurs sous-magasins. Pour la plupart des configurations SiteWise Edge, nous recommandons d'ajouter des certificats au COMPUTER | Trusted Root Certification Authorities magasin. Toutefois, en fonction de votre configuration spécifique et de vos exigences de sécurité, vous devrez peut-être utiliser un autre magasin.

Résolution des problèmes liés à Trust Store

Pour plus d'informations sur la résolution des problèmes liés au trust store liés à une passerelle SiteWise Edge, consultez [Problèmes liés à Trust Store](#).

Utiliser AWS IoT SiteWise APIs sur le bord

AWS IoT SiteWise en fournit un sous-ensemble APIs, ainsi que des éléments spécifiques à la périphérie APIs, permettant une interaction fluide avec les modèles d'actifs et leurs actifs associés déployés en périphérie. Ces modèles d'actifs doivent être configurés pour fonctionner en périphérie. Pour plus d'informations, consultez [Configuration d'un modèle d'actif pour le traitement des données sur SiteWise Edge](#) les instructions détaillées sur ce processus de configuration.

Après les avoir configurés APIs, vous pouvez récupérer des données complètes sur vos modèles d'actifs et vos actifs individuels. La récupération des informations sur le modèle d'actif, l'actif, le tableau de bord, le portail et le projet peut vous aider à surveiller les portails et les tableaux de bord déployés, et à accéder aux données des actifs collectées au niveau périphérique. Cela fournit un hôte central dans votre réseau pour les interactions AWS IoT SiteWise sans nécessiter un appel d'API Web.

Rubriques

- [Tous les appareils AWS IoT SiteWise Edge disponibles APIs](#)
- [Edge uniquement APIs pour une utilisation avec des appareils Edge AWS IoT SiteWise](#)
- [Activer CORS sur Edge AWS IoT SiteWise APIs](#)
- [Configurer les délais d'expiration des sessions pour Edge AWS IoT SiteWise](#)
- [Tutoriel : Répertoire des modèles d'actifs sur une passerelle AWS IoT SiteWise Edge](#)

Tous les appareils AWS IoT SiteWise Edge disponibles APIs

AWS IoT SiteWise propose une variété de périphériques APIs à utiliser en périphérie afin que vous puissiez effectuer des tâches localement sur l'appareil. Parmi les avantages disponibles, APIs citons la récupération de modèles d'actifs, la création et la mise à jour des propriétés des actifs et l'envoi de flux de données vers le cloud. En les exploitant APIs, vous pouvez créer des solutions capables de fonctionner dans des environnements où la connectivité réseau est intermittente ou limitée.

Disponible AWS IoT SiteWise APIs

Les éléments suivants AWS IoT SiteWise APIs sont disponibles sur les appareils Edge :

- [ListAssetModels](#)
- [DescribeAssetModel](#)
- [ListAssets](#)
- [DescribeAsset](#)
- [DescribeAssetProperty](#)
- [ListAssociatedAssets](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjectAssets](#)
- [ListProjects](#)
- [DescribeDashboard](#)
- [DescribePortal](#)
- [DescribeProject](#)

Disponible uniquement sur les bords APIs

APIs Les éléments suivants sont utilisés localement sur les appareils situés en périphérie :

- [Authentifier](#)— Utilisez cette API pour obtenir les informations d'identification temporaires SigV4 que vous utiliserez pour effectuer des appels d'API.

Edge uniquement APIs pour une utilisation avec des appareils Edge AWS IoT SiteWise

En plus de ceux AWS IoT SiteWise APIs disponibles sur le bord, il existe des modèles spécifiques aux bords. Ces caractéristiques spécifiques aux APIs arêtes sont décrites ci-dessous.

Authentifier

Obtient les informations d'identification de la passerelle SiteWise Edge. Vous devez ajouter des utilisateurs locaux ou vous connecter à votre système via LDAP ou un groupe d'utilisateurs Linux. Pour plus d'informations sur l'ajout d'utilisateurs, consultez [LDAP](#) ou [groupe d'utilisateurs Linux](#).

Syntaxe de demande

```
POST /authenticate HTTP/1.1
Content-type: application/json
{
  "username": "string",
  "password": "string",
  "authMechanism": "string"
}
```

Paramètres de demande d'URI

La demande n'utilise pas de paramètres URI.

Corps de la demande

Cette demande accepte les données suivantes au format JSON.

nom d'utilisateur

Le nom d'utilisateur utilisé pour valider l'appel de demande.

Type : String

Obligatoire : oui

mot de passe

Le mot de passe de l'utilisateur demandant les informations d'identification.

Type : String

Obligatoire : oui

authMechanism

Méthode d'authentification permettant de valider cet utilisateur sur l'hôte.

Type : String

Valeurs valides: ldap, linux, winnt

Obligatoire : oui

Syntaxe de réponse

```
HTTP/1.1 200
Content-type: application/json
{
  "accessKeyId": "string",
  "secretAccessKey": "string",
  "sessionToken": "string",
  "region": "edge"
}
```

Éléments de réponse

Si l'action aboutit, le service renvoie une réponse HTTP 200.

Les données suivantes sont renvoyées au format JSON.

accessKeyId

L'ID de clé d'accès qui identifie les informations d'identification de sécurité temporaires.

Contraintes de longueur : longueur minimale de 16. Longueur maximale de 128.

Modèle : `[\w]*`

secretAccessKey

La clé d'accès secrète qui peut être utilisée pour signer les demandes.

Type : String

sessionToken

Le jeton que les utilisateurs doivent transmettre à l'API du service pour utiliser les informations d'identification temporaires.

Type : String

region

La région que vous ciblez pour les appels d'API.

Type : CONSTANT - edge

Erreurs

IllegalArgumentException

La demande a été rejetée car le corps du document fourni était mal formé. Le message d'erreur décrit l'erreur spécifique.

Code d'état HTTP : 400

AccessDeniedException

L'utilisateur ne dispose pas d'informations d'identification valides basées sur le fournisseur d'identité actuel. Le message d'erreur décrit le mécanisme d'authentification.

Code d'état HTTP : 403

TooManyRequestsException

La demande a atteint sa limite de tentatives d'authentification. Le message d'erreur indique le temps d'attente avant que de nouvelles tentatives d'authentification ne soient effectuées.

Code d'état HTTP : 429

Activer CORS sur Edge AWS IoT SiteWise APIs

L'activation du CORS (Cross-Origin Resource Sharing) sur AWS IoT SiteWise Edge APIs permet aux applications Web de communiquer directement avec les APIs différents domaines. Cela permet une intégration fluide, un échange de données en temps réel et un accès aux données entre domaines

sans serveurs intermédiaires ni solutions de contournement. Les paramètres CORS peuvent être configurés pour spécifier les origines autorisées, garantissant ainsi un accès contrôlé entre les origines.

 Note

CORS est disponible pour les versions 3.3.1 et ultérieures du composant. Cette fonctionnalité est disponible pour les versions 3.3.1 et ultérieures du composant. `aws.iot.SiteWiseEdgeProcessor` Pour plus d'informations, consultez la section [AWS IoT SiteWise processeur](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Pour activer CORS sur Edge SiteWise APIs

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle SiteWise Edge pour laquelle vous souhaitez activer CORS. Vous pouvez activer CORS sur le type de AWS IoT Greengrass V2 déploiement.
4. Dans la section Configuration de la passerelle, choisissez le périphérique principal Greengrass associé.
5. Dans l'onglet Déploiements, sous Appareils Greengrass, sélectionnez le lien de déploiement approprié.
6. Sous Actions, choisissez Réviser, puis Réviser le déploiement.

 Important

La création d'une configuration CORS révisée remplace la configuration actuelle de l'appareil.

7. À l'étape 1, Spécifiez la cible, fournissez un nom facultatif pour identifier le déploiement.
8. À l'étape 2, Sélection des composants - facultatif, vous pouvez laisser toutes les sélections en cours telles quelles et choisir Suivant.
9. À l'étape 3, Configuration des composants - facultatif, sélectionnez `aws.iot.SiteWiseEdgeProcessor`, puis choisissez Configurer le composant.
10. Dans la section Mise à jour de la configuration, sous Configuration à fusionner, entrez le code JSON suivant :

```
{  
  "AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN": "*"   
}
```

Note

L'utilisation * comme valeur pour `AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` autorise toutes les origines. Pour les environnements de production, il est recommandé de spécifier l'origine exacte URLs pour une meilleure sécurité.

11. Choisissez Confirmer.
12. Choisissez Suivant pour passer aux étapes restantes jusqu'à ce que vous arriviez à l'étape 5, Révision.
13. Passez en revue vos modifications de configuration, puis choisissez Deploy pour appliquer les modifications à votre passerelle SiteWise Edge.

Note

Vous pouvez également activer CORS en définissant la variable `AWS_SITWISE_EDGE_ACCESS_CONTROL_ALLOW_ORIGIN` environnementale globale * sur votre AWS IoT SiteWise passerelle.

Note

Pour un proxy authentifié, `userinfo` il doit être inclus dans le `url` champ de la configuration du proxy plutôt que sous forme de `password` champs `username` et séparés.

Une fois le déploiement terminé, CORS est activé sur votre API SiteWise Edge, ce qui permet aux origines spécifiées d'envoyer des demandes d'origine croisée à l'API.

Configurer les délais d'expiration des sessions pour Edge AWS IoT SiteWise

SiteWise Edge vous permet de configurer les délais d'expiration des sessions pour l'API SiteWise Edge. Cette fonctionnalité améliore la sécurité en mettant automatiquement fin aux sessions inactives

après une période spécifiée. Cette section vous guide tout au long du processus de configuration du délai d'expiration de session à l'aide du Console AWS IoT SiteWise.

 Note

La configuration du délai d'expiration de session est disponible pour les versions 3.4.0 et ultérieures du `aws.iot.SiteWiseEdgeProcessor` composant. Pour plus d'informations, consultez la section [AWS IoT SiteWise processeur](#) dans le Guide du AWS IoT Greengrass Version 2 développeur.

Pour configurer un délai d'expiration de session pour une passerelle SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez la passerelle SiteWise Edge sur laquelle vous souhaitez configurer le délai d'expiration de session.

 Note

Vous pouvez configurer le délai d'expiration de session en fonction du type de AWS IoT Greengrass V2 déploiement.

4. Dans la section Configuration de la passerelle, choisissez le périphérique principal Greengrass associé.
5. Dans l'onglet Déploiements, sous Appareils Greengrass, sélectionnez le lien de déploiement approprié.
6. Sous Actions, sélectionnez Réviser. Lisez l'avertissement, puis choisissez Revise le déploiement.

 Important

La création d'une configuration de délai d'expiration de session révisée remplace la configuration actuelle de l'appareil.

7. À l'étape 1, Spécifiez la cible, fournissez un nom facultatif pour identifier le déploiement révisé, puis choisissez Next.

8. À l'étape 2, Sélection des composants - facultatif, vous pouvez laisser toutes les sélections en cours telles quelles et choisir Suivant.
9. À l'étape 3, Configuration des composants - facultatif, sélectionnez aws.iot.SiteWiseEdgeProcessor, puis choisissez Configurer le composant.
10. Dans la section Mise à jour de la configuration, sous Configuration à fusionner, entrez le code JSON suivant :

```
{  
  "AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES": "240"  
}
```

11. Définissez la valeur AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES en minutes. Les valeurs de délai d'expiration des sessions peuvent être comprises entre 1 minute et 10080 minutes (7 jours). La valeur par défaut est de 240 minutes (4 heures).
12. Choisissez Confirmer.
13. Choisissez Suivant pour passer aux étapes restantes jusqu'à ce que vous arriviez à l'étape 5, Révision.
14. Passez en revue vos modifications de configuration, puis choisissez Deploy pour appliquer les modifications à votre passerelle SiteWise Edge.

Note

Vous pouvez également configurer le délai d'expiration de session en définissant la variable environnementale globale AWS_SITEWISE_EDGE_SESSION_TIMEOUT_MINUTES sur la valeur souhaitée (en minutes) sur votre passerelle Edge. SiteWise

Une fois le déploiement terminé, la nouvelle configuration du délai d'expiration de session est appliquée à votre API SiteWise Edge.

Tutoriel : Répertoire des modèles d'actifs sur une passerelle AWS IoT SiteWise Edge

Vous pouvez utiliser un sous-ensemble des éléments disponibles AWS IoT SiteWise APIs et des éléments spécifiques APIs à la périphérie pour interagir avec les modèles d'actifs et leurs actifs en périphérie. Ce didacticiel vous expliquera comment obtenir des informations d'identification

temporaires sur une passerelle AWS IoT SiteWise Edge et comment obtenir une liste des modèles d'actifs sur la passerelle SiteWise Edge.

Prérequis

Dans les étapes de ce didacticiel, vous pouvez utiliser une variété d'outils. Pour utiliser ces outils, assurez-vous que les prérequis correspondants sont installés.

Pour suivre ce didacticiel, vous aurez besoin des éléments suivants :

- Un déployé et en cours d'exécution [AWS IoT SiteWise Exigences relatives à la passerelle auto-hébergée Edge](#)
- Accédez à votre passerelle SiteWise Edge sur le même réseau via le port 443.
- [OpenSSL installé](#)
- (AWS OpsHub pour AWS IoT SiteWise) L'[AWS OpsHubAWS IoT SiteWise application](#)
- (curl) [curl](#) installé
- (Python) [urllib3 installé](#)
- (Python) [Python3 installé](#)
- (Python) [Boto3 installé](#)
- (Python) [BotoCore](#) installé

Étape 1 : obtenir un certificat signé du service SiteWise Edge Gateway

Pour établir une connexion TLS avec la passerelle APIs disponible sur le SiteWise périphérique, vous avez besoin d'un certificat fiable. Vous pouvez générer ce certificat à l'aide d'un OpenSSL ou pour AWS OpsHub . AWS IoT SiteWise

OpenSSL

Note

[OpenSSL](#) doit être installé pour exécuter cette commande.

Ouvrez un terminal et exécutez la commande suivante pour obtenir un certificat signé auprès de la passerelle SiteWise Edge. Remplacez <sitewise_gateway_ip> par l'adresse IP de la passerelle SiteWise Edge.

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | openssl  
x509 -outform PEM > GatewayCert.pem
```

AWS OpsHub for AWS IoT SiteWise

Vous pouvez utiliser AWS OpsHub pour AWS IoT SiteWise. Pour de plus amples informations, veuillez consulter [Gérer les passerelles SiteWise Edge](#).

Le chemin absolu vers le certificat de passerelle SiteWise Edge téléchargé est utilisé dans ce didacticiel. Exécutez la commande suivante pour exporter le chemin complet de votre certificat, en le `<absolute_path_to_certificate>` remplaçant par le chemin d'accès au certificat :

```
export PATH_TO_CERTIFICATE='<absolute_path_to_certificate>'
```

Étape 2 : obtenir le nom d'hôte de votre passerelle SiteWise Edge

Note

[OpenSSL](#) doit être installé pour exécuter cette commande.

Pour terminer le didacticiel, vous aurez besoin du nom d'hôte de votre passerelle SiteWise Edge. Pour obtenir le nom d'hôte de votre passerelle SiteWise Edge, exécutez ce qui suit, en le `<sitewise_gateway_ip>` remplaçant par l'adresse IP de la passerelle SiteWise Edge :

```
openssl s_client -connect <sitewise_gateway_ip>:443 </dev/null 2>/dev/null | grep -Po  
'CN = \K.*' | head -1
```

Exécutez la commande suivante pour exporter le nom d'hôte à utiliser ultérieurement, en le `<your_edge_gateway_hostname>` remplaçant par le nom d'hôte de votre passerelle SiteWise Edge :

```
export GATEWAY_HOSTNAME='<your_edge_gateway_hostname>'
```

Étape 3 : obtenir des informations d'identification temporaires pour votre passerelle SiteWise Edge

Maintenant que vous avez le certificat signé et le nom d'hôte de votre passerelle SiteWise Edge, vous devez obtenir des informations d'identification temporaires pour pouvoir exécuter APIs sur la

passerelle. Vous pouvez obtenir ces informations d'identification via AWS OpsHub AWS IoT SiteWise ou directement depuis la passerelle SiteWise Edge à l'aide de APIs.

 Important

Les informations d'identification expirent toutes les 4 heures. Vous devez donc les obtenir juste avant d'utiliser la passerelle APIs on your SiteWise Edge. Ne mettez pas les informations d'identification en cache pendant plus de 4 heures.

Obtenez des informations d'identification temporaires à l'aide AWS OpsHub de for AWS IoT SiteWise

 Note

Vous devez installer l'[AWS IoT SiteWise application AWS OpsHub for](#).

AWS OpsHub Pour utiliser l' AWS IoT SiteWise application afin d'obtenir vos informations d'identification temporaires, procédez comme suit :

1. Connectez-vous à l'application.
2. Sélectionnez Paramètres.
3. Pour Authentification, choisissez Copier les informations d'identification.
4. Développez l'option adaptée à votre environnement et choisissez Copier.
5. Enregistrez les informations d'identification pour les utiliser ultérieurement.

Obtenez des informations d'identification temporaires à l'aide de l'API SiteWise Edge Gateway

Pour utiliser l'API de passerelle SiteWise Edge afin d'obtenir les informations d'identification temporaires, vous pouvez utiliser un script Python ou un curl. Vous devez d'abord disposer d'un nom d'utilisateur et d'un mot de passe pour votre passerelle SiteWise Edge. Les passerelles SiteWise Edge utilisent l'authentification et l'autorisation SigV4. Pour plus d'informations sur l'ajout d'utilisateurs, consultez [LDAP](#) ou [groupe d'utilisateurs Linux](#). Ces informations d'identification seront utilisées dans les étapes suivantes pour obtenir les informations d'identification locales sur votre passerelle SiteWise Edge qui sont nécessaires pour utiliser le AWS IoT SiteWise APIs.

Python

Note

Vous devez installer [urllib3](#) et [Python3](#).

Pour obtenir les informations d'identification à l'aide de Python

1. Créez un fichier appelé `get_credentials.py` et copiez-y le code suivant.

```
'''
The following demonstrates how to get the credentials from the SiteWise Edge
gateway. You will need to add local users or connect your system to LDAP/AD
https://docs.aws.amazon.com/iot-sitewise/latest/userguide/manage-gateways-
ggv2.html#create-user-pool

Example usage:
    python3 get_credentials.py -e https://<gateway_hostname> -c
    <path_to_certificate> -u '<gateway_username>' -p '<gateway_password>' -m
    '<method>'
'''
import urllib3
import json
import urllib.parse
import sys
import os
import getopt

'''
This function retrieves the AWS IoT SiteWise Edge gateway credentials.
'''
def get_credentials(endpoint, certificatePath, user, password, method):
    http = urllib3.PoolManager(cert_reqs='CERT_REQUIRED', ca_certs=
    certificatePath)
    encoded_body = json.dumps({
        "username": user,
        "password": password,
        "authMechanism": method,
    })

    url = urllib.parse.urljoin(endpoint, "/authenticate")
```

```

response = http.request('POST', url,
                        headers={'Content-Type': 'application/json'},
                        body=encoded_body)

if response.status != 200:
    raise Exception(f'Failed to authenticate! Response status
{response.status}')

auth_data = json.loads(response.data.decode('utf-8'))

accessKeyId = auth_data["accessKeyId"]
secretAccessKey = auth_data["secretAccessKey"]
sessionToken = auth_data["sessionToken"]
region = "edge"

return accessKeyId, secretAccessKey, sessionToken, region

def print_help():
    print('Usage:')
    print(f'{os.path.basename(__file__)} -e <endpoint> -c <path/to/certificate>
-u <user> -p <password> -m <method> -a <alias>')
    print('')
    print('-e, --endpoint    edge gateway endpoint. Usually the Edge gateway
hostname.')
    print('-c, --cert_path path to downloaded gateway certificate')
    print('-u, --user        Edge user')
    print('-p, --password    Edge password')
    print('-m, --method      (Optional) Authentication method (linux, winnt,
ldap), default is linux')
    sys.exit()

def parse_args(argv):
    endpoint = ""
    certificatePath = None
    user = None
    password = None
    method = "linux"

    try:
        opts, args = getopt.getopt(argv, "he:c:u:p:m:",
["endpoint=", "cert_path=", "user=", "password=", "method="])
    except getopt.GetoptError:
        print_help()

```

```
for opt, arg in opts:
    if opt == '-h':
        print_help()
    elif opt in ("-e", "--endpoint"):
        endpoint = arg
    elif opt in ("-u", "--user"):
        user = arg
    elif opt in ("-p", "--password"):
        password = arg
    elif opt in ("-m", "--method"):
        method = arg.lower()
    elif opt in ("-c", "--cert_path"):
        certificatePath = arg

if method not in ['ldap', 'linux', 'winnt']:
    print("not valid method parameter, required are ldap, linux, winnt")
    print_help()

if (user == None or password == None):
    print("To authenticate against edge user, password have to be passed
together, and the region has to be set to 'edge'")
    print_help()

if(endpoint == ""):
    print("You must provide a valid and reachable gateway hostname")
    print_help()

return endpoint,certificatePath, user, password, method

def main(argv):
    # get the command line args
    endpoint, certificatePath, user, password, method = parse_args(argv)

    accessKeyId, secretAccessKey, sessionToken, region=get_credentials(endpoint,
certificatePath, user, password, method)

    print("Copy and paste the following credentials into the shell, they are
valid for 4 hours:")
    print(f"export AWS_ACCESS_KEY_ID={accessKeyId}")
    print(f"export AWS_SECRET_ACCESS_KEY={secretAccessKey}")
    print(f"export AWS_SESSION_TOKEN={sessionToken}")
    print(f"export AWS_REGION={region}")
```

```
print()
```

```
if __name__ == "__main__":  
    main(sys.argv[1:])
```

2. Exécutez `get_credentials.py` depuis le terminal en remplaçant `<gateway_username>` et en utilisant `<gateway_password>` les informations d'identification que vous avez créées.

```
python3 get_credentials.py -e https://$GATEWAY_HOSTNAME -c $PATH_TO_CERTIFICATE  
-u '<gateway_username>' -p '<gateway_password>' -m 'linux'
```

curl

 Note

Vous devez installer [Curl](#).

Pour obtenir les informations d'identification à l'aide de curl

1. Exécutez la commande suivante depuis le terminal en remplaçant `<gateway_username>` et en utilisant `<gateway_password>` les informations d'identification que vous avez créées.

```
curl --cacert $PATH_TO_CERTIFICATE --location \  
-X POST https://$GATEWAY_HOSTNAME:443/authenticate \  
--header 'Content-Type: application/json' \  
--data-raw '{  
    "username": "<gateway_username>",  
    "password": "<gateway_password>",  
    "authMechanism": "linux"  
}'
```

Les résultats doivent avoir l'aspect suivant :

```
{  
    "username": "sweuser",  
    "accessKeyId": "<accessKeyId>",
```



```
"secretAccessKey": "<secretAccessKey>",
"sessionToken": "<sessionToken>",
"sessionExpiryTime": "2022-11-17T04:51:40.927095Z",
"authMechanism": "linux",
"role": "edge-user"
}
```

2. Exécutez la commande suivante à partir de votre terminal.

```
export AWS_ACCESS_KEY_ID=<accessKeyId>
export AWS_SECRET_ACCESS_KEY=<secretAccessKey>
export AWS_SESSION_TOKEN=<sessionToken>
export AWS_REGION=edge
```

Étape 4 : obtenir une liste des modèles d'actifs sur la passerelle SiteWise Edge

Maintenant que vous disposez d'un certificat signé, du nom d'hôte de votre passerelle SiteWise Edge et des informations d'identification temporaires pour votre passerelle SiteWise Edge, vous pouvez utiliser l'`ListAssetModelsAPI` pour obtenir une liste des modèles d'actifs de votre passerelle SiteWise Edge.

Python

Note

Vous devez installer [Python3](#) et [Boto3](#). [BotoCore](#)

Pour obtenir la liste des modèles d'actifs à l'aide de Python

1. Créez un fichier appelé `list_asset_model.py` et copiez-y le code suivant.

```
import json
import boto3
import botocore
import os

# create the client using the credentials
client = boto3.client("iotsitewise",
    endpoint_url= "https://" + os.getenv("GATEWAY_HOSTNAME"),
    region_name=os.getenv("AWS_REGION"),
```

```
aws_access_key_id=os.getenv("AWS_ACCESS_KEY_ID"),
aws_secret_access_key=os.getenv("AWS_SECRET_ACCESS_KEY"),
aws_session_token=os.getenv("AWS_SESSION_TOKEN"),
verify=os.getenv("PATH_TO_CERTIFICATE"),
config=botocore.config.Config(inject_host_prefix=False))

# call the api using local credentials
response = client.list_asset_models()
print(response)
```

2. Exécutez `list_asset_model.py` depuis le terminal.

```
python3 list_asset_model.py
```

curl

Note

Vous devez installer [Curl](#).

Pour obtenir la liste des modèles d'actifs à l'aide de curl

Exécutez la commande suivante depuis le terminal.

```
curl \
--request GET https://$GATEWAY_HOSTNAME:443/asset-models \
--cacert $PATH_TO_CERTIFICATE \
--aws-sigv4 "aws:amz:edge:iotsitewise" \
--user "$AWS_ACCESS_KEY_ID:$AWS_SECRET_ACCESS_KEY" \
-H "x-amz-security-token:$AWS_SESSION_TOKEN"
```

Les résultats doivent avoir l'aspect suivant :

```
{
  "assetModelSummaries": [
    {
      "arn": "arn:aws:iotsitewise:{region}:{account-id}:asset-model/{asset-
model-id}",
      "creationDate": 1.669245291E9,
      "description": "This is a small example asset model",
```

```
    "id": "{asset-model-id}",
    "lastUpdateDate": 1.669249038E9,
    "name": "Some Metrics Model",
    "status": {
      "error": null,
      "state": "ACTIVE"
    }
  },
  .
  .
  .
],
"nextToken": null
}
```

Hébergez une passerelle SiteWise Edge sur Siemens Industrial Edge

Hébergez votre passerelle sur Siemens Industrial Edge à l'aide de l'application AWS IoT SiteWise Edge. Tout comme avec AWS IoT Greengrass V2, vous pouvez optimiser les processus de fabrication ou améliorer les flux de travail opérationnels à l'aide de l' SiteWise Edge onSiemens Industrial Edge.

Vous pouvez intégrer les données de votre appareil Siemens Industrial Edge à votre AWS compte en exécutant une passerelle SiteWise Edge sur l'appareil. Pour ce faire, demandez l'accès à l'application AWS IoT SiteWise Edge auprès de l'équipe d'assistance SiteWise Edge. Créez ensuite une ressource de passerelle SiteWise Edge avec une cible de déploiement du périphérique Siemens Industrial Edge - nouveau. Téléchargez ensuite le fichier de configuration et chargez-le dans votre application via le Siemens Industrial Edge Management portail. Pour plus d'informations sur l'exécution d'applications Siemens Industrial Edge, notamment sur la configuration Siemens des ressources requises, voir [Qu'est-ce qu'Industrial Edge ?](#) dans la documentation de Siemens.

Note

Siemens n'est ni un vendeur ni un fournisseur d' SiteWise Edge. La Siemens Industrial Edge Marketplace est une place de marché indépendante.

Rubriques

- [Sécurité](#)
- [Siemens Secure Storage et l'application AWS IoT SiteWise Edge](#)
- [Destinations pour les Siemens Industrial Edge appareils](#)
- [Migrer depuis l'application de prévisualisation](#)
- [Résolution des problèmes](#)
- [AWS IoT SiteWise Journal des modifications de l'application Edge](#)
- [Exigences relatives à l'application AWS IoT SiteWise Edge](#)
- [Créez une passerelle pour Siemens Industrial Edge](#)
- [Créez un Siemens Databus user pour l'application](#)
- [Accédez à l'application AWS IoT SiteWise Edge](#)
- [Installation de l'application sur un Siemens appareil](#)
- [Mettre à jour la configuration de l'application AWS IoT SiteWise Edge](#)

Sécurité

Dans le cadre du [modèle de responsabilité partagée](#) entre AWS nos clients et nos partenaires, les informations suivantes décrivent qui est responsable des différents aspects de la sécurité :

Responsabilité du client

- Sélection du partenaire.
- Configuration de l'accès réseau accordé au partenaire.
- Sécurisation physique de l'appareil exécutant SiteWise Edge.

AWS responsabilité

- Isoler le partenaire des ressources AWS cloud du client.

Responsabilité du partenaire

- Utilisation de valeurs par défaut sécurisées.
- Garantir la sécurité de la solution au fil du temps grâce à des correctifs et à d'autres mises à jour appropriées.
- Préserver la confidentialité des données des clients.
- Vérification des autres applications disponibles sur le marché des partenaires.

Siemens Secure Storage et l'application AWS IoT SiteWise Edge

Pour protéger les informations d'identification et les secrets requis pour exécuter l'application AWS IoT SiteWise Edge, Siemens Industrial Edge fournit des mécanismes permettant de stocker en toute sécurité les informations d'identification sur l'appareil. L'application AWS IoT SiteWise Edge ne s'exécutera pas sur un appareil s'il n'est pas compatible avec le stockage sécurisé de ces informations d'identification. Les échecs d'exécution dus à l'absence de support de stockage sécurisé sont enregistrés dans des fichiers journaux.

Les versions minimales du système d'exploitation suivantes sont requises pour installer et exécuter l'application AWS IoT SiteWise Edge. Mettez à niveau vos appareils vers les dernières versions pour installer l'application.

- Pour les appareils virtuels : IEVD version 1.19 ou supérieure
- Pour les appareils physiques : IED-OS version 2.2 ou supérieure

L'application AWS IoT SiteWise Edge activée ne s'exécutera pas tant que vous n'aurez pas mis à niveau votre appareil.

Destinations pour les Siemens Industrial Edge appareils

Lorsque vous utilisez l'application AWS IoT SiteWise Edge Siemens Industrial Edge, les destinations aident à préparer les données avant de les envoyer à des AWS IoT SiteWise fins d'analyse et de distribution ultérieures. Vous pouvez configurer les paramètres de destination des données pour l'ingestion de données en mémoire tampon à l'aide d'Amazon S3 ou utiliser l'ingestion de données en temps réel. Les deux vous permettent de vous abonner à des rubriques MQTT à l'aide de filtres de chemin sur la cible de déploiement de l'appareil Siemens Industrial Edge.

La cible Siemens Industrial Edge de déploiement sur votre passerelle prend en charge deux méthodes principales de traitement des données :

- AWS IoT SiteWise paramètres en temps réel - Les données sont envoyées directement au AWS IoT SiteWise fur et à mesure de leur collecte
- AWS IoT SiteWise mise en mémoire tampon à l'aide des paramètres Amazon S3 - Les données sont collectées et stockées temporairement par lots avant d'être envoyées à Amazon S3

Pour plus d'informations sur la configuration de ces options, reportez-vous [Ajouter une destination AWS IoT SiteWise mise en mémoire tampon à l'aide d'Amazon S3](#) aux sections et [Ajouter une destination AWS IoT SiteWise Edge en temps réel](#).

Préfixes pour les filtres de chemin

Les filtres de chemin pour les passerelles utilisant des cibles de Siemens Industrial Edge déploient combinent à la fois le sujet et le nom du flux de données afin de créer un identifiant unique pour vos données. Le sujet combiné au nom du flux de données est appelé préfixe dans les Siemens Industrial Edge passerelles. Cela diffère des passerelles auto-hébergées où les filtres de chemin sont basés uniquement sur des sujets MQTT.

Exemple Structure de filtre de chemin pour les flux de données Siemens

Un filtre de chemin typique pour un flux de données Siemens inclut à la fois le chemin du sujet et le nom du flux de données :

```
ie/d/device1/application1/datastream1
```

Où :

- `ie/d/` est le préfixe obligatoire pour les flux de données Siemens
- `device1/application1` représente le chemin hiérarchique
- `datastream1` est le nom du flux de données spécifique

Note

Lorsque vous travaillez avec Siemens Industrial Edge des flux de données, veillez à inclure à la fois les rubriques métadonnées (`ie/m/`) et données (`ie/d/`) dans vos filtres de chemin pour recevoir des informations complètes sur vos flux de données.

Filtres de destinations et de chemins

Consultez les rubriques suivantes pour en savoir plus sur les destinations et les filtres de chemin dans les passerelles compatibles MQTT :

- [Comprendre les destinations AWS IoT SiteWise Edge](#)
- [Ajouter une destination AWS IoT SiteWise Edge en temps réel](#)

- [Ajouter une destination AWS IoT SiteWise mise en mémoire tampon à l'aide d'Amazon S3](#)
- [Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge](#)
- [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#)
- [Gérer les destinations AWS IoT SiteWise Edge](#)

Migrer depuis l'application de prévisualisation

Si vous avez SiteWise utilisé Edge Siemens Industrial Edge pendant la phase de prévisualisation, vous devrez passer de la version d'aperçu, version 1.0.1, à la dernière version. Procédez comme suit pour effectuer la migration :

1. Créez de nouvelles passerelles SiteWise Edge. Pour de plus amples informations, veuillez consulter [Créez une passerelle pour Siemens Industrial Edge](#).
2. Créez une nouvelle passerelle Siemens Databus user pour chaque nouvelle passerelle. Pour de plus amples informations, veuillez consulter [Créez un Siemens Databus user pour l'application](#).
3. Désinstallez l'application de passerelle AWS IoT SiteWise Edge version 1.0.1 sur votre IED.

Note

Préparez-vous aux interruptions du flux de données lorsque vous reconfigurez les AWS IoT SiteWise actifs précédemment utilisés par la version préliminaire de l'application AWS IoT SiteWise Edge. Bien que l'historique des données soit préservé, il existe un risque de perte de données lors de la réinstallation de la nouvelle passerelle.

4. Supprimez les passerelles SiteWise Edge que vous avez créées lors de l'aperçu dans le [Console AWS IoT SiteWise](#).
5. Installez l'application de passerelle AWS IoT SiteWise Edge sur IED à l'aide du nouveau fichier de configuration de passerelle. Pour de plus amples informations, veuillez consulter [Installation de l'application sur un Siemens appareil](#).

Important

L'installation de la nouvelle passerelle remplace la version préliminaire de l'application SiteWise Edge. Il n'est pas possible de revenir à la version 1.0.1 après avoir installé la version 2.0.0.

Après avoir configuré la nouvelle passerelle Siemens Databus user, vos données sont transmises à vos propriétés.

Vous pouvez également mettre à niveau votre application SiteWise Edge directement à partir de la version. Cependant, une nouvelle configuration de passerelle est toujours nécessaire.

Résolution des problèmes

Pour résoudre les problèmes liés à la passerelle SiteWise Edge sur votre Siemens Industrial Edge appareil, consultez [Résolution des problèmes liés à l'application AWS IoT SiteWise Edge sur Siemens Industrial Edge](#).

Vous pouvez également accéder à [AWS Re:Post](#) pour trouver des réponses à vos questions.

AWS IoT SiteWise Journal des modifications de l'application Edge

Le tableau suivant décrit les modifications apportées à chaque version de l'application AWS IoT SiteWise Edge.

Version	Modifications	
3.0.0	Nouvelles fonctionnalités • Ajoute des options de configuration supplémentaires dans la AWS IoT SiteWise console pour les passerelles installées sur Siemens Industrial Edge : • Ajoutez et configurez des destinations Amazon S3. • Supprimez la destination AWS IoT SiteWise en temps réel. • Ajoutez des filtres pour chaque destination afin de contrôler le routage des données. • Ajoute l'option d'une configuration de préfixe globale pour envoyer toutes les données de l' SiteWise éditeur IoT dans le Siemens IE programme d'installation. Le préfixe est appliqué après le filtrage.	
	Corrections de bogues et améliorations	

Version	Modifications	
	Permet de réessayer de se connecter au bus de données Siemens Industrial Edge en cas d'échec de la tentative initiale de l'application AWS IoT SiteWise Edge.	
2.0.1	Corrections de bogues et améliorations Résout un problème selon lequel l'application entrait en état d'échec et se fermait si elle ne parvenait pas à récupérer les AWS informations d'identification au démarrage. Permet de réessayer jusqu'à ce que la récupération des informations d'identification soit réussie.	
2.0.0	- L'application AWS IoT SiteWise Edge est désormais disponible pour tous. - L'application nécessite la version 1.19 de Siemens IEVD ou la version 2.2 de Siemens IED-OS. - Améliorations des performances : réduction de l'utilisation de la mémoire et du processeur. - Améliorations du débogage : vous pouvez désormais télécharger un fichier de configuration facultatif pour activer les journaux de débogage. - Améliorations de sécurité : l'application permet SecureStorage API de stocker en toute sécurité les informations d'identification sur l'appareil. - Dockervaleur du condensé : sha256:4a960f29234a190ebb5224c1fd0f3e99faafccc4cb3d93ca13fef247b6656d18	
1.0.1	Première version	

Exigences relatives à l'application AWS IoT SiteWise Edge

Pour exécuter AWS IoT SiteWise Edge onSiemens Industrial Edge, vous devez disposer des éléments suivants :

- Un compte [Siemens Digital Exchange Platform](#).
- Un compte Siemens Industrial Edge Hub (iehub).
- Une Siemens Industrial Edge Management instance.
 - Le service de configuration de l'application IE. Pour en savoir plus, consultez [Installing the IE App Configuration Service manually](#) la Siemens Industrial Edge Management documentation.
- Accès à la version 2.0.1 ou supérieure de l'application AWS IoT SiteWise Edge. Pour de plus amples informations, veuillez consulter [Accédez à l'application AWS IoT SiteWise Edge](#).
- Il s'agit soit d'un Siemens Industrial Edge périphérique (IED), soit d'un périphérique Siemens Industrial Edge virtuel (IEVD).
 - Un minimum de 15 Go d'espace disque pour répondre aux exigences matérielles.
 - 1 Go de RAM avec 1 Go de mémoire d'échange supplémentaire.
 - Configuration de l'appareil pour autoriser le trafic sortant sur les ports 443 et 8883.
 - Un processeur x86-64 bits.
 - Siemens Industrial Edge Management version 1.13.10 ou supérieure.
 - Conformité de l'appareil aux Siemens Secure Storage exigences.
 - Sur les appareils virtuels, IEVD version 1.19 ou supérieure.
 - Sur les appareils physiques, IDE-OS version 2.2 ou supérieure.
 - La dernière version de Docker Compose.
 - Docker Engine version 18.091 ou supérieure.
- Accès au domaine requis. Pour de plus amples informations, veuillez consulter [AWS IoT SiteWise points de terminaison](#).

Créez une passerelle pour Siemens Industrial Edge

Une fois que vous disposez des comptes Siemens et des instances IEM appropriés, vous pouvez créer une passerelle SiteWise Edge du type de déploiement : appareil Siemens Industrial Edge.

Note

Assurez-vous de répondre à toutes les exigences relatives à l'utilisation d'un appareil Siemens Industrial Edge Management. Pour de plus amples informations, veuillez consulter [Exigences relatives à l'application AWS IoT SiteWise Edge](#).

Pour créer le fichier de configuration

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Cliquez sur Create gateway (Créer une passerelle).
4. Pour le type de déploiement, choisissez l'appareil Siemens Industrial Edge.
5. Entrez un nom pour votre passerelle SiteWise Edge ou utilisez le nom généré par AWS IoT SiteWise.
6. (Facultatif) Dans la section Configuration avancée, procédez comme suit :
 - Entrez un nom pour votre AWS IoT Core objet ou utilisez le nom généré par AWS IoT SiteWise.
7. Cliquez sur Create gateway (Créer une passerelle).
8. Dans la boîte de dialogue Générer le fichier de configuration de la passerelle SiteWise Edge, sélectionnez Générer et télécharger. AWS IoT SiteWise génère automatiquement un fichier de configuration que vous utiliserez pour configurer l'application AWS IoT SiteWise Edge.

Important

Vous utilisez le fichier de configuration de la passerelle pour sauvegarder et restaurer votre application AWS IoT SiteWise Edge. Enregistrez le fichier de configuration de votre passerelle SiteWise Edge [AWS Secrets Manager](#) pour le stocker et le gérer en toute sécurité. Secrets Manager stocke, gère et récupère les informations sensibles en toute sécurité.

Créez un Siemens Databus user pour l'application

AWS IoT SiteWise Edge on Siemens Industrial Edge ingère les données de l'Siemens Databus application. Pour connecter SiteWise Edge au Siemens Databus, vous avez besoin d'un Siemens Databus user appareil qui donne accès aux données vers lesquelles vous souhaitez effectuer un transfert sécurisé AWS IoT SiteWise. Pour commencer, créez un, Siemens Databus user puis fournissez les informations d'identification à l'application SiteWise Edge.

Pour créer une Siemens Databus user

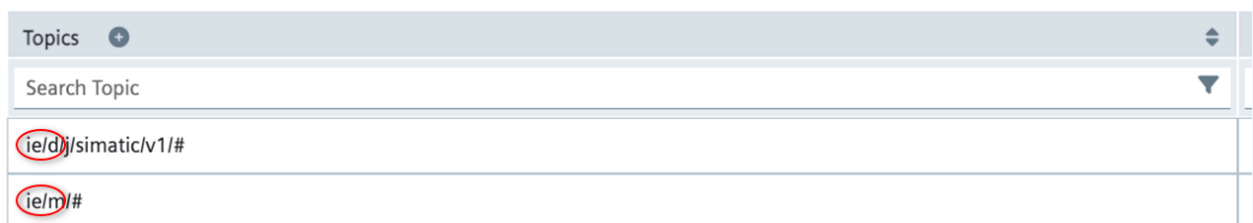
1. Dans votre Siemens Industrial Edge Management instance, choisissez Edge Management dans la section Applications de plateforme.
2. Cliquez sur l'icône Connexions de données.
3. Sélectionnez Databus. La liste de vos appareils connectés s'affiche.
4. Sélectionnez l'appareil à connecter à l'application AWS IoT SiteWise Edge.
5. Choisissez Lancer. Le correspondant Databus Configurator à l'appareil que vous avez sélectionné apparaît.
6. Créez un utilisateur pour votre appareil Edge sous Utilisateurs. Pour plus d'informations sur la création d'un utilisateur, consultez la section [Utilisateurs](#) dans la documentation de Siemens Industrial Edge Management.
7. Sélectionnez les sujets auxquels cela Siemens Databus devrait avoir accès. Ces rubriques limitent les accès auxquels AWS IoT SiteWise Edge peut accéder.

Important

Tous les sujets auxquels Siemens Databus user a accès sont publiés sur AWS IoT SiteWise.

Note

Siemens Databus useront besoin d'accéder à la fois aux rubriques relatives aux données et aux métadonnées. Les sujets commençant par `ie/d/` sont des sujets relatifs aux données. Et les sujets qui commencent par `ie/m/` sont des sujets relatifs aux métadonnées. Partagez des sujets par paires afin qu' SiteWise Edge ait accès aux données et aux métadonnées de chaque sujet.



8. Définissez les autorisations appropriées pour votre Siemens Databus configuration.

Après avoir créé votre Siemens Databus configuration, vous pouvez installer l'application AWS IoT SiteWise Edge sur votre Siemens Industrial Edge Management. Pour de plus amples informations, veuillez consulter [Installation de l'application sur un Siemens appareil](#).

Vous pouvez également éventuellement configurer des destinations et des filtres de chemin pour votre Siemens Industrial Edge passerelle. Pour de plus amples informations, veuillez consulter [Destinations et filtres de parcours](#).

Accédez à l'application AWS IoT SiteWise Edge

Pour accéder à l'application AWS IoT SiteWise Edge sur Siemens Industrial Edge, [envoyez un e-mail](#) demandant l'accès à l'équipe d'assistance SiteWise Edge.

Incluez les informations suivantes dans votre e-mail :

- Votre nom et vos coordonnées
- Nom de la société
- Siemens Industrial Edge identifiant du locataire

Installation de l'application sur un Siemens appareil

Après avoir accédé à l'application AWS IoT SiteWise Edge en envoyant un e-mail à l'équipe d'assistance SiteWise Edge pour Siemens Industrial Edge, assignez l'application à une instance de Siemens Industrial Edge Management. Vous pouvez ensuite installer l'application AWS IoT SiteWise Edge sur votre appareil.

Pour installer l'application AWS IoT SiteWise Edge

1. Vérifiez que le Docker résumé fourni Siemens Industrial Edge Management correspond à la dernière version répertoriée dans le [AWS IoT SiteWise Journal des modifications de l'application Edge](#).

Pour plus d'informations sur la localisation de la valeur Docker condensée pour Siemens, consultez la section [Gestion d'une application](#) dans le Siemens Industrial Edge Device de la Siemens documentation.

Siemens Industrial Edge Management prend en charge une version de l'application AWS IoT SiteWise Edge à la fois. Effectuez cette étape pour vous assurer que vous utilisez la dernière

version de l'application avant d'installer l'application AWS IoT SiteWise Edge sur votre Siemens Industrial Edge appareil.

2. Attribuez l'application AWS IoT SiteWise Edge à Siemens Industrial Edge Management. Pour plus d'informations, consultez [la section Gestion d'une application](#) dans la section Industrial Edge Management de la Siemens documentation.
3. Dans Edge Management, parcourez le catalogue de l'AWS IoT SiteWise Edge et choisissez-le.
4. Choisissez Installer.

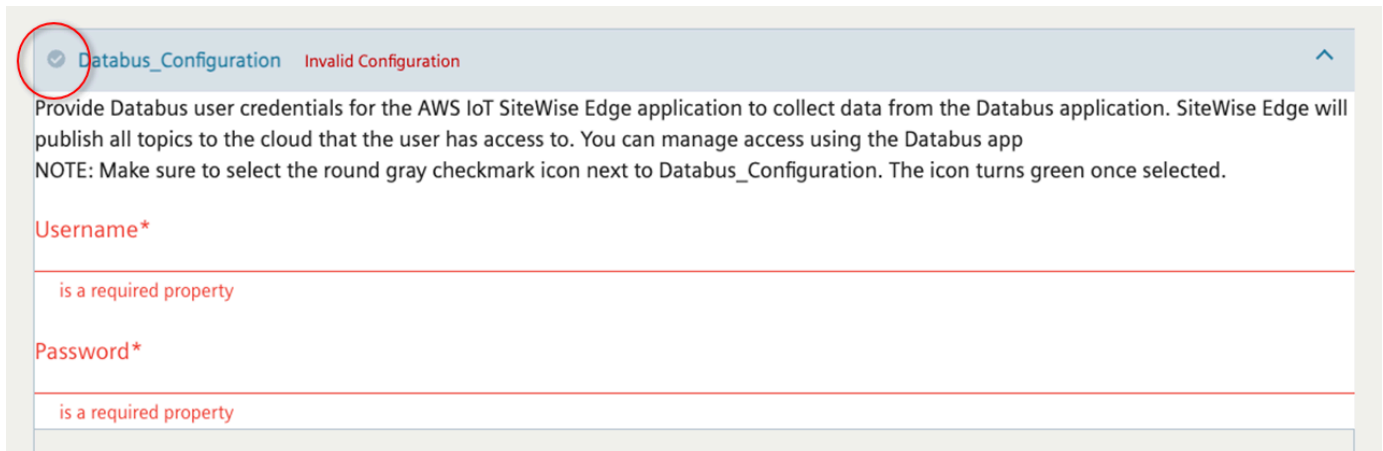
 Note

Si un bouton Contactez-nous s'affiche, choisissez-le et suivez les étapes pour demander l'accès à l'application AWS IoT SiteWise Edge activée Siemens Industrial Edge. Pour de plus amples informations, veuillez consulter [Accédez à l'application AWS IoT SiteWise Edge](#).

5. Sélectionnez Databus_Configuration dans les options de configuration du schéma.
6. Entrez le nom d'utilisateur et le mot de passe pour la configuration du bus de données. Pour plus d'informations sur la création d'un Siemens Databus user, consultez [Créez un Siemens Databus user pour l'application](#).
7. Choisissez la petite icône ronde en forme de coche grise située à côté de Databus_Configuration pour que l'icône devienne verte.

 Note

Les configurations d'entrée ne s'appliquent que si l'icône en forme de coche passe du gris au vert. Dans le cas contraire, la configuration d'entrée est ignorée.



 **Databus_Configuration** Invalid Configuration 

Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app

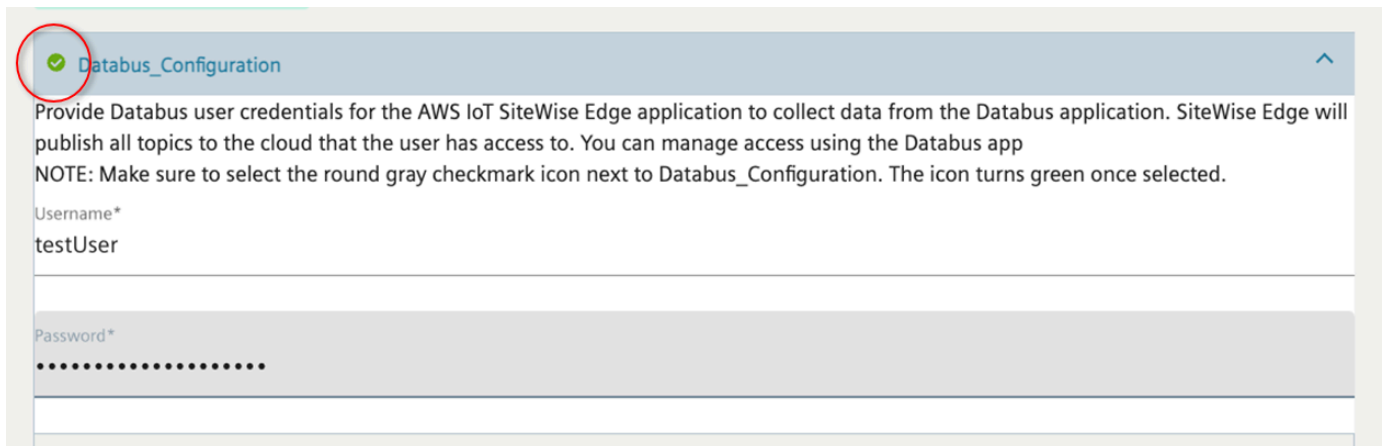
NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

is a required property

Password*

is a required property



 **Databus_Configuration** 

Provide Databus user credentials for the AWS IoT SiteWise Edge application to collect data from the Databus application. SiteWise Edge will publish all topics to the cloud that the user has access to. You can manage access using the Databus app

NOTE: Make sure to select the round gray checkmark icon next to Databus_Configuration. The icon turns green once selected.

Username*

testUser

Password*

.....

8. Choisissez Suivant pour passer à Autres configurations où vous pouvez télécharger le fichier de configuration de votre passerelle.
9. Choisissez SiteWise_Edge_Gateway_Config comme emplacement pour télécharger le fichier de configuration de la passerelle.

 Note

Assurez-vous de choisir SiteWise_Edge_Gateway_Config plutôt que _Edge_Support_Config_Optional. SiteWise

10. Sélectionnez l'appareil sur lequel installer l'application.
11. Choisissez Install Now (Installer maintenant).

Vous pouvez éventuellement configurer le composant éditeur pour exporter les données vers le AWS Cloud. Pour plus d'informations, voir [configurer le composant AWS IoT SiteWise éditeur](#).

Pour configurer les destinations de votre Siemens Industrial Edge passerelle, consultez [Destinations et filtres de parcours](#).

Mettre à jour la configuration de l'application AWS IoT SiteWise Edge

Quelques éléments doivent être pris en compte lors de la mise à jour de la configuration d'une application AWS IoT SiteWise Edge sur Siemens Industrial Edge.

Note

Toute modification de la configuration de l'application AWS IoT SiteWise Edge nécessite un redémarrage de l'application.

Raisons justifiant le redémarrage de l'application AWS IoT SiteWise Edge

- Une nouveauté Siemens Databus user pour l'application AWS IoT SiteWise Edge.
- Modification du fichier de configuration de la passerelle (votre fichier SiteWise_Edge_Gateway_Config).
- Une mise à jour de la configuration du proxy (qui nécessite également un redémarrage complet de l'IEVD)
- Pour activer les journaux de débogage pour les problèmes de débogage

Redémarrage de l'application

1. Dans votre Siemens Industrial Edge Management instance, choisissez Edge Management dans la section Applications de plateforme.
2. Choisissez Mes applications installées.
3. Sélectionnez l'application AWS IoT SiteWise Edge.
4. Choisissez Redémarrer.

Destinations et filtres de parcours

Les destinations dans AWS IoT SiteWise Edge constituent un moyen flexible et efficace de gérer le flux de vos données industrielles entre les appareils Edge et le cloud. Cette section explique

comment configurer les destinations, utiliser des filtres de chemin pour acheminer des flux de données spécifiques et choisir le type de destination adapté à votre cas d'utilisation.

Vous pouvez utiliser des destinations et des filtres de chemin sur des passerelles V3 auto-hébergées compatibles avec MQTT et des passerelles utilisées conjointement avec l'application Edge hébergée sur AWS IoT SiteWise Siemens Industrial Edge. Les destinations et les filtres de chemin ne fonctionnent pas avec les passerelles Classic Streams et V2.

Rubriques

- [Comprendre les destinations AWS IoT SiteWise Edge](#)
- [Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge](#)
- [Ajouter une destination AWS IoT SiteWise Edge en temps réel](#)
- [Ajouter une destination AWS IoT SiteWise mise en mémoire tampon à l'aide d'Amazon S3](#)
- [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#)
- [Gérer les destinations AWS IoT SiteWise Edge](#)

Comprendre les destinations AWS IoT SiteWise Edge

Utilisez les destinations AWS IoT SiteWise Edge pour déterminer où envoyer vos données sources. Vous pouvez choisir la destination de vos données en fonction des caractéristiques spécifiques dont vous avez besoin, telles que la rentabilité, la faible latence ou les exigences de stockage. Intégrez les données des appareils capturées par AWS IoT SiteWise nos partenaires ou des applications personnalisées pour publier et vous abonner à des filtres de chemin (rubriques) en périphérie. Vous pouvez ensuite modéliser, transférer et stocker les données de votre appareil dans le cloud.

Note

Pour utiliser pleinement toutes les fonctionnalités de destination sur les passerelles auto-hébergées, passez aux dernières versions de l' SiteWise éditeur IoT et du collecteur IoT SiteWise OPC UA. La prise en charge des flux est maintenue sur les flux classiques et les passerelles V2 afin de maintenir la compatibilité avec les configurations existantes. Pour de plus amples informations, consultez [Streams classiques, passerelles V2 pour Edge AWS IoT SiteWise](#).

Rubriques

- [Comment les destinations SiteWise Edge améliorent la gestion des données](#)
- [Types de destination](#)
- [Comparez les fonctionnalités de destination entre les versions de passerelle](#)
- [Limitations relatives aux destinations](#)
- [Cas d'utilisation pour les destinations SiteWise Edge](#)

Comment les destinations SiteWise Edge améliorent la gestion des données

Exportez des données de la périphérie vers des données AWS IoT SiteWise en temps réel, ou par lots à l'aide d'Amazon S3.

Les destinations améliorent la flexibilité et l'évolutivité de votre AWS IoT SiteWise environnement. Les destinations mettent en œuvre un modèle de gestion des données centralisé, dans lequel les sources publient les données dans un système central. Les destinations déterminent l'endroit où vos données sont envoyées à l'aide de filtres de chemin. Les destinations peuvent s'abonner à plusieurs filtres de chemin.

Les passerelles compatibles MQTT, qu'elles soient auto-hébergées ou exécutées Siemens Industrial Edge, utilisent MQTT pour les communications locales et sont fournies avec une destination en temps réel par défaut dont les filtres sont définis sur `#`. Cela signifie que, par défaut, tous les messages relatifs à tous les sujets sont publiés sur la destination AWS IoT SiteWise en temps réel. Pour de plus amples informations, veuillez consulter [Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge](#). Vous pouvez ajouter une destination en temps réel dans chaque passerelle.

Types de destination

Lorsque vous configurez une destination pour votre passerelle, vous disposez de deux options principales : une configuration en temps réel à l'aide d'AWS IoT SiteWise Amazon S3 et une configuration en mémoire tampon à l'aide d'Amazon S3. Chaque type de destination possède son propre ensemble de paramètres et de considérations.

AWS IoT SiteWise paramètres en temps réel

Choisissez cette option pour envoyer les données directement vers le stockage à AWS IoT SiteWise haut niveau afin de faciliter l'ingestion et le suivi des données en temps réel. Les paramètres en temps réel gèrent le flux de données, en particulier lorsqu'une passerelle rencontre

des problèmes de connectivité avec le cloud. En cas de perte de connexion, les données sont temporairement stockées localement sur la passerelle. Une fois la connexion rétablie, les données stockées sont automatiquement envoyées vers le cloud.

Vous pouvez ajuster différents aspects du processus de publication des données, tels que la quantité maximale de données à stocker localement, le taux d'envoi des données vers le cloud lors de la reconnexion et le moment où il convient de supprimer les données une fois que le stockage a atteint sa capacité maximale.

Pour plus d'informations sur les niveaux AWS IoT SiteWise de stockage, consultez [Gérez le stockage des données dans AWS IoT SiteWise](#).

AWS IoT SiteWise mise en mémoire tampon à l'aide des paramètres Amazon S3

Ce type de destination vous permet de mettre en mémoire tampon les données localement sur la passerelle et de les envoyer régulièrement par lots vers un compartiment Amazon S3. Les données sont stockées dans le format Parquet efficace, optimisé pour les charges de travail analytiques. Une fois les données enregistrées dans Amazon S3, vous pouvez les importer AWS IoT SiteWise pour les stocker, les traiter et les analyser.

Choisissez cette option pour ingérer les données par lots et stocker les données historiques de manière rentable. Vous pouvez configurer l'emplacement de votre compartiment Amazon S3 préféré et la fréquence à laquelle vous souhaitez que les données soient chargées sur Amazon S3. Vous pouvez également choisir ce que vous souhaitez faire avec les données après leur ingestion dans AWS IoT SiteWise. Vous pouvez choisir que les données soient disponibles à la fois dans Amazon S3 SiteWise et dans Amazon S3 ou vous pouvez choisir de les supprimer automatiquement d'Amazon S3.

Comparez les fonctionnalités de destination entre les versions de passerelle

La fonctionnalité de destination des passerelles compatibles MQTT rationalise la gestion des flux de données. Les destinations simplifient la gestion des données grâce à la configuration centralisée du routage des données vers différents points de terminaison. Cette approche élimine le besoin de configurations de flux individuelles complexes, ce qui rend l'ensemble du système plus flexible et plus facile à gérer.

À titre de comparaison, les flux classiques, la passerelle V2 et SiteWise Edge transmettent les données des sources de données aux éditeurs via des AWS IoT Greengrass flux, en configurant les destinations de données individuellement pour chaque source de données.

Avec la fonctionnalité de AWS IoT SiteWise destination, la configuration de routage de l'éditeur est consolidée. La configuration des destinations vous permet de gérer les destinations et les filtres de chemin de manière centralisée. Vous pouvez facilement ajouter une destination, gérer les filtres de chemin, supprimer les filtres ou les destinations inutiles, en fonction de vos besoins.

De plus, la fonctionnalité de destination utilise le protocole MQTT (Message Queuing Telemetry Transport), un protocole standard largement utilisé dans les applications IoT industrielles. Cette adoption de MQTT permet AWS IoT SiteWise de faciliter l'intégration avec divers appareils et systèmes.

Limitations relatives aux destinations

Les limites actuelles pour les destinations sur les passerelles SiteWise Edge sont les suivantes :

- Le pack de traitement de données n'est pas pris en charge sur les passerelles compatibles MQTT.
- La prise en charge des types de données est limitée aux types de AWS IoT SiteWise données. Pour plus d'informations sur l'activation de la conversion des types de données, consultez [Conversion de types de données non pris en charge](#).

Cas d'utilisation pour les destinations SiteWise Edge

SiteWise Les destinations périphériques sont utilisées dans diverses applications. Voici quelques exemples clés :

Automatisation industrielle, Surveillance en temps réel et maintenance prédictive

Dans les environnements industriels, les capteurs et les appareils de l'usine peuvent publier des données sur SiteWise Edge. Les destinations peuvent être configurées pour filtrer et acheminer les données pertinentes, ce qui permet de surveiller et d'analyser en temps réel les performances des machines. Vous pouvez vous abonner à des rubriques MQTT pertinentes à l'aide de filtres de chemin, traiter les données, puis publier les données traitées. De cette façon, vous pouvez acheminer de manière sélective les données traitées vers des services d'analyse AWS dans le cloud ou des systèmes sur site. Les fabricants peuvent ensuite mettre en œuvre des stratégies de maintenance prédictive, optimiser les processus de production et réduire les temps d'arrêt.

Bâtiments intelligents, Efficacité énergétique et optimisation de l'occupation

Les systèmes d'automatisation des bâtiments génèrent des flux de données pour surveiller et contrôler divers aspects d'un bâtiment, tels que les systèmes CVC, l'éclairage et le contrôle

d'accès. Avec SiteWise Edge, ces flux de données peuvent être ingérés, traités et acheminés vers différentes destinations. Les responsables des installations peuvent configurer les destinations pour filtrer et transmettre les données pertinentes, ce qui permet des fonctionnalités avancées telles que les mesures d'efficacité énergétique et l'optimisation du taux d'occupation, tout en garantissant la confidentialité et la conformité des données.

Ces cas d'utilisation montrent comment la fonctionnalité Destinations d' SiteWise Edge peut être exploitée dans différents secteurs pour ingérer, traiter et acheminer les données de manière efficace. Cela permet des fonctionnalités avancées telles que la surveillance en temps réel, la maintenance prédictive, l'efficacité énergétique et le diagnostic à distance, tout en garantissant la confidentialité et la conformité des données.

Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge

Rubriques

- [Exigences relatives au filtre de chemin](#)
- [Bonnes pratiques pour les filtres de chemins](#)
- [Filtres de chemin pour serveurs OPC UA](#)
- [Caractères spéciaux dans les noms des filtres de chemin](#)

Chaque destination est configurée pour acheminer les données vers Amazon S3 AWS IoT SiteWise ou Amazon S3. Les filtres de chemin vous permettent de sélectionner des données spécifiques à filtrer lorsque vous recevez des messages MQTT pour une destination. Les filtres de chemin représentent les noms logiques de vos flux de données, agissant comme des abonnements aux sujets MQTT souhaités.

Dans MQTT, les données sont organisées en rubriques, qui sont des chaînes hiérarchiques séparées par des barres obliques (/). Par exemple, un appareil peut publier des données de température dans le sujet `home/livingroom/sensor1/temperature`. Ici, `home/livingroom/sensor1` représente le chemin ou le nom logique du capteur, ainsi `temperature` que le type de données publié.

Vous pouvez utiliser des filtres de chemin pour vous abonner à des sujets spécifiques ou à une gamme de sujets à l'aide de caractères génériques (+et#). Le + joker correspond à un seul niveau de la hiérarchie des rubriques. Par exemple, `home/+sensor1/temperature` correspondrait à `home/`

livingroom/sensor1/temperature ethome/bedroom/sensor1/temperature. Le # joker, lorsqu'il est utilisé à la fin d'un filtre, correspond à plusieurs niveaux.

Vous pouvez également utiliser divers caractères généralement interdits dans la spécification MQTT dans le nom d'un filtre de chemin. Ces caractères ne fonctionnent pas comme des caractères génériques lorsqu'ils sont utilisés dans un nom. AWS IoT SiteWise convertit ces caractères en utilisant le codage pour garantir la conformité MQTT tout en préservant votre structure de dénomination d'origine. Cette fonctionnalité est particulièrement utile pour s'adapter aux conventions de dénomination existantes d'autres systèmes. Pour de plus amples informations, veuillez consulter [Caractères spéciaux dans les noms des filtres de chemin](#).

En sélectionnant soigneusement les filtres de chemin appropriés, vous pouvez contrôler les données envoyées vers une destination spécifique. Adaptez le flux de données aux exigences de votre système IoT à l'aide de filtres de chemin.

Exigences relatives au filtre de chemin

Lorsque vous entrez des filtres de trajectoire à l'aide du Console AWS IoT SiteWise, gardez à l'esprit les points suivants :

- Les filtres de chemin sont délimités par une nouvelle ligne, chaque ligne représentant un filtre de chemin distinct.
- Les filtres de chemin individuels peuvent avoir une taille comprise entre 1 et 65 535 octets.
- Un filtre de chemin ne peut pas être vide.
- Les valeurs nulles (U+0000) ne sont pas autorisées.
- Vous pouvez saisir jusqu'à 100 filtres de chemin ou 65 535 caractères à la fois, selon la première limite atteinte.
- La limite globale est de 20 000 filtres de chemin pour toutes les destinations d'une passerelle combinées.
- Vous pouvez utiliser des \$ caractères %#,+, et dans les noms des filtres de chemin, mais ils AWS IoT SiteWise sont automatiquement convertis en code URI.

Bonnes pratiques pour les filtres de chemins

Lorsque vous créez des filtres de chemin pour vos AWS IoT SiteWise destinations, considérez les stratégies suivantes pour gérer efficacement vos données.

- Structurez vos filtres pour refléter la hiérarchie de vos appareils. Par exemple, dans un environnement de fabrication `factory/+/machine/#`, capture les données de toutes les machines des différentes lignes de production.
- Utilisez des niveaux spécifiques pour les types, les emplacements ou les fonctions des appareils. Par exemple, `factory/assembly-line/robot/temperature`. Ou, dans le cadre de l'agriculture intelligente `farm/+/crop/+/moisture`, pour surveiller les niveaux d'humidité de différentes cultures dans différents champs.
- Utilisez les jokers de manière stratégique : + utilisez-les pour les variations d'un seul niveau et # pour capturer tous les niveaux suivants. Par exemple `building/+/+/energy-consumption`, suit la consommation d'énergie dans les différentes zones et étages d'un bâtiment. Cela suppose que le premier + capture tous les étages et le second + toutes les zones.
- Équilibrez spécificité et flexibilité en créant des filtres suffisamment spécifiques pour capturer des données pertinentes, mais suffisamment flexibles pour s'adapter aux modifications futures. `site/+/equipment-type/+/measurement` Permet par exemple d'ajouter de nouveaux sites ou types d'équipements sans modifier la structure du filtre.

Testez soigneusement vos filtres pour vous assurer qu'ils capturent les données souhaitées et qu'ils correspondent à l'architecture et aux objectifs de votre système IoT.

Filtres de chemin pour serveurs OPC UA

Pour les serveurs OPC UA, vos filtres de chemin doivent correspondre aux noms des balises OPC UA. Le niveau final de votre filtre de chemin doit correspondre exactement au nom de la balise OPC UA. Par exemple, si votre balise OPC UA l'est `Device1.Temperature`, votre filtre de chemin pourrait l'être `factory/line1/Device1.Temperature`. Vous pouvez utiliser des caractères génériques dans les niveaux précédents, par exemple `factory/+/Device1.Temperature` pour capturer le tag sur plusieurs lignes de production. Si les noms de vos filtres de chemin contiennent des caractères spéciaux, consultez [Caractères spéciaux dans les noms des filtres de chemin](#) pour plus d'informations.

Caractères spéciaux dans les noms des filtres de chemin

AWS IoT SiteWise prend en compte les caractères couramment utilisés dans les protocoles industriels tels que OPC UA, qui ne sont généralement pas autorisés dans les noms de sujets MQTT standard. Cette fonctionnalité facilite l'intégration des systèmes industriels avec les architectures basées sur MQTT.

 Note

Bien que notre gestion des caractères spéciaux soit utile pour l'intégration et la migration, il est recommandé de s'aligner sur les conventions de dénomination MQTT standard pour les nouvelles implémentations dans la mesure du possible afin de garantir une plus grande compatibilité.

Lorsque vous recevez des données provenant de sources industrielles, AWS IoT SiteWise normalise les noms des sujets en utilisant le codage URI pour les caractères spéciaux :

- %devient %25 (encodé d'abord en tant que caractère d'échappement)
- #devient %23
- +devient %2B
- \$devient %24 (uniquement au début d'un sujet)

Ce codage garantit que les données sources contenant ces caractères MQTT spéciaux peuvent être utilisées en toute sécurité comme noms de sujets MQTT tout en préservant les conventions de dénomination industrielle d'origine.

Exemple : caractères spéciaux dans les noms des filtres de chemin

Voici des exemples de la façon dont les noms de sujets industriels peuvent apparaître dans les filtres de AWS IoT SiteWise chemin :

- Factory1/Line#2/Sensor+3devient Factory1/Line%232/Sensor%2B3
- Plant%A/Unit\$1/Tempdevient Plant%25A/Unit%241/Temp
- Site1/#Section/+Nodedevient Site1/%23Section/%2BNode

Lorsque vous créez des abonnements ou que vous consultez les noms des rubriques AWS IoT SiteWise, vous verrez les versions originales non codées. L'encodage est géré automatiquement pour garantir la conformité au MQTT.

Ajouter une destination AWS IoT SiteWise Edge en temps réel

Le type de destination en temps réel vous permet de diffuser des données IoT directement depuis vos appareils et vos passerelles vers le AWS IoT SiteWise stockage en temps réel. Cette option est

idéale pour les cas d'utilisation qui nécessitent l'ingestion et le traitement immédiats des données dès leur génération, sans nécessiter de traitement par lots ou de mise en mémoire tampon. Vous ne pouvez configurer qu'une seule destination en temps réel dans chaque passerelle, car elle diffuse des données en continu vers AWS IoT SiteWise.

 Note

La duplication TQVs peut entraîner une double charge.

Pour ajouter une destination en temps réel

Utilisez la AWS IoT SiteWise console ou AWS CLI pour ajouter une destination en temps réel à votre passerelle SiteWise Edge compatible MQTT.

Console

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle à laquelle vous souhaitez ajouter une destination.
4. Dans la section Destinations, choisissez Ajouter une destination.
5. Sur la page Ajouter une destination, entrez les détails de la destination :
 - a. Le nom de votre destination dans le champ Nom de la destination.
 - b. Sélectionnez le AWS IoT SiteWise temps réel pour le type de destination.
6. Configurez l'ordre de publication de la passerelle en définissant l'ordre de publication sur Publier les anciennes données en premier ou Publier les données les plus récentes en premier. Par défaut, la passerelle publie les données les plus anciennes en premier.
7. Utilisez le délai d'attente maximal par lots pour définir le temps d'attente maximal que l'éditeur doit attendre avant d'envoyer un lot de données à AWS IoT SiteWise. Ce paramètre s'applique à chaque alias. Les données sont stockées localement jusqu'à ce que :
 - Le temps défini est écoulé, ou
 - 10 entrées time-quality-value (TQV) sont reçues pour l'alias

Quelle que soit la condition remplie, cela déclenche d'abord l'envoi du lot vers le cloud.

8. Pour compresser les données téléchargées, cochez la case Activer la compression lors du téléchargement des données. Le fait de laisser la passerelle compresser vos données avant de les télécharger dans le cloud réduit l'utilisation de la bande passante.
9. Pour filtrer les données d'éditeur expirées, cochez la case Exclure les données expirées. Cette sélection envoie uniquement les données actives et actuelles à AWS IoT SiteWise.
10. Dans le champ Période limite, entrez la fréquence à laquelle les données doivent être considérées comme expirées dans votre ensemble de données. Vous pouvez déterminer si les données sont comptées en minutes ou en jours. La période de coupure minimale est de cinq minutes. La période limite maximale est de sept jours.

11. Configurez éventuellement les paramètres de stockage local :

- a. Définissez la fréquence de la période de conservation : durée pendant laquelle la passerelle stocke localement des données antérieures à la période limite. La durée de conservation minimale est d'une minute.

La période de conservation maximale est de 30 jours et est supérieure ou égale à la période de rotation.

- b. Définir la période de rotation : intervalle de temps à spécifier lors de l'enregistrement de données antérieures à la période limite pour un seul fichier. La passerelle transfère un lot de données vers le répertoire local suivant à la fin de chaque période de rotation : `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`.

La durée de conservation doit être supérieure à une minute et égale à la durée de conservation.

- c. Indiquez la valeur de capacité de stockage (Go) pour définir la taille maximale des données stockées localement en Go. Si les données dépassent la taille maximale de stockage local déterminée, la passerelle commence par supprimer les données les plus anciennes. La passerelle continue à effectuer des suppressions jusqu'à ce que la taille des données stockées localement soit égale ou inférieure au quota.

La capacité de stockage doit être supérieure ou égale à un Go.

12. Ajoutez des filtres de chemin à votre destination. Pour plus d'informations, voir, [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#).

Pour de plus amples informations, veuillez consulter [Types de destination](#).

AWS CLI

Exemple : créer une nouvelle destination AWS IoT SiteWise en temps réel

Utilisez l'[UpdateGatewayCapabilityConfiguration](#) API pour configurer l'éditeur.

Définissez le paramètre `capabilityNamespace` sur `iotsitewise:publish:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "maxBatchWaitTime": "10s"
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```

Pour mettre à jour une destination AWS IoT SiteWise en temps réel existante, utilisez d'abord l'`DescribeGatewayCapabilityConfiguration` API pour trouver le `destinationId`.

Exemple : Mettre à jour une destination AWS IoT SiteWise en temps réel

Utilisez l'[UpdateGatewayCapabilityConfiguration](#) API pour configurer l'éditeur.

Définissez le paramètre `capabilityNamespace` sur `iotsitewise:publish:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_REALTIME",
      "name": "your-destination-name",
      "config": {
        "publishingOrder": "TIME_ORDER",
        "enableCompression": true,
        "dropPolicy": {
          "cutoffAge": "7d",
          "exportPolicy": {
            "retentionPeriod": "7d",
            "rotationPeriod": "6h",
            "exportSizeLimitGB": 10
          }
        },
        "maxBatchWaitTime": "10s"
      },
      "filters": [
        {
          "type": "PATH",
          "config": {
            "paths": [
              "#"
            ]
          }
        }
      ]
    }
  ]
}
```

Les options de configuration suivantes sont spécifiques aux passerelles utilisant l'espace de `iotsitewise:publish:3` noms.

sources

Définit les sources de données vers lesquelles transférer les données de votre équipement industriel AWS IoT SiteWise. Pour les passerelles compatibles MQTT, utilisez. MQTT

Type : tableau d'objets

Obligatoire : oui

destinations

Définit où envoyer les données. Les destinations sont soit en temps réel, soit mises en mémoire tampon à l'aide d'Amazon S3. Au moins un objet de destination est requis, mais vous pouvez ajouter un tableau vide. Vous pouvez avoir une destination en temps réel pour chaque passerelle. Pour de plus amples informations, veuillez consulter [Comprendre les destinations AWS IoT SiteWise Edge](#).

Type : tableau d'objets

Obligatoire : oui

id

Identifiant unique de la destination. Vous pouvez soit fournir un identifiant de destination existant, soit le laisser vide. Si vous ne spécifiez aucun identifiant, un UUID est généré par défaut.

Type : chaîne

Obligatoire : non

type

Type de destination. Les options incluent : SITEWISE_REALTIME et SITEWISE_BUFFERED.

- SITEWISE_REALTIME— Envoyez les données directement au AWS IoT SiteWise stockage en temps réel.
- SITEWISE_BUFFERED— Envoyez des données à Amazon S3 par lots au format Parquet, puis importez-les dans le AWS IoT SiteWise stockage.

Type : String

Obligatoire : oui

name

Un nom unique pour la destination.

Type : String

Obligatoire : oui

config

Configuration spécifique au type de destination au format JSON. La configuration varie entre les destinations en temps réel et les destinations mises en mémoire tampon.

Type : objet

Obligatoire : oui

Ordre de publication

Détermine l'ordre dans lequel les données sont publiées. Les données sont publiées en fonction de leur horodatage. Les options incluent `TIME_ORDER` et `RECENT_DATA`.

- `TIME_ORDER`(par défaut) — Publie d'abord les anciennes données.
- `RECENT_DATA`— Publie d'abord les données les plus récentes.

Type : chaîne

Obligatoire : non

Activer la compression

Lorsque ce paramètre est défini sur `true`, active la compression des données avant de les envoyer à AWS IoT SiteWise. Le fait de laisser la passerelle compresser vos données avant de les télécharger dans le cloud réduit l'utilisation de la bande passante. La valeur par défaut est `true`.

Type : booléen

Obligatoire : non

Politique de dépôt

Définit la manière de gérer les anciennes données.

Type : objet

Obligatoire : non

- `cutoffAge`

L'âge maximal des données à publier est indiqué en jours, heures et minutes. Par exemple, 7d ou 1d7h16m. Les données plus anciennes que celles que vous spécifiez ne sont pas envoyées à AWS IoT SiteWise.

Les données antérieures à la période limite ne sont pas publiées dans le cloud. L'âge limite doit être compris entre cinq minutes et sept jours.

Vous pouvez utiliser mh, et d lorsque vous spécifiez un âge limite. Notez que cela m h représente les minutes, les heures et les d jours.

Type : String

Obligatoire : oui

- `exportPolicy`

Définit la manière de traiter les données qui dépassent l'âge limite.

Type : objet

Obligatoire : non

- `retentionPeriod`

Votre passerelle SiteWise Edge supprime toutes les données de la périphérie antérieures à la période limite du stockage local une fois qu'elles ont été stockées pendant la période de conservation spécifiée. La période de conservation doit être comprise entre une minute et 30 jours, et être supérieure ou égale à la période de rotation.

Vous pouvez utiliser mh, et d lorsque vous spécifiez une période de conservation. Notez que cela m h représente les minutes, les heures et les d jours.

Type : chaîne

Obligatoire : non

- `rotationPeriod`

Intervalle de temps pendant lequel les données antérieures à la période limite doivent être regroupées et enregistrées dans un seul fichier. La passerelle SiteWise Edge transfère un lot de données vers le répertoire local suivant à la fin de chaque période de rotation : `/greengrass/v2/work/aws.iot.SiteWiseEdgePublisher/exports`. La période de rotation doit être supérieure à une minute et égale ou inférieure à la période de conservation.

Vous pouvez utiliser `mh`, et `d` lorsque vous spécifiez une période de rotation. Notez que cela `m h` représente les minutes, les heures et les `d` jours.

Type : chaîne

Obligatoire : non

- `exportSizeLimitGB`

Taille maximale autorisée des données stockées localement, en Go. Si ce quota est dépassé, la passerelle SiteWise Edge commence à supprimer les données les plus anciennes jusqu'à ce que la taille des données stockées localement soit égale ou inférieure au quota. La valeur de ce paramètre doit être supérieure ou égale à 1.

Type : entier

Obligatoire : non

`maxBatchWaitTime`

Définit le délai maximal pendant lequel l'éditeur doit attendre avant d'envoyer un lot de données à AWS IoT SiteWise. Ce paramètre s'applique à chaque alias. Les données sont stockées localement jusqu'à ce que :

- Le temps défini est écoulé, ou
- 10 entrées time-quality-value (TQV) sont reçues pour l'alias

Utilisez `mh`, et `d` pour spécifier une heure limite. Notez que cela `m h` représente les minutes, les heures et les `d` jours.

Type : chaîne

Obligatoire : non

filters

Filtres à appliquer aux données. Au moins un filtre est requis.

Type : String

Obligatoire : oui

type

Type de filtre Utilisez PATH.

Type : String

Obligatoire : oui

config

Configuration spécifique au type de filtre au format JSON. Au moins un objet est requis, mais le tableau peut être vide.

Type : objet

Obligatoire : oui

- paths

Un ensemble de filtres de chemin. Pour de plus amples informations, veuillez consulter [Comprendre les filtres de chemin pour les destinations AWS IoT SiteWise Edge](#). Le chemin par défaut est #.

Type : tableau de chaînes

Obligatoire : oui

Ajouter une destination AWS IoT SiteWise mise en mémoire tampon à l'aide d'Amazon S3

Le type de destination mise en mémoire tampon vous permet de réduire les coûts d'ingestion AWS IoT SiteWise si vous n'avez pas besoin des données en temps réel. Il vous permet de stocker temporairement vos données IoT dans un compartiment Amazon S3 avant de les importer dans AWS IoT SiteWise. Vous pouvez également simplement télécharger vos données dans S3 pour les

stocker, que vous envisagiez de les importer ou non AWS IoT SiteWise. Cela est utile pour regrouper et mettre en mémoire tampon les données de vos appareils et de vos passerelles avant de les ingérer. AWS IoT SiteWise Avec cette option, les données sont téléchargées dans le compartiment S3 spécifié au format Parquet à une fréquence configurée. Vous pouvez ensuite importer ces données dans le AWS IoT SiteWise stockage pour une analyse et un traitement plus approfondis.

Pour ajouter une destination mise en mémoire tampon à l'aide d'Amazon S3

Utilisez la AWS IoT SiteWise console ou AWS CLI pour ajouter une destination qui met en mémoire tampon les données à l'aide d'Amazon S3 sur votre passerelle SiteWise Edge.

Console

Utilisez le AWS Management Console pour ajouter une AWS IoT SiteWise destination mise en mémoire tampon à l'aide d'Amazon S3.

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle à laquelle vous souhaitez ajouter une destination.
4. Dans la section Destinations, choisissez Ajouter une destination.
5. Sur la page Ajouter une destination, entrez les détails de la destination :
 - a. Le nom de votre destination dans le champ Nom de la destination.
 - b. Sélectionnez le AWS IoT SiteWise type de destination mis en mémoire tampon à l'aide d'Amazon S3. AWS IoT SiteWise mis en mémoire tampon à l'aide d'Amazon S3 envoie des données à Amazon Simple Storage Service par lots, au format Parquet, puis les importe dans AWS IoT SiteWise le stockage.
6. Entrez l'URL Amazon S3 correspondant à l'emplacement où vous souhaitez stocker les données de votre passerelle. Vous pouvez rechercher le chemin en choisissant Parcourir S3. Une fois qu'un bucket est ajouté, vous pouvez également le consulter en choisissant View.
7. Spécifiez la fréquence à laquelle votre passerelle doit télécharger des données vers Amazon S3 en saisissant un intervalle de temps et en sélectionnant un intervalle de temps pour la fréquence de téléchargement des données. La valeur de fréquence doit être supérieure à 0 et inférieure ou égale à 30 jours.
8. Dans Paramètres de stockage des données, déterminez ce que vous devez faire avec les données de votre passerelle après les avoir importées dans AWS IoT SiteWise. Il y a deux décisions à prendre concernant le stockage des données :

- Si vous souhaitez copier les données importées dans le AWS IoT SiteWise stockage, cochez la case Copier les données dans le stockage. Cette option duplique les données importées depuis votre compartiment Amazon S3 configuré vers AWS IoT SiteWise le stockage.
 - Si vous choisissez d'importer vos données depuis votre compartiment Amazon S3 vers le AWS IoT SiteWise stockage, vous pouvez également spécifier si les données importées doivent être supprimées une fois l'importation terminée. Cochez la case Supprimer les données d'Amazon S3 pour supprimer la date importée du compartiment Amazon S3 configuré après l'avoir importée dans le AWS IoT SiteWise stockage.
9. Ajoutez des filtres de chemin à votre destination. Pour plus d'informations, voir, [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#).

AWS CLI

Exemple : Création d'une nouvelle AWS IoT SiteWise destination mise en mémoire tampon à l'aide d'Amazon S3

Utilisez l'[UpdateGatewayCapabilityConfiguration](#) API pour configurer l'éditeur.

Définissez le paramètre `capabilityNamespace` sur `iotsitewise:publisher:3`.

```
{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "type": "SITEWISE_BUFFERED",
      "name": "your-s3-destination-name",
      "config": {
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
        "publishPolicy": {
          "publishFrequency": "15m",
          "localSizeLimitGB": 10
        },
        "siteWiseImportPolicy": {
          "enableSiteWiseStorageImport": true,

```

```

        "enableDeleteAfterImport": true,
        "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
    },
    "filters": [
        {
            "type": "PATH",
            "config": {
                "paths": [
                    "#"
                ]
            }
        }
    ]
}

```

Exemple : Mettre à jour une AWS IoT SiteWise destination mise en mémoire tampon à l'aide d'Amazon S3

Pour mettre à jour une destination AWS IoT SiteWise en temps réel existante, utilisez d'abord l'`DescribeGatewayCapabilityConfigurationAPI` pour trouver le `destinationId`.

L'espace de noms de l'éditeur : `iotsitewise:publisher:3`

```

{
  "sources": [
    {
      "type": "MQTT"
    }
  ],
  "destinations": [
    {
      "id": "your-existing-destination-id",
      "type": "SITEWISE_BUFFERED",
      "name": "your-s3-destination-name",
      "config": {
        "targetBucketArn": "arn:aws:s3:::amzn-s3-demo-bucket/Optional/SomeFolder",
        "publishPolicy": {
          "publishFrequency": "15m",
          "localSizeLimitGB": 10
        }
      }
    }
  ]
}

```

```

    },
    "siteWiseImportPolicy": {
      "enableSiteWiseStorageImport": true,
      "enableDeleteAfterImport": true,
      "bulkImportJobRoleArn": "arn:aws:iam::123456789012:role/your-role-name"
    }
  },
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "#"
        ]
      }
    }
  ]
}
]
}

```

Les options de configuration suivantes sont spécifiques aux passerelles compatibles MQTT utilisant l'espace de noms. `iotsitewise:publisher:3`

sources

Définit les sources de données vers lesquelles transférer les données de votre équipement industriel AWS IoT SiteWise. Pour les passerelles compatibles MQTT, utilisez. MQTT

Type : tableau d'objets

Obligatoire : oui

destinations

Définit où envoyer les données. Les destinations sont soit en temps réel, soit mises en mémoire tampon à l'aide d'Amazon S3. Au moins un objet de destination est requis, mais vous pouvez ajouter un tableau vide. Vous pouvez avoir une destination en temps réel pour chaque passerelle. Pour de plus amples informations, veuillez consulter [Comprendre les destinations AWS IoT SiteWise Edge](#).

Type : tableau d'objets

Obligatoire : oui

id

Identifiant unique de la destination. Vous pouvez soit fournir un identifiant de destination existant, soit le laisser vide pour qu'un nouvel identifiant soit automatiquement généré pour la destination.

Type : chaîne

Obligatoire : non

type

Type de destination. Les options incluent : SITEWISE_REALTIME et SITEWISE_BUFFERED. Sélectionnez SITEWISE_BUFFERED.

- SITEWISE_REALTIME(par défaut) : envoyez les données directement au AWS IoT SiteWise stockage en temps réel. Pour de plus amples informations, veuillez consulter [Ajouter une destination AWS IoT SiteWise Edge en temps réel](#).
- SITEWISE_BUFFERED— Envoyez des données à Amazon S3 par lots au format Parquet, puis importez-les dans le AWS IoT SiteWise stockage.

Type : String

Obligatoire : oui

name

Un nom unique pour la destination.

Type : String

Obligatoire : oui

config

Configuration spécifique au type de destination au format JSON. La configuration varie entre les destinations en temps réel et les destinations mises en mémoire tampon.

Type : objet

Obligatoire : oui

targetBucketArn

Le compartiment ARN dans lequel publier. Choisissez la même option Région AWS pour les deux AWS IoT SiteWise et pour Amazon S3. Si un préfixe est choisi, il doit comporter entre 1 et 255 caractères.

Note

AWS IoT SiteWise, y compris la passerelle, auront accès à l'intégralité du compartiment S3 spécifié. Nous vous recommandons d'utiliser un bucket dédié pour l'ingestion de données en mémoire tampon.

Type : String

Obligatoire : oui

publishPolicy

Détails de la politique de publication.

Type : objet

Obligatoire : oui

publishFrequency

Fréquence à laquelle la passerelle SiteWise Edge publie dans le compartiment Amazon S3. La fréquence de chargement des données vers Amazon S3 doit être supérieure à 0 minute et inférieure ou égale à 30 jours. Vous pouvez utiliser mh, et d quand vous spécifiez une fréquence de publication, un âge. Notez que cela m h représente les minutes, les heures et les d jours. La valeur par défaut est de 15 minutes.

Type : String

Obligatoire : oui

localSizeLimitGB

Taille maximale des fichiers écrits sur le disque local en Go. Si ce seuil est dépassé, l'éditeur publie toutes les données mises en mémoire tampon vers leur destination.

Type : entier

Obligatoire : oui

`siteWiseImportPolicy`

Détails de la politique d'importation pour l'importation de données vers AWS IoT SiteWise.

Type : objet

Obligatoire : oui

`enableSiteWiseStorageImport`

Définissez ce paramètre sur `true` pour importer les données d'un compartiment Amazon S3 vers le AWS IoT SiteWise stockage. Il effectue initialement une copie des données dans AWS IoT SiteWise. Ensuite, si vous définissez sur `enableDeleteAfterImport true`, les données de S3 sont supprimées après avoir été copiées vers AWS IoT SiteWise. Les implications tarifaires s'appliquent. La valeur par défaut est `true`.

Type : booléen

Obligatoire : oui

`enableDeleteAfterImport`

Définissez cette option `true` pour supprimer le fichier dans le compartiment Amazon S3 après son ingestion dans le AWS IoT SiteWise stockage. La valeur par défaut est `true`.

Type : booléen

Obligatoire : oui

`bulkImportJobRoleArn`

L'ARN du rôle IAM qui AWS IoT SiteWise suppose de lire les données mises en mémoire tampon depuis Amazon S3 lors de l'ingestion des données. Ce rôle est utilisé lorsqu'un appareil Edge appelle AWS IoT SiteWise APIs pour lancer le processus d'importation en bloc.

 Note

S'enableSiteWiseStorageImport est défini sur true, ce paramètre est obligatoire.

Type : chaîne

Obligatoire : non

Ajoutez des filtres de chemin pour votre destination. Pour de plus amples informations, veuillez consulter [Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge](#).

Ajouter des filtres de chemin aux destinations AWS IoT SiteWise Edge

Ajoutez des filtres de chemin à une destination. Les filtres de chemin utilisent la syntaxe des rubriques MQTT, où il # s'agit d'un caractère générique correspondant à un nombre quelconque de niveaux et + d'un caractère générique correspondant à un seul niveau. Vous pouvez ajouter plusieurs destinations à une passerelle, chacune disposant de son propre ensemble de filtres de chemin abonnés à la télémétrie de votre équipement.

Siemens Industrial Edge les passerelles utilisent un préfixe pour des raisons de compatibilité. Pour de plus amples informations, veuillez consulter [Préfixes pour les filtres de chemin](#).

Console

Pour ajouter des filtres de chemin

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle à laquelle vous souhaitez ajouter des filtres de chemin.
4. Dans la section Filtres de chemin sous Ajouter une destination, choisissez Ajouter un filtre de chemin.
5. Entrez le filtre de chemin auquel vous souhaitez que cette destination s'abonne. Vous pouvez utiliser des caractères génériques (#et+) pour vous abonner à plusieurs chemins.
6. Choisissez Ajouter un filtre de chemin pour ajouter le filtre de chemin à la liste.
7. Répétez les étapes pour ajouter des filtres de chemin supplémentaires, si nécessaire.

- Une fois que vous avez ajouté tous les filtres de chemin requis, choisissez **Create**.

AWS CLI for self-hosted gateways

Exemple : Configuration du filtre de chemin

```
{
  "destinations": [
    {
      ...
    }
  ],
  "filters": [
    {
      "type": "PATH",
      "config": {
        "paths": [
          "home/+/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "home/livingroom/sensor1/temperature",
          "building/#"
        ]
      }
    }
  ]
}
```

AWS CLI for Siemens IEGateways

Exemple : Configuration des préfixes pour les filtres de chemin

Capturez toutes les données en utilisant à la fois les préfixes data (ie/die/m) et metadata () pour chaque filtre de chemin.

```
{
  "destinations": [
    {
      ...
    }
  ]
}
```

```
],
"filters": [
  {
    "type": "PATH",
    "config": {
      "paths": [
        "ie/d/home/+sensor12/temperature",
        "ie/m/home/livingroom/sensor12/temperature",
        "ie/d/home/livingroom/sensor13/temperature2",
        "ie/m/home/livingroom/sensor13/temperature2",
        "ie/d/building/#",
        "ie/m/building/#"
      ]
    }
  }
]
```

Note

Copiez les filtres de chemin entre les destinations en téléchargeant la liste des filtres de chemin. Pour de plus amples informations, veuillez consulter [Télécharger tous les filtres de chemin d'une destination \(console\)](#).

Téléchargez des filtres de chemin en bloc

Pour télécharger des filtres de chemin en bloc, utilisez un fichier CSV ou texte. AWS IoT SiteWise supprime automatiquement les doublons exacts lorsque vous téléchargez des fichiers. Par exemple, `windfarm/site1/` et `windfarm/site1/` sont des doublons exacts qui sont AWS IoT SiteWise interceptés parce que la chaîne est exactement la même. Les doublons partiels ne sont pas supprimés et entraînent des frais supplémentaires. Par exemple, `windfarm/#` et `windfarm/site1` sont des sujets qui se chevauchent parce qu'ils `windfarm/site1` sont déjà couverts par `windfarm/#`.

Note

Évitez les doublons pour éviter des frais supplémentaires. Le fichier chargé doit être au format `.csv` ou `.txt`. Il ne peut pas contenir d'en-têtes et doit être composé d'une seule

colonne. Dans la colonne, listez vos filtres de chemin, chaque filtre étant sur une ligne distincte. Aucune autre information ne doit être incluse dans le fichier.

Exigences relatives au téléchargement de fichiers

Il s'agit d'exigences supplémentaires en matière de filtre de chemin.

- Vous pouvez télécharger un fichier .csv ou .txt. Les autres formats de fichiers ne sont pas pris en charge.
- Les fichiers CSV (.csv) ne peuvent pas comporter d'en-têtes et ne doivent contenir qu'une seule colonne.
- Vous pouvez avoir un filtre de chemin sur chaque ligne.
- Les fichiers téléchargés ne peuvent pas être vides.
- Lorsque vous l'utilisez # comme caractère générique, il doit s'agir du dernier caractère du filtre de rubrique. Par exemple, `topic/#` ou en tant que personnage autonome à un niveau de sujet particulier. Cependant, notez qu'il # peut également être utilisé comme caractère normal dans le nom d'un niveau de sujet, tel que `factory/machine#1/topic`. Pour plus d'informations, consultez [Caractères spéciaux dans les noms des filtres de chemin](#).
- Vous pouvez également utiliser le + personnage. Par exemple, utilisez-le `factory/+ /temp` pour obtenir toutes les températures pour les usines plutôt que `factory/machine3/temp` individuellement. `factory/machine2/temp`

Gérer les destinations AWS IoT SiteWise Edge

Après avoir ajouté des destinations, vous pouvez effectuer diverses opérations pour les gérer, telles que la modification des configurations de destination, la suppression de destinations et la gestion des filtres de chemin.

Modifier une destination

Sélectionnez le bouton radio à côté de la destination dans le tableau et cliquez sur le bouton Modifier pour modifier une destination.

Console

Pour modifier une destination à l'aide de la AWS IoT SiteWise console

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Sélectionnez la passerelle appropriée.
4. Dans la section Destinations, choisissez la destination que vous souhaitez modifier, puis sélectionnez Modifier.
5. Modifiez la destination, puis choisissez Enregistrer.

AWS CLI

Pour modifier une destination à l'aide de AWS CLI

- Vous pouvez modifier une destination en modifiant les informations de configuration de la fonctionnalité JSON.

```
aws iotsitewise update-gateway-capability-configuration \  
--gateway-id your-gateway-id \  
--capability-namespace "iotsitewise:publisher:3" \  
--capability-configuration '{  
  "sources": [  
    {  
      "type": "MQTT"  
    }  
  ],  
  "destinations": [  
    {  
      "id": "your-existing-destination-id",  
      "type": "SITEWISE_REALTIME",  
      "name": "your-updated-destination-name",  
      "config": {  
        "publishingOrder": "TIME_ORDER",  
        "enableCompression": true,  
        "dropPolicy": {  
          "cutoffAge": "10d",  
          "exportPolicy": {  
            "retentionPeriod": "10d",  
            "rotationPeriod": "6h",
```

```
        "exportSizeLimitGB": 10
      },
      "maxBatchWaitTime": "15s"
    },
    "filters": [
      {
        ...
      }
    ]
  }
]
```

Note

Vous ne pouvez pas mettre à jour la destination type `oucapability-namespace`. Par exemple, vous ne pouvez pas passer d'un type de `SITewise_REALTIME` à `SITewise_BUFFERED`. Vous pouvez avoir une destination en temps réel pour chaque passerelle compatible MQTT.

Supprimer une destination

Si vous n'avez plus besoin d'une destination, vous pouvez la supprimer de votre passerelle SiteWise Edge.

Console

Pour supprimer une destination à l'aide de la AWS IoT SiteWise console

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Sélectionnez la passerelle appropriée.
4. Dans la section Destinations, choisissez la destination que vous souhaitez supprimer, puis sélectionnez Supprimer. Un écran de confirmation apparaît.
5. Pour confirmer votre choix de supprimer la destination, tapez « supprimer » dans le champ de confirmation.

AWS CLI

Pour supprimer une destination à l'aide de AWS CLI

- Supprimez la configuration des fonctionnalités de la passerelle en spécifiant l'ID de la passerelle et en modifiant la configuration des fonctionnalités pour supprimer la destination que vous souhaitez supprimer.

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --capability-configuration '{
    "sources": [
      {
        "type": "MQTT"
      }
    ],
    "destinations": []
  }'
```

Note

Le tableau de destinations peut être vide ([]), mais l'objet de destinations lui-même doit être inclus dans la configuration des fonctionnalités.

Télécharger tous les filtres de chemin d'une destination (console)

Téléchargez un fichier CSV contenant tous vos filtres de chemin dans la AWS IoT SiteWise console. Vous pouvez utiliser une liste de filtres de chemin téléchargée pour partager facilement des listes de filtres de chemin entre les destinations de passerelle.

Pour télécharger un fichier CSV contenant tous les filtres de chemin à l'aide de la AWS IoT SiteWise console

- Ouvrez la [AWS IoT SiteWise console](#).
- Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
- Sélectionnez la passerelle contenant vos filtres de chemin.
- Choisissez Ajouter une destination ou Modifier la destination.

5. Accédez à la section Filtres de chemin et choisissez Télécharger le fichier CSV.

Note

Le fichier CSV inclut tous les filtres de chemin d'une destination donnée, quels que soient ceux que vous avez sélectionnés dans la liste des filtres de chemin.

Modifier un filtre de chemin

Vous pouvez modifier les filtres de chemin individuels pour affiner les données que reçoit votre destination.

Console

À l'aide de la AWS IoT SiteWise console, vous pouvez modifier chaque filtre de chemin individuel dans chaque zone de texte correspondante.

Pour modifier un filtre de chemin à l'aide de la AWS IoT SiteWise console

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Sélectionnez la passerelle contenant vos filtres de chemin.
4. Sélectionnez la destination appropriée.
5. Choisissez Modifier.
6. Choisissez la zone de texte correspondant à la ligne contenant le filtre de chemin que vous souhaitez modifier.
7. Mettez à jour le texte du filtre de chemin en vous assurant que la case du filtre de chemin modifié est cochée.
8. Choisissez Enregistrer.

AWS CLI

Pour modifier les filtres de chemin d'une destination à l'aide du AWS CLI, récupérez d'abord la configuration actuelle, modifiez-la, puis mettez-la à jour à l'aide de la `update-gateway-capability-configuration` commande.

Pour modifier un filtre de chemin à l'aide de AWS CLI

1. Récupérez la configuration de capacité actuelle :

```
aws iotsitewise describe-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --query "capabilityConfiguration"
```

2. Modifiez le JSON pour modifier les filtres de chemin selon vos besoins.
3. Mettez à jour la configuration des fonctionnalités avec les filtres de chemin modifiés :

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id your-gateway-id \
  --capability-namespace "iotsitewise:publisher:3" \
  --capability-configuration json-containing-your-updated-path-filters
```

Supprimer un filtre de chemin

Vous pouvez supprimer les filtres de chemin d'une destination afin de contrôler les données qu'elle reçoit des sources MQTT et des pipelines de traitement de données.

Console

Pour supprimer un filtre de chemin à l'aide de la AWS IoT SiteWise console

1. Ouvrez la [AWS IoT SiteWise console](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateway dans la section Edge.
3. Sélectionnez la passerelle contenant vos filtres de chemin.
4. Sélectionnez la destination appropriée.
5. Choisissez Modifier.
6. Sur l'écran Modifier la destination, dans la section Filtres de chemin, sélectionnez un ou plusieurs filtres de chemin à supprimer.
7. Sélectionnez Delete (Supprimer). Un message de confirmation de suppression s'affiche. Si vous souhaitez procéder à la suppression des filtres de chemin, choisissez Supprimer sur l'écran de confirmation.

AWS CLI

Pour supprimer une destination à l'aide de AWS CLI

- Supprimez un filtre de chemin en le retirant de la configuration des fonctionnalités.

```
aws iotsitewise update-gateway-capability-configuration \  
  --gateway-id your-gateway-id \  
  --capability-namespace "iotsitewise:publisher:3" \  
  --capability-configuration '{  
    "sources": [  
      {  
        "type": "MQTT"  
      }  
    ],  
    "destinations": [  
      {  
        "id": "your-destination-id",  
        "type": "SITEWISE_REALTIME",  
        "name": "your-destination-name",  
        "config": {  
          ...  
        },  
        "filters": [  
          {  
            "type": "PATH",  
            "config": {  
              "paths": [  
                "/path1",  
                "/path2",  
                "/delete-a-path-to-remove-it"  
              ]  
            }  
          }  
        ]  
      }  
    ]  
  }  
}
```

 Note

Le tableau de filtres peut être vide ([]), mais l'objet des filtres lui-même doit être inclus dans la configuration des fonctionnalités.

Gérer les passerelles SiteWise Edge

Vous pouvez utiliser les opérations de AWS IoT SiteWise console et d'API pour gérer les passerelles AWS IoT SiteWise Edge. Vous pouvez également utiliser l'application [AWS OpsHub for AWS IoT SiteWise for Windows pour](#) gérer certains aspects de votre passerelle SiteWise Edge à partir de votre appareil local.

Nous vous recommandons vivement d'utiliser l' AWS IoT SiteWise application AWS OpsHub for pour surveiller l'utilisation du disque sur votre appareil local. Vous pouvez également surveiller les CloudWatch métriques Gateway .AvailableDiskSpace et Gateway .UsedPercentageDiskSpace celles d'Amazon et créer des alarmes pour être averti lorsque l'espace disque devient insuffisant. Pour plus d'informations sur les CloudWatch alarmes Amazon, consultez [Créer une CloudWatch alarme basée sur un seuil statique](#).

Assurez-vous que votre appareil dispose de suffisamment d'espace pour les données à venir. Lorsque vous êtes sur le point de manquer d'espace sur votre appareil local, le service supprime automatiquement une petite quantité de données avec les horodatages les plus anciens pour faire de la place aux données à venir.

Pour vérifier si le service a supprimé vos données, procédez comme suit :

1. Connectez-vous au formulaire AWS OpsHub de AWS IoT SiteWise candidature.
2. Sélectionnez Paramètres.
3. Pour Logs, spécifiez un intervalle de temps, puis choisissez Download.
4. Décompressez le fichier journal.
5. Si le fichier journal contient le message suivant, le service a supprimé vos données : des *number* octets de données ont été supprimés pour empêcher le stockage de la passerelle SiteWise Edge de manquer d'espace.

Gérez votre passerelle SiteWise Edge à l'aide de la AWS IoT SiteWise console

Vous pouvez utiliser la AWS IoT SiteWise console pour configurer, mettre à jour et surveiller toutes les passerelles SiteWise Edge de votre AWS compte.

[Vous pouvez consulter vos passerelles SiteWise Edge en accédant à la page Passerelles Edge de la console.AWS IoT SiteWise](#) Pour accéder à la page de détails de la passerelle Edge d'une passerelle spécifique, choisissez le nom d'une passerelle Edge.

Dans l'onglet Vue d'ensemble de la page de détails de la passerelle Edge, vous pouvez effectuer les opérations suivantes :

- Dans la section Sources de données, mettez à jour la configuration des sources de données et configurez des sources de données supplémentaires
- Choisissez Open CloudWatch metrics pour afficher le nombre de points de données ingérés par source de données dans la console CloudWatch des métriques
- Dans la section Fonctionnalités Edge, ajoutez des packs de données à votre passerelle SiteWise Edge en cliquant sur Modifier
- Dans la section Configuration de la passerelle, consultez l'état de connectivité de vos passerelles SiteWise Edge
- Dans la section Configuration de l'éditeur, consultez l'état de synchronisation de la passerelle SiteWise Edge et la configuration du composant de l' AWS IoT SiteWise éditeur

Dans l'onglet Mises à jour de la page de détails de la passerelle Edge, vous pouvez voir les versions actuelles des composants et des packs déployées sur la passerelle Edge. C'est également là que vous déployez les nouvelles versions, lorsqu'elles sont disponibles.

Gérez les passerelles SiteWise Edge à l'aide de AWS OpsHubAWS IoT SiteWise

Vous utilisez l' AWS IoT SiteWise application AWS OpsHub for pour gérer et surveiller vos passerelles SiteWise Edge auto-hébergées. Cette application propose les options de surveillance et de gestion suivantes :

- Sous Vue d'ensemble, vous pouvez effectuer les opérations suivantes :

- Consultez les détails de la passerelle SiteWise Edge qui vous aident à obtenir des informations sur les données de votre appareil de passerelle SiteWise Edge, à identifier les problèmes et à améliorer les performances de la passerelle SiteWise Edge.
- Portails View SiteWise Monitor qui surveillent les données provenant des serveurs locaux et des équipements situés en périphérie. Pour plus d'informations, voir [Contenu AWS IoT SiteWise Monitor](#) du guide de AWS IoT SiteWise Monitor candidature.
- Sous Health, un tableau de bord affiche les données de votre passerelle SiteWise Edge. Les experts du domaine, tels que les ingénieurs de processus, peuvent utiliser le tableau de bord pour avoir une vue d'ensemble du comportement de la passerelle SiteWise Edge.
- Sous Ressources, consultez les ressources déployées sur le périphérique local et la dernière valeur collectée ou calculée pour les propriétés des ressources.
- Dans Réglages, vous pouvez effectuer les opérations suivantes :
 - Si le pack de traitement des données est installé, consultez les informations de configuration de la passerelle SiteWise Edge et synchronisez les ressources avec le AWS cloud.
 - Téléchargez les fichiers d'authentification que vous pouvez utiliser pour accéder à la passerelle SiteWise Edge à l'aide d'autres outils.
 - Téléchargez des journaux que vous pouvez utiliser pour résoudre les problèmes liés à la passerelle SiteWise Edge.
 - Affichez les AWS IoT SiteWise composants déployés sur la passerelle SiteWise Edge.

 Important

Les éléments suivants doivent être utilisés AWS OpsHub pour AWS IoT SiteWise :

- Votre appareil local et l' AWS IoT SiteWise application AWS OpsHub for doivent être connectés au même réseau.
- Le pack de traitement des données doit être activé.

Pour gérer les passerelles SiteWise Edge à l'aide de AWS OpsHub

1. Téléchargez et installez l'application [AWS OpsHubAWS IoT SiteWise pour Windows](#).
2. Ouvrez l'application .

3. Si vous n'avez pas configuré d'informations d'identification locales pour votre passerelle, suivez les étapes [Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification du système d'exploitation local](#) ci-dessous pour les configurer.
4. Vous pouvez vous connecter à votre passerelle SiteWise Edge avec vos informations d'identification Linux ou LDAP (Lightweight Directory Access Protocol). Pour vous connecter à votre passerelle SiteWise Edge, effectuez l'une des opérations suivantes :

Linux

1. Dans Nom d'hôte ou adresse IP, entrez le nom d'hôte ou l'adresse IP de votre appareil local.
2. Pour Authentification, choisissez Linux.
3. Dans Nom d'utilisateur, entrez le nom d'utilisateur de votre système d'exploitation Linux.
4. Dans Mot de passe, entrez le mot de passe de votre système d'exploitation Linux.
5. Choisissez Sign in (Connexion).

LDAP

1. Dans Nom d'hôte ou adresse IP, entrez le nom d'hôte ou l'adresse IP de votre appareil local.
2. Pour Authentification, choisissez LDAP.
3. Dans Nom d'utilisateur, entrez le nom d'utilisateur de votre LDAP.
4. Dans le champ Mot de passe, entrez le mot de passe de votre LDAP.
5. Choisissez Sign in (Connexion).

Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification du système d'exploitation local

Outre le protocole LDAP (Lightweight Directory Access Protocol), vous pouvez utiliser les informations d'identification Linux ou Windows pour accéder à votre passerelle SiteWise Edge auto-hébergée.

 Important

Pour accéder à votre passerelle SiteWise Edge avec des informations d'identification Linux, vous devez activer le pack de traitement de données pour votre passerelle SiteWise Edge.

Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification du système d'exploitation Linux

Les étapes suivantes supposent que vous utilisez un appareil avec Ubuntu. Si vous utilisez une autre distribution Linux, consultez la documentation correspondante à votre appareil.

Pour créer un groupe d'utilisateurs Linux

1. Pour créer un groupe d'administrateurs, exécutez la commande suivante.

```
sudo groupadd --system SWE_ADMIN_GROUP
```

Les utilisateurs du SWE_ADMIN_GROUP groupe peuvent autoriser l'accès administrateur à la passerelle SiteWise Edge.

2. Pour créer un groupe d'utilisateurs, exécutez la commande suivante.

```
sudo groupadd --system SWE_USER_GROUP
```

Les utilisateurs du SWE_USER_GROUP groupe peuvent autoriser l'accès en lecture seule à la passerelle SiteWise Edge.

3. Pour ajouter un utilisateur au groupe d'administrateurs, exécutez la commande suivante. Remplacez *user-name* et *password* par le nom d'utilisateur et le mot de passe que vous souhaitez ajouter.

```
sudo useradd -p $(openssl passwd -1 password) user-name
```

4. Pour ajouter un utilisateur à l'une SWE_ADMIN_GROUP ou l'autre SWE_USER_GROUP, *user-name* remplacez-le par le nom d'utilisateur que vous avez ajouté à l'étape précédente.

```
sudo usermod -a -G SWE_ADMIN_GROUP user-name
```

Vous pouvez désormais utiliser le nom d'utilisateur et le mot de passe pour vous connecter à la passerelle SiteWise Edge sur l' AWS IoT SiteWise application AWS OpsHub for.

Accédez à votre passerelle SiteWise Edge à l'aide des informations d'identification Windows

Les étapes suivantes supposent que vous utilisez un appareil Windows.

 Important

La sécurité est une responsabilité partagée entre vous AWS et vous. Créez une politique de mot de passe robuste comportant au moins 12 caractères et une combinaison de majuscules, de minuscules, de chiffres et de symboles. Définissez également les règles du pare-feu Windows pour autoriser le trafic entrant sur le port 443 et pour bloquer le trafic entrant sur tous les autres ports.

Pour créer un groupe d'utilisateurs Windows Server

1. Exécutez PowerShell en tant qu'administrateur.
 - a. Sur le serveur Windows sur lequel vous souhaitez installer SiteWise Edge Gateway, connectez-vous en tant qu'administrateur.
 - b. Entrez PowerShell dans la barre de recherche de Windows.
 - c. Dans les résultats de recherche, cliquez avec le bouton droit de la souris sur PowerShell l'application Windows. Choisissez Exécuter en tant qu'administrateur.
2. Pour créer un groupe d'administrateurs, exécutez la commande suivante.

```
net localgroup SWE_ADMIN_GROUP /add
```

Vous devez être un utilisateur du SWE_ADMIN_GROUP groupe pour autoriser l'accès administrateur à la passerelle SiteWise Edge.

3. Pour créer un groupe d'utilisateurs, exécutez la commande suivante.

```
net localgroup SWE_USER_GROUP /add
```

Vous devez être un utilisateur du SWE_USER_GROUP groupe pour autoriser l'accès prêt à l'emploi à la passerelle SiteWise Edge.

4. Pour ajouter un utilisateur, exécutez la commande suivante. Remplacez *user-name* et *password* par le nom d'utilisateur et le mot de passe que vous souhaitez créer.

```
net user user-name password /add
```

5. Pour ajouter un utilisateur au groupe d'administrateurs, exécutez la commande suivante. *user-name* Remplacez-le par le nom d'utilisateur que vous souhaitez ajouter.

```
net localgroup SWE_ADMIN_GROUP user-name /add
```

Vous pouvez désormais utiliser le nom d'utilisateur et le mot de passe pour vous connecter à la passerelle SiteWise Edge sur l' AWS IoT SiteWise application AWS OpsHub for.

Gérer le certificat de passerelle SiteWise Edge

Vous pouvez utiliser SiteWise Monitor et des applications tierces, telles que Grafana, sur vos appareils de passerelle SiteWise Edge. Ces applications nécessitent une connexion TLS au service. SiteWise Les passerelles Edge utilisent actuellement un certificat auto-signé. Si vous utilisez un navigateur pour ouvrir les applications, par exemple un portail SiteWise Monitor, il est possible que vous receviez un avertissement concernant un certificat non fiable.

Ce qui suit montre comment télécharger le certificat sécurisé depuis l' AWS IoT SiteWise application AWS OpsHub for.

1. Connectez-vous à l'application.
2. Sélectionnez Paramètres.
3. Pour Authentification, choisissez Télécharger le certificat.

Ce qui suit suppose que vous utilisez Google Chrome ou FireFox. Si vous utilisez un autre navigateur, consultez la documentation correspondante. Pour ajouter le certificat que vous avez téléchargé à l'étape précédente dans un navigateur, effectuez l'une des opérations suivantes :

- Si vous utilisez Google Chrome, suivez les instructions relatives à la [configuration des certificats](#) dans la documentation d'aide de Google Chrome Enterprise.
- Si vous utilisez Firefox, suivez les instructions [pour charger le certificat dans le navigateur Mozilla ou Firefox](#) de la documentation Oracle.

Modifier la version des packs de composants de la passerelle SiteWise Edge

Vous pouvez utiliser la AWS IoT SiteWise console pour modifier la version des packs de composants sur vos passerelles SiteWise Edge.

Pour modifier la version d'un pack de composants de passerelle SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, sélectionnez Passerelles.
3. Sélectionnez la passerelle SiteWise Edge dont vous souhaitez modifier les versions du pack.
4. Sous Configuration de la passerelle, choisissez Afficher les versions du logiciel.
5. Sur la page Modifier les versions du logiciel, pour le pack dont vous souhaitez mettre à jour la version, sélectionnez la version que vous souhaitez déployer et choisissez Déployer.
6. Sélectionnez Exécuté.

Passerelles List SiteWise Edge

Console

Pour répertorier les passerelles SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Consultez la liste de toutes vos passerelles SiteWise Edge.

AWS CLI

Pour répertorier vos passerelles en utilisant AWS CLI, procédez comme suit :

- Utilisez la commande `list-gateways` pour afficher toutes vos passerelles :

```
aws iotsitewise list-gateways
```

Cette commande renvoie une liste de vos passerelles avec leurs noms IDs, ainsi que d'autres informations.

Vous pouvez également définir les paramètres de pagination :

```
aws iotsitewise list-gateways --max-results 10 --next-token your-token
```

Pour plus d'informations, consultez [list-gateways](#) dans la AWS CLI référence des commandes.

Décrire une passerelle SiteWise Edge

Console

Pour afficher les détails de la passerelle

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez le nom de la passerelle dont vous souhaitez consulter les détails.
4. Consultez les détails de la passerelle sur la page des détails de la passerelle Edge.

AWS CLI

Pour obtenir des informations détaillées sur une passerelle spécifique utilisant AWS CLI, procédez comme suit :

- Utilisez la commande `describe-gateway` avec l'ID de passerelle :

```
aws iotsitewise describe-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Cette commande renvoie des informations détaillées sur la passerelle.

Pour plus d'informations, consultez [describe-gateway](#) dans le manuel de référence des AWS CLI commandes.

Création d'une passerelle SiteWise Edge

Console

Pour créer une passerelle SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Cliquez sur Create gateway (Créer une passerelle).
4. Entrez un nom pour votre passerelle.
5. Sélectionnez le groupe Greengrass pour votre passerelle.
6. Ajoutez éventuellement des balises à votre passerelle.
7. Choisissez Créer.

AWS CLI

Pour créer une nouvelle SiteWise passerelle IoT à l'aide de AWS CLI, procédez comme suit :

- Utilisez la commande `create-gateway` pour créer une nouvelle passerelle :

```
aws iotsitewise create-gateway \
  --gateway-name "NewSiteWiseGateway" \
  --gateway-platform '{
    "greengrass": {
      "groupArn": "arn:aws:greengrass:us-east-1:123456789012:group/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
    }
  }' \
  --tags '{
    "Environment": "Production",
    "Location": "Factory1"
  }'
```

Cette commande renvoie l'ID et l'ARN de la nouvelle passerelle :

```
{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

```
}
```

Pour plus d'informations, consultez la section [create-gateway](#) dans le manuel de référence des AWS CLI commandes.

Mettre à jour une passerelle SiteWise Edge

Console

Pour mettre à jour une passerelle SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle que vous souhaitez mettre à jour.
4. Choisissez Modifier.
5. Mettez à jour le nom de la passerelle ou les autres paramètres selon vos besoins.
6. Choisissez Enregistrer.

AWS CLI

Pour mettre à jour une passerelle existante à l'aide de AWS CLI, procédez comme suit :

- Utilisez la commande `update-gateway` pour mettre à jour le nom d'une passerelle :

```
aws iotsitewise update-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \  
  --gateway-name "UpdatedGatewayName"
```

Cette commande ne produit aucune sortie lorsqu'elle réussit.

Pour plus d'informations, consultez [update-gateway](#) dans la référence des AWS CLI commandes.

Mettre à jour la configuration des fonctionnalités de passerelle

Console

Pour mettre à jour la configuration des fonctionnalités de la passerelle

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez le nom de la passerelle que vous souhaitez mettre à jour.
4. Dans la section Sources de données, choisissez Modifier.
5. Mettez à jour la configuration de la source de données selon les besoins.
6. Choisissez Enregistrer.

AWS CLI

Pour mettre à jour la configuration des fonctionnalités d'une passerelle à l'aide de AWS CLI, procédez comme suit :

- Utilisez la `update-gateway-capability-configuration` commande pour mettre à jour la configuration des fonctionnalités :

```
aws iotsitewise update-gateway-capability-configuration \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE \
  --capability-namespace "iotsitewise:opcuacollector:1" \
  --capability-configuration '{
    "sources": [
      {
        "name": "OPC-UA Server",
        "endpoint": {
          "certificateTrust": {
            "type": "TrustAny"
          },
          "endpointUri": "opc.tcp://10.0.0.1:4840",
          "securityPolicy": "NONE",
          "messageSecurityMode": "NONE",
          "identityProvider": {
            "type": "Anonymous"
          }
        },
        "measurementDataStreamPrefix": ""
      }
    ]
  }
```

```
}
]
}'
```

Cette commande renvoie l'espace de noms des fonctionnalités et l'état de synchronisation :

```
{
  "capabilityNamespace": "iotsitewise:opcuacollector:1",
  "capabilitySyncStatus": "CONFIGURING"
}
```

Pour plus d'informations, consultez [update-gateway-capability-configuration](#) le manuel de référence des AWS CLI commandes.

Tag : ressources de passerelle

Console

Pour étiqueter une ressource de passerelle

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez le nom de la passerelle que vous souhaitez étiqueter.
4. Sélectionnez l'onglet Tags (Identifications).
5. Choisissez Gérer les balises.
6. Choisissez Ajouter une nouvelle balise et entrez une clé et une valeur pour chaque balise.
7. Choisissez Enregistrer.

AWS CLI

Pour ajouter des balises à une passerelle à l'aide de AWS CLI, procédez comme suit :

- Utilisez la commande tag-resource pour ajouter des balises à une passerelle :

```
aws iotsitewise tag-resource \
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \
```

```
--tags '{
  "Department": "Operations",
  "Project": "FactoryAutomation"
}'
```

Cette commande ne produit aucune sortie lorsqu'elle réussit.

Pour plus d'informations, consultez [tag-resource](#) dans le manuel de référence des AWS CLI commandes.

Répertorier les balises d'une passerelle

Console

Pour répertorier les balises d'une passerelle

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez le nom de la passerelle pour laquelle vous souhaitez afficher les balises.
4. Sélectionnez l'onglet Tags (Identifications).
5. Consultez la liste des balises associées à la passerelle.

AWS CLI

Pour répertorier les balises associées à une passerelle à l'aide de AWS CLI, procédez comme suit :

- Utilisez la `list-tags-for-resource` commande pour répertorier les balises d'une passerelle :

```
aws iotsitewise list-tags-for-resource \
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
```

Cette commande renvoie les balises associées à la passerelle :

```
{
  "tags": {
    "Environment": "Production",
```



```
    "Location": "Factory1",  
    "Department": "Operations",  
    "Project": "FactoryAutomation"  
  }  
}
```

Pour plus d'informations, consultez [list-tags-for-resource](#) le manuel de référence des AWS CLI commandes.

Supprimer les tags d'une passerelle

Console

Pour supprimer des balises d'une passerelle

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez le nom de la passerelle dont vous souhaitez supprimer les balises.
4. Sélectionnez l'onglet Tags (Identifications).
5. Choisissez Gérer les balises.
6. Cliquez sur l'icône de suppression (X) à côté de chaque étiquette que vous souhaitez supprimer.
7. Choisissez Enregistrer.

AWS CLI

Pour supprimer des balises d'une passerelle à l'aide de AWS CLI, procédez comme suit :

- Utilisez la commande `untag-resource` pour supprimer les balises d'une passerelle :

```
aws iotsitewise untag-resource \  
  --resource-arn "arn:aws:iotsitewise:us-east-1:123456789012:gateway/  
a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE" \  
  --tag-keys '["Project", "Department"]'
```

Cette commande ne produit aucune sortie lorsqu'elle réussit.

Pour plus d'informations, consultez [untag-resource](#) dans le manuel de référence des AWS CLI commandes.

Mettre à jour la version d'un AWS IoT SiteWise composant

Mettez à jour le composant AWS IoT SiteWise passerelle de votre appareil AWS IoT Greengrass principal pour garantir votre accès aux dernières fonctionnalités, aux améliorations de performances et aux correctifs de sécurité.

Pour mettre à jour un AWS IoT SiteWise composant sur AWS IoT Greengrass

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Edge gateways.
3. Sélectionnez la passerelle à modifier, puis choisissez Modifier.
4. Dans Edge Capabilities, sous Versions logicielles, sélectionnez Mises à jour logicielles disponibles. La page Modifier les versions du logiciel s'affiche.
5. Choisissez la version du composant.

Note

Il est recommandé de sélectionner la dernière version disponible. La conservation des composants de la passerelle vous up-to-date permet de conserver des fonctionnalités optimales pour la collecte et le traitement des données industrielles.

6. Choisissez Déployer. Cela lance un AWS IoT Greengrass V2 déploiement pour mettre à jour le AWS IoT SiteWise composant sur la passerelle.

Supprimer une passerelle SiteWise Edge

Console

Pour supprimer la passerelle SiteWise Edge

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Choisissez la passerelle que vous souhaitez supprimer.

4. Sélectionnez Delete (Supprimer).
5. Pour confirmer que vous souhaitez supprimer la passerelle, tapez « Supprimer », puis choisissez Supprimer dans la fenêtre qui apparaît.

AWS CLI

Pour supprimer une passerelle à l'aide de AWS CLI, procédez comme suit :

1. Répertoriez vos passerelles pour identifier l'ID de passerelle que vous souhaitez supprimer.

```
aws iotsitewise list-gateways
```

Cette commande renvoie une liste de vos passerelles avec leurs noms IDs, ainsi que d'autres informations :

```
{
  "gatewaySummaries": [
    {
      "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
      "gatewayName": "ExampleCorpGateway",
      "gatewayCapabilitySummaries": [
        {
          "capabilityNamespace": "iotsitewise:opcuacollector:1",
          "capabilitySyncStatus": "IN_SYNC"
        }
      ],
      "creationDate": 1588369971.457,
      "lastUpdateDate": 1588369971.457
    }
  ]
}
```

2. Supprimez la passerelle en spécifiant son ID :

```
aws iotsitewise delete-gateway --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Cette commande ne produit aucune sortie lorsqu'elle réussit.

Note

Lorsque vous supprimez une passerelle, certains de ses fichiers restent dans le système de fichiers de votre passerelle.

3. Pour vérifier que la passerelle a été supprimée, vous pouvez à nouveau répertorier vos passerelles :

```
aws iotsitewise list-gateways
```

La passerelle supprimée ne doit plus apparaître dans la liste.

Pour plus d'informations, voir [delete-gateway](#) dans la référence des AWS CLI commandes.

Sauvegardez et restaurez les passerelles SiteWise Edge

Cette rubrique explique comment restaurer les passerelles SiteWise Edge et sauvegarder vos données métriques. Si vous rencontrez des problèmes liés à une passerelle SiteWise Edge défectueuse sur le même ordinateur et que vous devez résoudre le problème, veuillez lire la AWS IoT SiteWise documentation [Résolution des problèmes liés à la passerelle SiteWise Edge](#).

Note

Les conseils présentés dans cette rubrique concernent les passerelles SiteWise Edge installées sur la AWS IoT Greengrass V2 version 2.1.0 ou supérieure.

Sauvegardes quotidiennes des données métriques

La création d'une sauvegarde est importante si vous souhaitez transférer ou restaurer les données sur un nouvel ordinateur. La sauvegarde de vos données réduit considérablement le risque de perte de données d'exploitation lors d'un processus de transfert ou de restauration.

Cette section s'applique aux passerelles qui utilisent le pack de traitement des données. Pour plus d'informations sur le pack de traitement des données, voir [Configuration d'un modèle d'actif pour le traitement des données sur SiteWise Edge](#).

Le chemin du dossier influxdb est le suivant :

Linux

```
/greengrass/v2/work/aws.iot.SiteWiseEdgeProcessor/influxdb
```

Windows

```
C:\greengrass\v2\work\aws.iot.SiteWiseEdgeProcessor\influxdb
```

Nous vous recommandons de sauvegarder l'intégralité du dossier avec tout ce qui se trouve en dessous.

Nous vous recommandons de sauvegarder régulièrement vos données métriques depuis le 1.0 SiteWise Edge sur un disque dur externe ou sur le AWS cloud.

Restaurer une passerelle SiteWise Edge

Avant de tenter de restaurer une passerelle SiteWise Edge, assurez-vous que tous les appareils Edge connectés à la passerelle sont arrêtés ou déconnectés.

Utilisez la procédure suivante pour restaurer une passerelle SiteWise Edge :

1. Utilisez le script d'installation téléchargé lors de la création de la passerelle SiteWise Edge pour restaurer la passerelle SiteWise Edge sur le nouvel ordinateur. Lisez la procédure [d'installation du logiciel de passerelle SiteWise Edge sur votre appareil local](#) pour configurer la passerelle SiteWise Edge.

Si vous perdez ou ne trouvez pas le script d'installation, contactez [AWS le Support client](#).

2. Une fois la passerelle SiteWise Edge installée, connectez-vous à la [AWS IoT Greengrass console](#).
3. Pour redéployer les composants, accédez à Gérer, puis sous AWS IoT Greengrass Appareils, sélectionnez Appareils principaux.
4. Dans le tableau des appareils AWS IoT Greengrass principaux, sélectionnez le périphérique principal correspondant à votre passerelle SiteWise Edge.
5. Une fois sur la page de l'appareil, ouvrez l'onglet Déploiements et sélectionnez votre ID de déploiement, cela ouvrira la page Déploiements avec l'ID que vous avez sélectionné.
6. Une fois que vous êtes sur la page Déploiements, en haut à droite, appuyez sur le bouton Actions et sélectionnez l'option Réviser pour lancer un nouveau déploiement. Configurez le

déploiement. Si vous souhaitez conserver le déploiement tel quel, passez à la section Révision et déploiement.

7. Attendez que le statut de déploiement devienne le mêmeCompleted.

 Note

Il faudra également quelques minutes pour que tous les composants de l' SiteWise Edge soient complètement configurés et opérationnels.

Restaurer AWS IoT SiteWise les données

Pour restaurer les données sur une nouvelle machine, procédez comme suit.

1. Copiez le influxdb dossier sur le nouvel ordinateur.
2. Arrêtez le SiteWise EdgeProcessor composant en exécutant la commande suivante dans votre terminal :

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component stop -n  
aws.iot.SiteWiseEdgeProcesso
```

3. Localisez le chemin où vous avez sauvegardé vos données, puis exécutez la commande suivante :

Linux

```
sudo yes | sudo cp -rf <influxdb_backup_path> /greengrass/v2/work/  
aws.iot.SiteWiseEdgeProcessor/influxdb
```

PowerShell

```
Copy-Item -Recurse -Force <influxdb_backup_path>\* C:\greengrass  
\v2\work\aws.iot.SiteWiseEdgeProcessor\
```

Windows

```
robocopy <influxdb_backup_path> C:\greengrass\v2\work  
aws.iot.SiteWiseEdgeProcessor\ /E
```

4. Redémarrez le SiteWiseEdgeProcessor composant :

Linux

```
sudo /greengrass/v2/bin/greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Windows

```
C:\greengrass\v2\bin\greengrass-cli component restart -n  
aws.iot.SiteWiseEdgeProcessor
```

Validez les sauvegardes et restaurations réussies

Utilisez cette procédure pour valider vos données sauvegardées et les restaurations de la passerelle SiteWise Edge.

Note

Cette procédure nécessite que vous ayez installé AWS OpsHub pour AWS IoT SiteWise. Pour plus d'informations, voir [Gestion des passerelles SiteWise Edge à l'aide de AWS OpsHub for AWS IoT SiteWise](#).

1. Ouvrez AWS OpsHub pour AWS IoT SiteWise.
2. Sur la page Paramètres d' SiteWise Edge Gateway, vérifiez l'état de chaque composant répertorié dans le tableau des composants. Vérifiez que la couleur d'état est verte et que l'affichage affiche RUNNING.
3. Validez vos anciennes données sur le tableau de bord du portail pour vérifier que les données passées et les nouvelles données sont correctement configurées. Il y aura un temps d'arrêt entre les données passées et les nouvelles données. Vous devriez, sauf pour voir une durée pendant laquelle aucun point de données n'est collecté.

Si vous rencontrez des problèmes lors de la sauvegarde ou de la restauration d'une passerelle SiteWise Edge, consultez les rubriques de résolution des problèmes suivantes. [Résolution des problèmes liés à une passerelle AWS IoT SiteWise Edge](#).

Passerelles traditionnelles (AWS IoT Greengrass Version 1

Note

SiteWise Les passerelles Edge exécutées ne AWS IoT Greengrass V1 sont disponibles que si vous avez commencé à utiliser cette fonctionnalité avant le 29 juillet 2021. Pour plus d'informations sur l'exécution d'une AWS IoT SiteWise passerelle à AWS IoT Greengrass V2 l'aide de [Hébergez vous-même une passerelle AWS IoT SiteWise Edge avec AWS IoT Greengrass V2](#).

SiteWise Les passerelles Edge fonctionnent désormais exclusivement sur Internet AWS IoT Greengrass V2, offrant ainsi des fonctionnalités et des performances améliorées à vos applications IoT industrielles. Cette dernière version AWS IoT Greengrass V2 représente une évolution architecturale, basée sur un framework moderne basé sur des composants qui permet le déploiement de logiciels modulaires. Il rationalise l'installation via un programme d'installation unifié tout en offrant aux développeurs une plus grande flexibilité dans le déploiement de composants personnalisés et la réalisation de tests locaux. Le modèle basé sur les composants permet une gestion plus efficace des ressources et introduit une approche de configuration simplifiée grâce à des recettes de composants. Cette conception facilite une meilleure gestion des dépendances entre les composants, prend en charge les pratiques de déploiement continu et fournit des fonctionnalités CLI améliorées pour le développement local. En outre, il AWS IoT Greengrass V2 centralise la gestion de la configuration grâce à AWS IoT Core des fonctionnalités de journalisation et de surveillance améliorées, toutes protégées par un modèle d'autorisations de sécurité plus granulaire.

Pour plus d'informations sur la prise en main des passerelles SiteWise Edge en utilisant AWS IoT Greengrass V2, [AWS IoT SiteWise Exigences relatives à la passerelle auto-hébergée Edge](#). Ces ressources fournissent des step-by-step instructions sur la configuration de vos passerelles, la configuration des sources de données et la gestion de votre infrastructure IoT industrielle.

Note

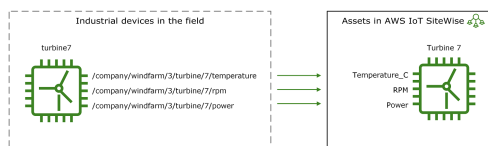
Alors qu'elle AWS continue d'innover et d'améliorer ses services IoT, il est recommandé de se tenir au courant des dernières fonctionnalités et améliorations. Consultez régulièrement

la AWS IoT Greengrass documentation AWS IoT SiteWise et les nouvelles fonctionnalités susceptibles d'optimiser davantage vos solutions IoT industrielles.

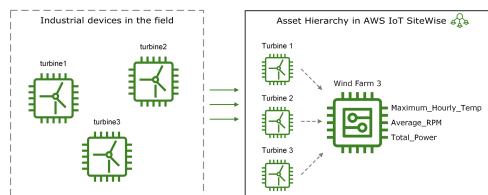
Modéliser des actifs industriels

Vous pouvez créer des représentations virtuelles de votre activité industrielle à l'aide d' AWS IoT SiteWise actifs. Un actif représente un appareil, un équipement ou un processus qui télécharge un ou plusieurs flux de données vers le AWS Cloud. Par exemple, un appareil ressource peut être une éolienne qui envoie des mesures en séries chronologiques de la température de l'air, de la vitesse de rotation de l'hélice et de la puissance de sortie vers les propriétés des ressources dans AWS IoT SiteWise.

Chaque flux de données correspond à un alias de propriété unique. Par exemple, l'alias `/company/windfarm/3/turbine/7/temperature` identifie de manière unique le flux de données de température provenant de la turbine n° 7 dans le parc éolien n° 3. Vous pouvez configurer AWS IoT SiteWise des actifs pour transformer les données de mesure entrantes à l'aide d'expressions mathématiques, par exemple pour convertir les données de température de degrés Celsius en degrés Fahrenheit.



Une ressource peut également représenter un regroupement logique de dispositifs, par exemple un parc éolien entier. Vous pouvez associer des actifs à d'autres actifs pour créer des hiérarchies d'actifs représentant des opérations industrielles complexes. Les actifs peuvent accéder aux données contenues dans leurs actifs enfants associés. Ce faisant, vous pouvez utiliser des AWS IoT SiteWise expressions pour calculer des mesures agrégées, telles que la puissance de production nette d'un parc éolien.



Vous devez créer chaque actif à partir d'un modèle d'actif. Les modèles de ressources sont des structures déclaratives qui normalisent le format de vos ressources. Les modèles d'actifs appliquent des informations cohérentes sur plusieurs actifs du même type afin que vous puissiez traiter les données dans des actifs qui représentent des groupes d'appareils. Dans le diagramme précédent,

vous utilisez le même modèle de ressource pour les trois turbines car toutes les turbines partagent un ensemble commun de propriétés.

Vous pouvez également créer des modèles de composants. Un modèle de composant est un type spécial de modèle d'actif que vous pouvez inclure dans des modèles d'actifs ou d'autres modèles de composants. Vous pouvez utiliser des modèles de composants pour définir des sous-assemblages réutilisables communs, tels que des capteurs, des moteurs, etc., que vous partagez entre plusieurs modèles d'actifs.

Après avoir défini vos modèles de ressources, vous pouvez créer vos ressources industrielles. Pour créer une ressource, sélectionnez un modèle de ressources ACTIVE afin de créer une ressource à partir de ce modèle. Ensuite, vous pouvez remplir des informations spécifiques aux ressources, telles que les alias de flux de données et les attributs. Dans le diagramme précédent, vous créez trois ressources de turbine à partir d'un modèle de ressource, puis vous associez des alias de flux de données comme `/company/windfarm/3/turbine/7/temperature` pour chaque turbine.

Vous pouvez également mettre à jour et supprimer des actifs, des modèles d'actifs et des modèles de composants existants. Lorsque vous mettez à jour un modèle de ressource, chaque ressource basée sur ce modèle de ressource reflète toutes les modifications que vous apportez au modèle sous-jacent. Lorsque vous mettez à jour un modèle de composant, cela s'applique à chaque actif en fonction de chaque modèle d'actif qui fait référence au modèle de composant.

Vos modèles d'actifs peuvent être très complexes, par exemple lorsque vous modélisez un équipement complexe comportant de nombreux sous-composants. Pour que ces modèles d'actifs restent organisés et maintenables, vous pouvez utiliser des modèles composites personnalisés pour regrouper les propriétés associées ou pour réutiliser des composants partagés. Pour de plus amples informations, veuillez consulter [Modèles composites personnalisés \(composants\)](#).

Rubriques

- [État des ressources et des modèles](#)
- [Versions du modèle d'actifs](#)
- [Modèles composites personnalisés \(composants\)](#)
- [Configurer l' AWS IoT SiteWise objet IDs](#)
- [Créez des modèles d'actifs et de composants pour AWS IoT SiteWise](#)
- [Créez des actifs pour les modèles d'actifs dans AWS IoT SiteWise](#)
- [Rechercher des actifs sur Console AWS IoT SiteWise](#)
- [Mettre à jour les valeurs des attributs](#)

- [Associer et dissocier des actifs](#)
- [Mettre à jour les actifs et les modèles](#)
- [Supprimer des actifs et des modèles dans AWS IoT SiteWise](#)
- [Opérations groupées avec actifs et modèles](#)

État des ressources et des modèles

Lorsque vous créez, mettez à jour ou supprimez une ressource, un modèle de ressource ou un modèle de composant, les modifications mettent du temps à se propager. AWS IoT SiteWise résout ces opérations de manière asynchrone et met à jour le statut de chaque ressource. Chaque actif, modèle d'actif et modèle de composant possède un champ d'état qui contient l'état de la ressource et tout message d'erreur, le cas échéant. L'état peut avoir l'une des valeurs suivantes :

- **ACTIVE**— La ressource est active. Il s'agit du seul état dans lequel vous pouvez interroger et interagir avec les actifs, les modèles d'actifs et les modèles de composants.
- **CREATING**— La ressource est en cours de création.
- **UPDATING**— La ressource est en cours de mise à jour.
- **DELETING**— La ressource est en cours de suppression.
- **PROPAGATING**— (Modèles d'actifs et modèles de composants uniquement) Les modifications se propagent à toutes les ressources dépendantes (du modèle d'actif aux actifs, ou du modèle de composant aux modèles d'actifs).
- **FAILED**— La ressource n'a pas pu être validée lors d'une opération de création ou de mise à jour, probablement en raison d'une référence circulaire dans une expression. Vous pouvez supprimer les ressources qui se trouvent dans **FAILED** cet état.

Certaines des opérations de création, de mise à jour et de suppression mettent en AWS IoT SiteWise place un actif, un modèle d'actif ou un modèle de composant dans un état autre que celui **ACTIVE** pendant la résolution de l'opération. Pour interroger ou interagir avec une ressource après avoir effectué l'une de ces opérations, vous devez attendre que son état passe à **ACTIVE**. Sinon, vos demandes échouent.

Rubriques

- [Vérifier l'état d'un actif](#)
- [Vérifier l'état d'un modèle d'actif ou de composant](#)

Vérifier l'état d'un actif

Vous pouvez utiliser la AWS IoT SiteWise console ou l'API pour vérifier l'état d'un actif.

Rubriques

- [Vérifier l'état d'un actif \(console\)](#)
- [Vérifier l'état d'un actif \(AWS CLI\)](#)

Vérifier l'état d'un actif (console)

Utilisez la procédure suivante pour vérifier l'état d'une ressource dans la console AWS IoT SiteWise .

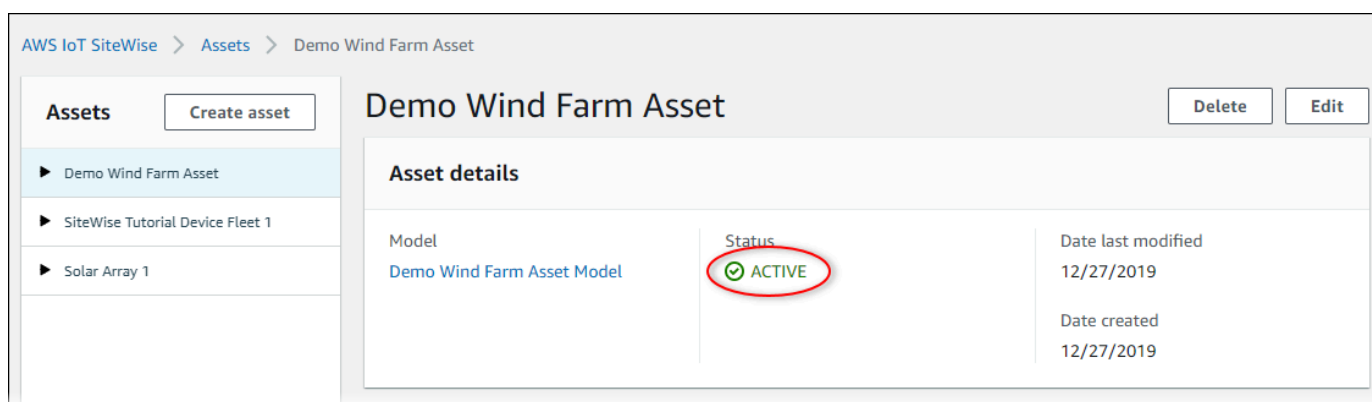
Pour vérifier l'état d'une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource à vérifier.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Recherchez l'état dans le volet Informations relatives à la ressource.



Vérifier l'état d'un actif (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour vérifier l'état d'un actif.

Pour vérifier l'état d'un actif, utilisez l'[DescribeAsset](#) opération avec le `assetId` paramètre.

Pour vérifier l'état d'un actif (AWS CLI)

- Exécutez la commande suivante pour décrire la ressource. Remplacez *asset-id* par l'ID de la ressource ou par l'ID externe. L'identifiant externe est un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

```
aws iotsitewise describe-asset --asset-id asset-id
```

L'opération renvoie une réponse qui contient les détails de la ressource. La réponse contient un `assetStatus` objet dont la structure est la suivante :

```
{
  ...
  "assetStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

L'état de la ressource est dans `assetStatus.state` dans l'objet JSON.

Vérifier l'état d'un modèle d'actif ou de composant

Vous pouvez utiliser la AWS IoT SiteWise console ou l'API pour vérifier l'état d'un modèle d'actif ou d'un modèle de composant.

Rubriques

- [Vérifier l'état d'un modèle d'actif ou d'un modèle de composant \(console\)](#)
- [Vérifier l'état d'un modèle d'actif ou d'un modèle de composant \(AWS CLI\)](#)

Vérifier l'état d'un modèle d'actif ou d'un modèle de composant (console)

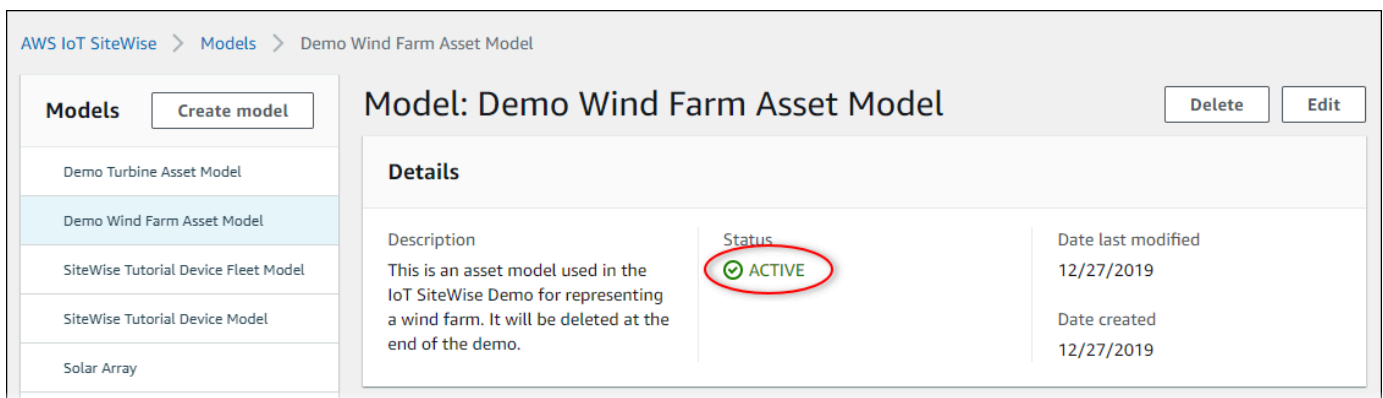
Utilisez la procédure suivante pour vérifier l'état d'un modèle de ressource ou d'un modèle de composant dans la AWS IoT SiteWise console.

Tip

Les modèles d'actifs et les modèles de composants sont tous deux répertoriés sous Modèles dans le volet de navigation. Le panneau Détails du modèle d'actif ou du modèle de composant sélectionné indique de quel type il s'agit.

Pour vérifier l'état d'un modèle d'actif ou d'un modèle de composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle à vérifier.
4. Recherchez État dans le volet Détails.



Vérifier l'état d'un modèle d'actif ou d'un modèle de composant (AWS CLI)

Vous pouvez utiliser le AWS CLI pour vérifier l'état d'un modèle d'actif ou d'un modèle de composant.

Pour vérifier l'état d'un modèle d'actif ou d'un modèle de composant, utilisez l'[DescribeAssetModel](#) opération avec le `assetModelId` paramètre.

 Tip

AWS CLI définit les modèles de composants comme un type de modèle d'actif. Par conséquent, vous utilisez la même [DescribeAssetModel](#) opération pour les deux types de modèles. Le `assetModelType` champ de la réponse indique s'il s'agit d'un `ASSET_MODEL` ou d'un `COMPONENT_MODEL`.

Pour vérifier l'état d'un modèle d'actif ou d'un modèle de composant (AWS CLI)

- Exécutez la commande suivante pour décrire le modèle. Remplacez *asset-model-id* par l'ID ou l'ID externe du modèle d'actif ou du modèle de composant. L'identifiant externe est un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

L'opération renvoie une réponse contenant les détails du modèle. La réponse contient un objet `assetModelStatus` qui présente la structure suivante.

```
{
  ...
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  }
}
```

L'état du modèle se trouve `assetModelStatus.state` dans l'objet JSON.

Versions du modèle d'actifs

AWS IoT SiteWise prend en charge le traitement asynchrone des opérations de création et de mise à jour sur les modèles d'actifs et les modèles de composants. Il met également à jour le statut du modèle.

AWS IoT SiteWise propage les modifications d'un modèle valide dans les demandes de création et de mise à jour à ses ressources dépendantes (du modèle d'actif aux actifs, ou du modèle de composant aux modèles d'actifs). Il place ensuite le modèle en ACTIVE état.

Si la définition de modèle fournie n'est pas valide, AWS IoT SiteWise place le modèle dans un FAILED état. Les modifications ne sont pas répercutées sur les ressources dépendantes. Les ressources dépendantes font référence à la dernière définition du modèle propagée lorsque le modèle était dans un ACTIVE état.

Sur la base des informations ci-dessus, les définitions de modèles comportent deux types de versions de modèle :

1. Dernière version : dernière définition acceptée dans le cadre d'une demande de création ou de mise à jour.
2. Version active : la dernière définition a été traitée avec succès et l'état du modèle est ACTIVE.

Par défaut, les détails de la dernière version du modèle APIs sont renvoyés lorsque describe est appelé sur un modèle d'actif ou un modèle de composant. Dans certains scénarios, la version active du modèle d'actif ou du modèle de composant est requise. Consultez les exemples de scénarios ci-dessous :

- Une opération de mise à jour dont la définition n'est pas valide place votre modèle d'actif dans un FAILED état. Vous devez annuler vos modifications en récupérant la version active du modèle d'actif et en créant une autre demande de mise à jour faisant référence à cette définition valide.
- AWS IoT SiteWise Il existe une application permettant aux clients de consulter les actifs et leurs modèles d'actifs correspondants. Lorsqu'un utilisateur fait référence à la définition du modèle d'actif correspondant à un actif particulier et que le modèle d'actif est dans un FAILED état transitoire UPDATINGPROPAGATING, la dernière version renvoie la définition du modèle d'actif qui n'est pas encore propagée à ses actifs. Dans ce cas, vous devez récupérer la version active du modèle d'actifs auprès des clients.

Rubriques

- [Récupérer la version active d'un modèle d'actif ou d'un modèle de composant \(console\)](#)
- [Récupérez la version active d'un modèle d'actif ou d'un modèle de composant \(AWS CLI\)](#)

Récupérer la version active d'un modèle d'actif ou d'un modèle de composant (console)

Suivez cette procédure pour récupérer la version active d'un modèle de ressource ou d'un modèle de composant dans la AWS IoT SiteWise console.

Tip

Les modèles d'actifs et les modèles de composants sont tous deux répertoriés sous Modèles dans le volet de navigation. Le panneau Détails du modèle d'actif ou du modèle de composant sélectionné indique de quel type il s'agit.

Pour récupérer la version active d'un modèle de ressource ou d'un modèle de composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle pour récupérer sa version active.
 - a. Si le modèle est dans un ACTIVE état, vous visualisez sa version active.
 - b. Si le modèle est dans un FAILED état transitoire UPDATINGPROPAGATING, recherchez la section Voir la version active sous État dans le panneau Détails.

Récupérez la version active d'un modèle d'actif ou d'un modèle de composant (AWS CLI)

Utilisez le AWS CLI pour récupérer la version active d'un modèle d'actif ou d'un modèle de composant.

Pour récupérer la version active d'un modèle d'actif ou d'un modèle de composant, utilisez l'[DescribeAssetModel](#) opération avec le `assetModelVersion` paramètre.

Tip

AWS CLI définit les modèles de composants comme un type de modèle d'actif. Par conséquent, vous utilisez la même [DescribeAssetModel](#) opération pour les deux types de

modèles. Le `assetModelType` champ de la réponse indique s'il s'agit d'un `ASSET_MODEL` ou d'un `COMPONENT_MODEL`.

Pour récupérer la version active d'un modèle d'actif ou d'un modèle de composant (AWS CLI)

- Exécutez la commande suivante pour décrire le modèle. Remplacez *asset-model-id* par l'ID ou l'ID externe du modèle d'actif ou du modèle de composant. L'identifiant externe est un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

L'opération renvoie une réponse contenant les détails du modèle. La réponse contient un `assetModelStatus` objet dont la structure est la suivante.

```
{
  ...
  "assetModelName": "string",
  "assetModelProperties": [ ... ],
  ...,
  "assetModelVersion": "string"
}
```

Modèles composites personnalisés (composants)

Lorsque vous modélisez un actif industriel particulièrement complexe, tel qu'une machine complexe comportant de nombreuses pièces, il peut être difficile de garder vos modèles d'actifs organisés et maintenables.

Dans ce cas, vous pouvez ajouter des modèles composites personnalisés, ou des composants si vous utilisez la console, à vos modèles d'actifs et modèles de composants existants. Ils vous aident à rester organisé en regroupant les propriétés associées et en réutilisant les définitions de sous-composants.

Il existe deux types de modèles composites personnalisés :

- Les modèles composites personnalisés en ligne définissent un ensemble de propriétés groupées qui s'appliquent au modèle d'actif ou au modèle de composant auquel appartient le modèle composite personnalisé. Vous les utilisez pour regrouper les propriétés associées. Ils se composent d'un nom, d'une description et d'un ensemble de propriétés du modèle d'actif. Ils ne sont pas réutilisables.
- Les modèles composites omponent-model-based personnalisés C font référence à un modèle de composant que vous souhaitez inclure dans votre modèle d'actif ou votre modèle de composant. Vous les utilisez pour inclure des sous-assemblages standard dans votre modèle. Ils se composent d'un nom, d'une description et de l'ID du modèle de composant auquel il fait référence. Ils n'ont aucune propriété propre ; le modèle de composant référencé fournit ses propriétés associées à tous les actifs créés.

Les sections suivantes montrent comment utiliser des modèles composites personnalisés dans vos conceptions.

Rubriques

- [Modèles composites personnalisés en ligne](#)
- [Component-model-based modèles composites personnalisés](#)
- [Utiliser des chemins pour référencer les propriétés personnalisées du modèle composite](#)

Modèles composites personnalisés en ligne

Les modèles composites personnalisés en ligne permettent d'organiser votre modèle d'actifs en regroupant les propriétés associées.

Supposons, par exemple, que vous souhaitiez modéliser un actif robotique. Le robot comprend un servomoteur, une alimentation électrique et une batterie. Chacun de ces éléments constitutifs possède ses propres propriétés que vous souhaitez inclure dans le modèle. Vous pouvez définir un modèle d'actif appelé `robot_model` qui possède des propriétés telles que les suivantes.

- `robot_model`
 - `servo_status` (entier)
 - `servo_position` (double)
 - `powersupply_status` (entier)
 - `powersupply_temperature` (double)

- `battery_status` (entier)
- `battery_charge` (double)

Toutefois, dans certains cas, il peut y avoir de nombreux sous-assemblages ou les sous-assemblages eux-mêmes peuvent avoir de nombreuses propriétés. Dans ces cas, le nombre de propriétés peut être tel qu'il devient difficile de les référencer et de les gérer dans une seule liste plate à la racine du modèle, comme dans l'exemple précédent.

Pour faire face à de telles situations, vous pouvez utiliser un modèle composite personnalisé en ligne pour regrouper les propriétés. Un modèle composite personnalisé en ligne est un modèle composite personnalisé qui définit ses propres propriétés. Par exemple, vous pouvez modéliser votre robot comme suit.

- `robot_model`
 - `servo`
 - `status`(entier)
 - `position`(double)
 - `powersupply`
 - `status`(entier)
 - `temperature` (double)
 - `battery`
 - `status`(entier)
 - `charge`(double)

Dans l'exemple précédent, `servo`, `powersupply`, et `battery` sont les noms des modèles composites personnalisés en ligne définis dans le modèle `robot_model` d'actif. Chacun de ces modèles composites définit ensuite ses propres propriétés.

Note

Dans ce cas, chaque modèle composite personnalisé définit ses propres propriétés, de sorte que toutes les propriétés font partie du modèle de ressources lui-même (`robot_model` dans ce cas). Ces propriétés ne sont partagées avec aucun autre modèle d'actif ou modèle de composant. Par exemple, si vous avez créé un autre modèle d'actif qui portait également le

nom de modèle composite personnalisé en ligneservo, le fait d'apporter une modification au modèle interne servo n'robot_modelaffecterait pas la servo définition de l'autre modèle d'actif.

Si vous souhaitez implémenter un tel partage (par exemple, pour n'avoir qu'une seule définition pour un servo, que tous vos modèles d'actifs peuvent partager), vous devez plutôt créer un modèle de composant pour celui-ci, puis créer des modèles component-model-basedcomposites qui le référencent. Consultez la section suivante pour plus de détails.

Pour plus d'informations sur la création de modèles composites personnalisés en ligne, consultez [Création de modèles composites personnalisés \(composants\)](#).

Component-model-based modèles composites personnalisés

Vous pouvez créer un modèle de composant AWS IoT SiteWise pour définir un sous-assemblage standard réutilisable. Une fois que vous avez créé un modèle de composant, vous pouvez y ajouter des références dans vos autres modèles d'actifs et modèles de composants. Pour ce faire, ajoutez un modèle composite component-model-based personnalisé à n'importe quel modèle dans lequel vous souhaitez référencer le composant. Vous pouvez ajouter des références à votre composant à partir de nombreux modèles ou à plusieurs reprises dans le même modèle.

De cette façon, vous pouvez éviter de dupliquer les mêmes définitions entre les modèles. Cela simplifie également la maintenance de vos modèles, car toutes les modifications que vous apportez à un modèle de composant seront répercutées sur tous les modèles d'actifs qui l'utilisent.

Supposons, par exemple, que votre installation industrielle possède de nombreux types d'équipements qui utilisent tous le même type de servomoteur. Certains d'entre eux ont de nombreux servomoteurs dans un seul équipement. Vous créez un modèle d'actif pour chaque type d'équipement, mais vous ne souhaitez pas dupliquer la définition à servo chaque fois. Vous souhaitez le modéliser une seule fois et l'utiliser dans vos différents modèles d'actifs. Si vous modifiez ultérieurement la définition deservo, elle sera mise à jour pour tous vos modèles et actifs.

Pour modéliser le robot de l'exemple précédent de cette manière, vous pouvez définir des servomoteurs, des alimentations et des batteries en tant que modèles de composants, comme ceci.

- servo_component_model
 - status(entier)
 - position(double)

- `powersupply_component_model`
 - `status`(entier)
 - `temperature` (double)

- `battery__component_model`
 - `status`(entier)
 - `charge`(double)

Vous pouvez ensuite définir des modèles d'actifs `robot_model`, tels que ceux qui font référence à ces composants. Plusieurs modèles d'actifs peuvent faire référence au même modèle de composant. Vous pouvez également référencer le même modèle de composant plusieurs fois dans un même modèle d'actif, par exemple si votre robot comporte plusieurs servomoteurs.

- `robot_model`
 - `servo1`(référence : `servo_component_model`)
 - `servo2`(référence : `servo_component_model`)
 - `servo3`(référence : `servo_component_model`)
 - `powersupply` (référence : `powersupply_component_model`)
 - `battery`(référence : `battery_component_model`)

Pour plus d'informations sur la création de modèles de composants, consultez [Création de modèles de composants](#).

Pour plus d'informations sur la façon de référencer vos modèles de composants dans d'autres modèles, consultez [Création de modèles composites personnalisés \(composants\)](#).

Utiliser des chemins pour référencer les propriétés personnalisées du modèle composite

Lorsque vous créez une propriété sur un modèle d'actif, un modèle de composant ou un modèle composite personnalisé, vous pouvez la référencer à partir d'autres propriétés qui utilisent sa valeur, telles que les [transformations](#) et [les métriques](#).

AWS IoT SiteWise vous propose différentes manières de référencer votre propriété. La méthode la plus simple consiste souvent à utiliser son identifiant de propriété. Toutefois, si la propriété que vous souhaitez référencer se trouve sur un modèle composite personnalisé, il peut être préférable de la référencer par chemin.

Un chemin est une séquence ordonnée de segments de chemin qui spécifie une propriété en termes de position parmi les modèles composites imbriqués au sein d'un modèle d'actif et d'un modèle composite.

Obtenir les chemins des propriétés

Vous pouvez obtenir le chemin d'une propriété à partir du path champ correspondant [AssetModelProperty](#).

Supposons, par exemple, que vous disposiez d'un modèle `robot_model` d'actif contenant un modèle `servo` composite personnalisé doté d'une propriété `position`. Si vous appelez `servo`, [DescribeAssetModelCompositeModel](#) la position propriété listera un path champ qui ressemble à ceci :

```
"path": [  
  {  
    "id": "asset model ID",  
    "name": "robot_model"  
  },  
  {  
    "id": "composite model ID",  
    "name": "servo"  
  },  
  {  
    "id": "property ID",  
    "name": "position"  
  }  
]
```

Utilisation des chemins de propriété

Vous pouvez utiliser un chemin de propriété lorsque vous définissez une propriété qui fait référence à d'autres propriétés, telles qu'une transformation ou une métrique.

Une propriété utilise une variable pour référencer une autre propriété. Pour plus d'informations sur l'utilisation des variables, consultez [Utiliser des variables dans les expressions de formule](#).

Lorsque vous définissez une variable pour référencer une propriété, vous pouvez utiliser l'ID de la propriété ou son chemin.

Pour définir une variable qui utilise le chemin de la propriété référencée, spécifiez le `propertyPath` champ de sa valeur.

Par exemple, pour définir un modèle d'actif dont une métrique fait référence à une propriété à l'aide d'un chemin, vous pouvez transmettre une charge utile comme celle-ci à [CreateAssetModel](#):

```
{
  ...
  "assetModelProperties": [
    {
      ...
      "type": {
        "metric": {
          ...
          "variables": [
            {
              "name": "variable name",
              "value": {
                "propertyPath": [
                  path segments
                ]
              }
            }
          ]
        }
      },
      ...
    },
    ...
  ],
  ...
}
```

Configurer l' AWS IoT SiteWise objet IDs

AWS IoT SiteWise définit différents types d'objets persistants, tels que les actifs, les modèles d'actifs, les propriétés et les hiérarchies. Tous ces objets possèdent des identifiants uniques que vous pouvez utiliser pour les récupérer, les mettre à jour et les supprimer.

AWS IoT SiteWise propose différentes options aux clients pour la création d'un identifiant. AWS IoT SiteWise en génère un pour vous par défaut au moment de la création de l'objet. Les utilisateurs peuvent également ajouter les leurs IDs à vos objets.

Rubriques

- [Travailler avec un objet UUIDs](#)
- [Utiliser un usage externe IDs](#)

Travailler avec un objet UUIDs

Chaque objet persistant AWS IoT SiteWise possède un [UUID](#) pour l'identifier. Par exemple, les modèles d'actifs ont un ID de modèle d'actif, les actifs ont un ID d'actif, etc. Cet identifiant est attribué au moment de la création de l'objet et reste inchangé pendant toute la durée de vie de l'objet.

Lorsque vous créez un nouvel objet, il AWS IoT SiteWise génère un identifiant unique pour vous par défaut. Vous pouvez également fournir votre propre identifiant au moment de la création au format UUID.

Note

UUIDs doit être globalement unique dans la AWS région où il a été créé, et pour le même type d'objet. Lorsque vous AWS IoT SiteWise générez automatiquement un identifiant, celui-ci est toujours unique. Si vous choisissez votre propre identifiant, assurez-vous qu'il est unique.

Par exemple, si vous créez un nouveau modèle d'actif en appelant [CreateAssetModel](#), vous pouvez fournir votre propre UUID dans le `assetModelId` champ facultatif de la demande.

En revanche, si vous omettez `assetModelId` de le faire dans la demande, AWS IoT SiteWise génère un UUID pour le nouveau modèle d'actif.

Utiliser un usage externe IDs

Pour définir votre propre identifiant dans un format autre que l'UUID, vous pouvez attribuer un identifiant externe. Par exemple, vous pouvez le faire si vous réutilisez un identifiant que vous utilisez dans un système qui ne l'est pas AWS, ou pour qu'il soit plus lisible par l'homme. Les formats

externes IDs ont un format plus flexible. Vous pouvez les utiliser pour référencer vos objets dans des opérations AWS IoT SiteWise d'API où vous utiliseriez autrement l'UUID.

Comme le UUIDs, chaque identifiant externe doit être unique dans son contexte. Par exemple, vous ne pouvez pas avoir deux modèles d'actifs avec le même ID externe. De plus, comme le UUIDs, un objet ne peut avoir qu'un seul identifiant externe au cours de sa durée de vie, qui ne peut pas changer.

Différences entre externe IDs et UUIDs

IDs Les différences externes sont les suivantes : UUIDs

- Chaque objet possède un UUID, mais les objets externes IDs sont facultatifs.
- AWS IoT SiteWise ne génère jamais de données externes IDs. Vous les fournissez vous-même.
- Si l'objet n'en possède pas déjà un, vous pouvez attribuer un identifiant externe à tout moment.

Format externe IDs

Un identifiant externe valide possède les propriétés suivantes :

- Comporte entre 2 et 128 caractères.
- Les premier et dernier caractères doivent être alphanumériques (A-Z, a-z, 0-9).
- Les caractères autres que le premier et le dernier doivent être alphanumériques ou être l'un des suivants : `_ - . :`

Par exemple, un identifiant externe doit être conforme à l'expression régulière suivante :

```
[a-zA-Z0-9][a-zA-Z0-9_-.:]*[a-zA-Z0-9]+
```

Objets de référence avec interface externe IDs

Dans de nombreux endroits où vous pouvez référencer un objet à l'aide de son UUID, vous pouvez utiliser son identifiant externe à la place, s'il en possède un. Pour ce faire, ajoutez l'ID externe à la chaîne `externalId:`.

Supposons, par exemple, que vous disposiez d'un modèle d'actif dont l'UUID (ID de modèle d'actif) est `1b2c3d4-5678-90ab-cdef-1111EXAMPLE`, qui possède également l'ID externe. `myExternalId` Appelez [DescribeAssetModel](#) pour obtenir des détails à ce sujet. Vous pouvez utiliser l'une des valeurs suivantes comme valeur de `assetModelId` :

- Avec l'identifiant du modèle d'actif (UUID) lui-même : a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
- Avec l'ID externe : externalId:myExternalId

```
aws iotsitewise describe-asset-model --asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
aws iotsitewise describe-asset-model --asset-model-id externalId:myExternalId
```

Note

Le `externalId` : préfixe ne fait pas lui-même partie de l'identifiant externe. Vous ne devez fournir le préfixe que lorsque vous fournissez un identifiant externe à une opération d'API qui accepte l'un UUIDs ou l'autre élément externe IDs. Par exemple, fournissez le préfixe lorsque vous interrogez ou mettez à jour un objet existant.

Lorsque vous définissez un ID externe pour un objet, par exemple lorsque vous créez un modèle de ressource, n'incluez pas le préfixe.

Vous pouvez utiliser l'interface externe IDs à la place de cette UUIDs manière pour de nombreuses opérations d'API AWS IoT SiteWise, mais pas pour toutes. Par exemple, le [GetAssetPropertyValue](#), doit être utilisé UUIDs ; il ne prend pas en charge l'utilisation d'identifiants externes.

Pour déterminer si une opération d'API particulière prend en charge cette utilisation, consultez la [référence des API](#).

Créez des modèles d'actifs et de composants pour AWS IoT SiteWise

AWS IoT SiteWise les modèles d'actifs et les modèles de composants favorisent la standardisation de vos données industrielles. Les modèles d'actifs et de composants représentent la structure et les propriétés de vos actifs industriels et de leurs composants. Les modèles d'actifs définissent l'actif global, tel qu'une éolienne ou une chaîne de fabrication. Les modèles de composants représentent les composants individuels qui constituent l'actif, tels que les lames, les générateurs ou les capteurs. En créant ces modèles, vous pouvez organiser et structurer les données de vos actifs de manière à refléter les relations et les hiérarchies réelles de vos équipements industriels, ce qui facilite leur surveillance, leur analyse et leur maintenance.

Un modèle d'actif ou un modèle de composant contient un nom, une description, des propriétés d'actif et (éventuellement) des modèles composites personnalisés qui regroupent les propriétés ou qui font référence à des modèles de composants pour des sous-assemblages.

Dans AWS IoT SiteWise, vous pouvez créer des modèles d'actifs et des modèles de composants pour représenter la structure et les propriétés de vos actifs industriels et de leurs composants.

- Vous utilisez un modèle d'actifs pour créer des actifs. Outre les fonctionnalités répertoriées ci-dessus, un modèle d'actifs peut également contenir des définitions hiérarchiques qui définissent les relations entre les actifs.
- Un modèle de composant représente un sous-assemblage au sein d'un modèle d'actif ou d'un autre modèle de composant. Lorsque vous créez un modèle de composant, vous pouvez y ajouter des références dans des modèles d'actifs et dans d'autres modèles de composants. Toutefois, vous ne pouvez pas créer de ressources directement à partir de modèles de composants.

Après avoir créé un modèle d'actif ou un modèle de composant, vous pouvez créer des modèles composites personnalisés pour qu'il regroupe des propriétés ou fasse référence à des modèles de composants existants.

Pour plus d'informations sur la création de modèles d'actifs et de modèles de composants, consultez les sections suivantes.

Rubriques

- [Créez des modèles d'actifs dans AWS IoT SiteWise](#)
- [Création de modèles de composants](#)
- [Définir les propriétés des données](#)
- [Création de modèles composites personnalisés \(composants\)](#)

Créez des modèles d'actifs dans AWS IoT SiteWise

AWS IoT SiteWise les modèles d'actifs favorisent la standardisation de vos données industrielles. Un modèle de ressource contient un nom, une description, des propriétés de ressource et des définitions de hiérarchie de ressources. Par exemple, vous pouvez définir un modèle d'éolienne avec des propriétés de température, de rotation par minute (tr/min) et de puissance. Ensuite, vous pouvez définir un modèle de parc éolien avec une propriété de puissance de sortie nette et une définition de hiérarchie d'éoliennes.

 Note

- Nous vous recommandons de modéliser votre opération en commençant par les nœuds de niveau inférieur. Par exemple, créez votre modèle d'éolienne avant de créer votre modèle de parc éolien. Les définitions de hiérarchie des ressources contiennent des références à des modèles de ressources existants. En suivant cette approche, vous pouvez définir des hiérarchies de ressources lors de la création de vos modèles.
- Les modèles d'actifs ne peuvent pas contenir d'autres modèles d'actifs. Si vous devez définir un modèle auquel vous pouvez faire référence en tant que sous-assemblage au sein d'un autre modèle, vous devez plutôt créer un modèle composant-->. Pour de plus amples informations, veuillez consulter [Création de modèles de composants](#).

Les sections suivantes décrivent comment utiliser la AWS IoT SiteWise console ou l'API pour créer des modèles d'actifs. Les sections suivantes décrivent également les différents types de propriétés et de hiérarchies de ressources que vous pouvez utiliser pour créer des modèles.

Rubriques

- [Création d'un modèle d'actif \(console\)](#)
- [Création d'un modèle d'actifs \(AWS CLI\)](#)
- [Exemples de modèles de ressources](#)
- [Définir les hiérarchies des modèles d'actifs](#)

Création d'un modèle d'actif (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour créer un modèle d'actif. La AWS IoT SiteWise console fournit diverses fonctionnalités, telles que la saisie automatique des formules, qui peuvent vous aider à définir des modèles d'actifs valides.

Pour créer un modèle de ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Sélectionnez Create model.
4. Sur la page Créer un modèle, procédez comme suit :

- a. Saisissez un nom pour le modèle de ressource, tel que **Wind Turbine** ou **Wind Turbine Model1**. Ce nom doit être unique pour tous les modèles de votre compte dans cette région.
- b. (Facultatif) Ajoutez un ID externe pour le modèle. Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- c. (Facultatif) Ajoutez des définitions de mesures pour le modèle. Les mesures représentent des flux de données provenant de votre équipement. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).
- d. (Facultatif) Ajoutez des définitions de transformations pour le modèle. Les transformations sont des formules qui font correspondre les données d'un formulaire à un autre. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).
- e. (Facultatif) Ajoutez des définitions de métriques pour le modèle. Les métriques sont des formules qui regroupent les données sur des intervalles de temps. Les métriques peuvent saisir des données provenant des actifs associés, afin que vous puissiez calculer des valeurs qui représentent votre activité ou un sous-ensemble de celle-ci. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).
- f. (Facultatif) Ajoutez des définitions de hiérarchies pour le modèle. Les hiérarchies sont des relations entre les actifs. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).
- g. (Facultatif) Ajoutez des balises pour le modèle de ressource. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
- h. Sélectionnez Create model.

Lorsque vous créez un modèle de ressource, la AWS IoT SiteWise console accède à la page du nouveau modèle. Sur cette page, vous pouvez voir l'état du modèle, qui est initialement CRÉATION. Cette page est automatiquement mise à jour, de sorte que vous pouvez attendre la mise à jour de l'état du modèle.

 Note

Le processus de création de modèles de ressources peut prendre jusqu'à quelques minutes pour les modèles complexes. Une fois que le statut du modèle d'actif est

ACTIF, vous pouvez utiliser le modèle d'actif pour créer des actifs. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

5. (Facultatif) Après avoir créé votre modèle d'actif, vous pouvez configurer votre modèle d'actif pour la périphérie. Pour plus d'informations sur SiteWise Edge, consultez [Configuration des fonctionnalités Edge sur AWS IoT SiteWise Edge](#).
 - a. Sur la page du modèle, choisissez Configurer pour Edge.
 - b. Sur la page de configuration du modèle, choisissez la configuration des bords pour votre modèle. Cela contrôle les endroits où les propriétés associées à ce modèle d'actif AWS IoT SiteWise peuvent être calculées et stockées. Pour plus d'informations sur la configuration de votre modèle pour le bord, consultez [Configurer une source OPC UA dans Edge SiteWise](#).
 - c. Pour la configuration personnalisée de la périphérie, choisissez l'emplacement où vous AWS IoT SiteWise souhaitez calculer et stocker chacune des propriétés de votre modèle d'actifs.

 Note

Les transformations et les métriques associées doivent être configurées pour le même emplacement. Pour plus d'informations sur la configuration de votre modèle pour le bord, consultez [Configurer une source OPC UA dans Edge SiteWise](#).

- d. Choisissez Enregistrer. Sur la page du modèle, votre configuration Edge doit maintenant être configurée.

Création d'un modèle d'actifs (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer un modèle d'actif.

Utilisez l'opération [CreateAssetModel](#) pour créer un modèle de ressource avec des propriétés et des hiérarchies. Cette opération attend une charge utile avec la structure suivante.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition
```



```
}
```

Pour créer un modèle d'actif (AWS CLI)

1. Créez un fichier nommé `asset-model-payload.json` et copiez l'objet JSON suivant dans le fichier.

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "",
  "assetModelDescription": "",
  "assetModelProperties": [

  ],
  "assetModelHierarchies": [

  ],
  "assetModelCompositeModels": [

  ]
}
```

2. Utilisez votre éditeur de texte JSON préféré pour modifier le fichier `asset-model-payload.json` pour les éléments suivants :
- a. Saisissez un nom (`assetModelName`) pour le modèle de ressource, tel que **Wind Turbine** ou **Wind Turbine Model**. Ce nom doit être unique pour tous les modèles d'actifs et modèles de composants de votre compte Région AWS.
- b. (Facultatif) Entrez un ID externe (`assetModelExternalId`) pour le modèle d'actif. Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- c. (Facultatif) Saisissez une description (`assetModelDescription`) pour le modèle de ressource ou supprimez la paire clé-valeur `assetModelDescription`.
- d. (Facultatif) Définissez les propriétés de ressources (`assetModelProperties`) pour le modèle. Pour de plus amples informations, veuillez consulter [Définir les propriétés des données](#).
- e. (Facultatif) Définissez les hiérarchies de ressources (`assetModelHierarchies`) pour le modèle. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).

- f. (Facultatif) Définissez les alarmes pour le modèle. Les alarmes surveillent d'autres propriétés afin que vous puissiez identifier les équipements ou les processus nécessitant une attention particulière. Chaque définition d'alarme est un modèle composite (`assetModelCompositeModels`) qui normalise l'ensemble de propriétés utilisées par l'alarme. Pour plus d'informations, consultez [Surveillez les données avec des alarmes intégrées AWS IoT SiteWise](#) et [Définissez des alarmes sur les modèles d'actifs dans AWS IoT SiteWise](#).
 - g. (Facultatif) Ajoutez des balises (tags) pour le modèle de ressource. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
3. Exécutez la commande suivante pour créer un modèle de ressource à partir de la définition du fichier JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

L'opération renvoie une réponse qui contient l'élément `assetModelId` auquel vous faites référence lors de la création d'une ressource. La réponse contient également l'état du modèle (`assetModelStatus.state`), qui est initialement `CREATING`. L'état du modèle de ressource correspond à `CREATING` jusqu'à ce que les modifications se propagent.

Note

Le processus de création de modèles de ressources peut prendre jusqu'à quelques minutes pour les modèles complexes. Pour vérifier l'état actuel de votre modèle d'actif, utilisez l'[DescribeAssetModel](#) opération en spécifiant le `assetModelId`. Une fois que le statut du modèle de ressource est `ACTIVE`, vous pouvez l'utiliser pour créer des ressources. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

4. (Facultatif) Créez des modèles composites personnalisés pour votre modèle d'actif. Avec les modèles composites personnalisés, vous pouvez regrouper les propriétés au sein du modèle ou inclure un sous-assemblage en faisant référence à un modèle de composant. Pour de plus amples informations, veuillez consulter [Création de modèles composites personnalisés \(composants\)](#).

Exemples de modèles de ressources

Cette section contient des exemples de définitions de modèles d'actifs que vous pouvez utiliser pour créer des modèles d'actifs avec le AWS CLI et AWS IoT SiteWise SDKs. Ces modèles d'actifs représentent une éolienne et un parc éolien. Les actifs des éoliennes ingèrent les données brutes des capteurs et calculent des valeurs telles que la puissance et la vitesse moyenne du vent. Les actifs du parc éolien calculent des valeurs telles que la puissance totale de toutes les éoliennes du parc éolien.

Rubriques

- [Modèle de ressource d'éolienne](#)
- [Modèle de ressource de parc éolien](#)

Modèle de ressource d'éolienne

Le modèle de ressource suivant représente une éolienne dans un parc éolien. L'éolienne ingère les données des capteurs pour calculer des valeurs telles que la puissance et la vitesse moyenne du vent.

Note

Cet exemple de modèle ressemble au modèle d'éolienne de la AWS IoT SiteWise démo. Pour de plus amples informations, veuillez consulter [Utiliser la AWS IoT SiteWise démo](#).

```
{
  "assetModelType": "ASSET_MODEL",
  "assetModelName": "Wind Turbine Asset Model",
  "assetModelDescription": "Represents a turbine in a wind farm.",
  "assetModelProperties": [
    {
      "name": "Location",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Renton"
        }
      }
    }
  ],
}
```

```
{
  "name": "Make",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "Amazon"
    }
  }
},
{
  "name": "Model",
  "dataType": "INTEGER",
  "type": {
    "attribute": {
      "defaultValue": "500"
    }
  }
},
{
  "name": "Torque (KiloNewton Meter)",
  "dataType": "DOUBLE",
  "unit": "kNm",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Direction",
  "dataType": "DOUBLE",
  "unit": "Degrees",
  "type": {
    "measurement": {}
  }
},
{
  "name": "RotationsPerMinute",
  "dataType": "DOUBLE",
  "unit": "RPM",
  "type": {
    "measurement": {}
  }
},
{
  "name": "Wind Speed",
```

```

    "dataType": "DOUBLE",
    "unit": "m/s",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "RotationsPerSecond",
    "dataType": "DOUBLE",
    "unit": "RPS",
    "type": {
      "transform": {
        "expression": "rpm / 60",
        "variables": [
          {
            "name": "rpm",
            "value": {
              "propertyId": "RotationsPerMinute"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Overdrive State",
    "dataType": "DOUBLE",
    "type": {
      "transform": {
        "expression": "gte(torque, 3)",
        "variables": [
          {
            "name": "torque",
            "value": {
              "propertyId": "Torque (KiloNewton Meter)"
            }
          }
        ]
      }
    }
  },
  {
    "name": "Average Power",
    "dataType": "DOUBLE",

```

```

    "unit": "Watts",
    "type": {
      "metric": {
        "expression": "avg(torque) * avg(rps) * 2 * 3.14",
        "variables": [
          {
            "name": "torque",
            "value": {
              "propertyId": "Torque (Newton Meter)"
            }
          },
          {
            "name": "rps",
            "value": {
              "propertyId": "RotationsPerSecond"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    },
    {
      "name": "Average Wind Speed",
      "dataType": "DOUBLE",
      "unit": "m/s",
      "type": {
        "metric": {
          "expression": "avg(windspeed)",
          "variables": [
            {
              "name": "windspeed",
              "value": {
                "propertyId": "Wind Speed"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      }
    }
  ]
}

```

```

    }
  }
}
},
{
  "name": "Torque (Newton Meter)",
  "dataType": "DOUBLE",
  "unit": "Nm",
  "type": {
    "transform": {
      "expression": "knm * 1000",
      "variables": [
        {
          "name": "knm",
          "value": {
            "propertyId": "Torque (KiloNewton Meter)"
          }
        }
      ]
    }
  }
},
{
  "name": "Overdrive State Time",
  "dataType": "DOUBLE",
  "unit": "Seconds",
  "type": {
    "metric": {
      "expression": "statetime(overdrive_state)",
      "variables": [
        {
          "name": "overdrive_state",
          "value": {
            "propertyId": "Overdrive State"
          }
        }
      ]
    },
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  }
}
}

```

```
    }  
  }  
],  
"assetModelHierarchies": []  
}
```

Modèle de ressource de parc éolien

Le modèle de ressource suivant représente un parc éolien qui comprend plusieurs éoliennes. Ce modèle d'actif définit une [hiérarchie](#) par rapport au modèle d'éolienne. Cela permet au parc éolien de calculer des valeurs (telles que la puissance moyenne) à partir des données de toutes les éoliennes du parc éolien.

Note

Cet exemple de modèle ressemble au modèle de parc éolien présenté dans la AWS IoT SiteWise démo. Pour de plus amples informations, veuillez consulter [Utiliser la AWS IoT SiteWise démo](#).

Ce modèle de ressource dépend du [Modèle de ressource d'éolienne](#). Remplacez les valeurs `propertyId` et `childAssetModelId` par celles d'un modèle de ressource d'éolienne existant.

```
{  
  "assetModelName": "Wind Farm Asset Model",  
  "assetModelDescription": "Represents a wind farm.",  
  "assetModelProperties": [  
    {  
      "name": "Code",  
      "dataType": "INTEGER",  
      "type": {  
        "attribute": {  
          "defaultValue": "300"  
        }  
      }  
    },  
    {  
      "name": "Location",  
      "dataType": "STRING",  
      "type": {  
        "attribute": {  
          "defaultValue": "Renton"  
        }  
      }  
    }  
  ]  
}
```



```

    }
  },
  {
    "name": "Reliability Manager",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Mary Major"
      }
    }
  },
  {
    "name": "Total Overdrive State Time",
    "dataType": "DOUBLE",
    "unit": "seconds",
    "type": {
      "metric": {
        "expression": "sum(overdrive_state_time)",
        "variables": [
          {
            "name": "overdrive_state_time",
            "value": {
              "propertyId": "ID of Overdrive State Time property in Wind Turbine
Asset Model",
              "hierarchyId": "Turbine Asset Model"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "5m"
          }
        }
      }
    }
  },
  {
    "name": "Total Average Power",
    "dataType": "DOUBLE",
    "unit": "Watts",
    "type": {
      "metric": {
        "expression": "sum(turbine_avg_power)",

```

```

    "variables": [
      {
        "name": "turbine_avg_power",
        "value": {
          "propertyId": "ID of Average Power property in Wind Turbine Asset Model",
          "hierarchyId": "Turbine Asset Model"
        }
      }
    ],
    "window": {
      "tumbling": {
        "interval": "5m"
      }
    }
  }
},
"assetModelHierarchies": [
  {
    "name": "Turbine Asset Model",
    "childAssetModelId": "ID of Wind Turbine Asset Model"
  }
]
}

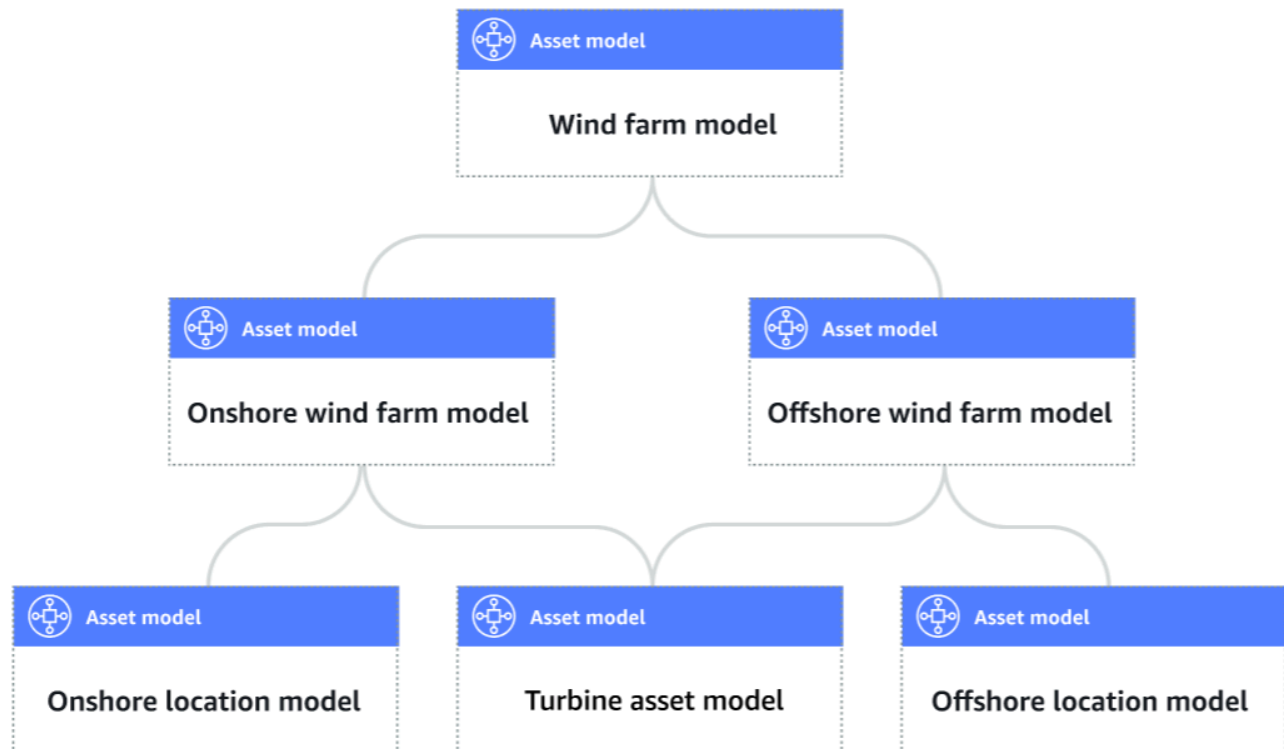
```

Définir les hiérarchies des modèles d'actifs

Vous pouvez définir des hiérarchies de modèles d'actifs afin de créer des associations logiques entre les modèles d'actifs de votre exploitation industrielle. Par exemple, vous pouvez définir un parc éolien composé de parcs éoliens terrestres et offshore. Un parc éolien terrestre comprend une turbine et un emplacement à terre. Un parc éolien en mer contient une turbine et un emplacement en mer.



Asset model hierarchy



Lorsque vous associez un modèle d'actif enfant à un modèle d'actif parent par le biais d'une hiérarchie, les métriques du modèle d'actif parent peuvent entrer des données à partir des métriques du modèle d'actif enfant. Vous pouvez utiliser les hiérarchies et les métriques des modèles d'actifs pour calculer des statistiques qui fournissent un aperçu de votre activité ou d'un sous-ensemble de celle-ci. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).

Chaque hiérarchie définit une relation entre un modèle d'actif parent et un modèle d'actif enfant. Dans un modèle d'actif parent, vous pouvez définir plusieurs hiérarchies pour le même modèle d'actif enfant. Par exemple, si vous avez deux types d'éoliennes différents dans vos parcs éoliens, où toutes les éoliennes sont représentées par le même modèle d'actif, vous pouvez définir une hiérarchie pour chaque type. Vous pouvez ensuite définir des mesures dans le modèle de parc éolien afin de calculer des statistiques indépendantes et combinées pour chaque type d'éolienne.

Un modèle d'actif parent peut être associé à plusieurs modèles d'actifs enfants. Par exemple, si vous avez un parc éolien terrestre et un parc éolien offshore représentés par deux modèles d'actifs différents, vous pouvez associer ces modèles d'actifs au même modèle d'actif de parc éolien parent.

Un modèle d'actif enfant peut également être associé à plusieurs modèles d'actifs parents. Par exemple, si vous avez deux types de parcs éoliens différents, où toutes les éoliennes sont représentées par le même modèle d'actif, vous pouvez associer le modèle d'actif d'éolienne à différents modèles d'actifs de parcs éoliens.

Note

Lorsque vous définissez une hiérarchie de modèles d'actifs, le modèle d'actif enfant doit être ACTIVE ou avoir une ACTIVE version précédente. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Après avoir défini des modèles de ressources hiérarchiques et créé des ressources, vous pouvez associer les ressources pour terminer la relation parent-enfant. Pour plus d'informations, consultez [Créez des actifs pour les modèles d'actifs dans AWS IoT SiteWise](#) et [Associer et dissocier des actifs](#).

Rubriques

- [Définition de hiérarchies de modèles d'actifs \(console\)](#)
- [Définir des hiérarchies d'actifs \(\)AWS CLI](#)

Définition de hiérarchies de modèles d'actifs (console)

Lorsque vous définissez une hiérarchie pour un modèle d'actif dans la AWS IoT SiteWise console, vous spécifiez les paramètres suivants :

- Nom de la hiérarchie : nom de la hiérarchie, par exemple **Wind Turbines**.
- Modèle hiérarchique : modèle d'actif enfant.
- ID externe de hiérarchie (facultatif) : il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actif \(console\)](#).

Définir des hiérarchies d'actifs (AWS CLI)

Lorsque vous définissez une hiérarchie pour un modèle d'actif avec l' AWS IoT SiteWise API, vous spécifiez les paramètres suivants :

- **name**— Le nom de la hiérarchie, par exemple **Wind Turbines**.
- **childAssetModelId**— L'ID ou l'ID externe du modèle d'actif enfant pour la hiérarchie. Vous pouvez utiliser cette [ListAssetModels](#) opération pour trouver l'ID d'un modèle d'actif existant.

Exemple Exemple de définition de hiérarchie

L'exemple suivant illustre une hiérarchie de modèles d'actifs qui représente la relation entre un parc éolien et les éoliennes. Cet objet est un exemple de [AssetModelHierarchy](#). Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actifs \(AWS CLI\)](#).

```
{
  ...
  "assetModelHierarchies": [
    {
      "name": "Wind Turbines",
      "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE"
    },
  ],
}
```

Création de modèles de composants

Utilisez des modèles de AWS IoT SiteWise composants pour définir des sous-assemblages auxquels vous pouvez faire référence à partir de modèles de ressources ou d'autres modèles de composants. De cette façon, vous pouvez réutiliser la définition du composant dans plusieurs autres modèles, ou plusieurs fois dans le même modèle.

Le processus de définition d'un modèle de composant est très similaire à celui d'un modèle d'actif. Tout comme un modèle d'actif, un modèle de composant possède un nom, une description et des propriétés d'actif. Toutefois, les modèles de composants ne peuvent pas inclure de définitions de hiérarchie des actifs, car les modèles de composants eux-mêmes ne peuvent pas être utilisés pour créer directement des actifs. Les modèles de composants ne peuvent pas non plus définir d'alarmes.

Par exemple, vous pouvez définir un composant pour un servomoteur avec des propriétés de température du moteur, de température du codeur et de résistance d'isolement. Vous pouvez ensuite

définir un modèle d'actif pour les équipements contenant des servomoteurs, tels qu'une machine à commande numérique.

Note

- Nous vous recommandons de modéliser votre opération en commençant par les nœuds de niveau inférieur. Par exemple, créez le composant de votre servomoteur avant de créer le modèle d'actif de votre machine CNC. Les modèles d'actifs contiennent des références à des modèles de composants existants.
- Vous ne pouvez pas créer un actif directement à partir d'un modèle de composant. Pour créer un actif qui utilise votre composant, vous devez créer un modèle d'actif pour votre actif. Ensuite, vous créez un modèle composite personnalisé qui fait référence à votre composant. Pour plus d'informations sur la création de modèles d'actifs, voir [Créez des modèles d'actifs dans AWS IoT SiteWise](#) Pour plus d'informations sur la création de modèles composites personnalisés, voir [Création de modèles composites personnalisés \(composants\)](#).

Les sections suivantes décrivent comment utiliser l' AWS IoT SiteWise API pour créer des modèles de composants.

Rubriques

- [Création d'un modèle de composant \(AWS CLI\)](#)
- [Exemple de modèle de composant](#)

Création d'un modèle de composant (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer un modèle de composant.

Utilisez cette [CreateAssetModel](#) opération pour créer un modèle de composant avec des propriétés. Cette opération attend une charge utile dont la structure est la suivante :

```
{
  "assetModelType": "COMPONENT_MODEL",
  "assetModelName": "String",
  "assetModelDescription": "String",
```

```
"assetModelProperties": Array of AssetModelProperty,  
}
```

Pour créer un modèle de composant (AWS CLI)

1. Créez un fichier appelé, `component-model-payload.json` puis copiez l'objet JSON suivant dans le fichier :

```
{  
  "assetModelType": "COMPONENT_MODEL",  
  "assetModelName": "",  
  "assetModelDescription": "",  
  "assetModelProperties": [  
  
  ]  
}
```

2. Utilisez votre éditeur de texte JSON préféré pour modifier le fichier `component-model-payload.json` pour les éléments suivants :
 - a. Entrez un nom (`assetModelName`) pour le modèle de composant, tel que **Servo Motor** ou **Servo Motor Model**. Ce nom doit être unique pour tous les modèles d'actifs et modèles de composants de votre compte Région AWS.
 - b. (Facultatif) Entrez un ID externe (`assetModelExternalId`) pour le modèle de composant. Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
 - c. (Facultatif) Saisissez une description (`assetModelDescription`) pour le modèle de ressource ou supprimez la paire clé-valeur `assetModelDescription`.
 - d. (Facultatif) Définissez les propriétés des actifs (`assetModelProperties`) pour le modèle de composant. Pour de plus amples informations, veuillez consulter [Définir les propriétés des données](#).
 - e. (Facultatif) Ajoutez des balises (tags) pour le modèle de ressource. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
3. Exécutez la commande suivante pour créer un modèle de composant à partir de la définition du fichier JSON.

```
aws iotsitewise create-asset-model --cli-input-json file://component-model-  
payload.json
```

L'opération renvoie une réponse contenant ce à quoi vous `assetModelId` faites référence lorsque vous ajoutez une référence à votre modèle de composant dans un modèle d'actif ou un autre modèle de composant. La réponse contient également l'état du modèle (`assetModelStatus.state`), qui est initialement `CREATING`. L'état du modèle de composant est maintenu `CREATING` jusqu'à ce que les modifications se propagent.

 Note

Le processus de création de modèles de composants peut prendre jusqu'à quelques minutes pour les modèles complexes. Pour vérifier l'état actuel de votre modèle de composant, utilisez l'[DescribeAssetModel](#) opération en spécifiant le `assetModelId`. Une fois que le statut du modèle de composant est `ACTIVE` défini, vous pouvez ajouter des références à votre modèle de composant dans des modèles d'actifs ou d'autres modèles de composants. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

4. (Facultatif) Créez des modèles composites personnalisés pour votre modèle de composant. Avec les modèles composites personnalisés, vous pouvez regrouper les propriétés au sein du modèle ou inclure un sous-assemblage en faisant référence à un autre modèle de composant. Pour de plus amples informations, veuillez consulter [Création de modèles composites personnalisés \(composants\)](#).

Exemple de modèle de composant

Cette section contient un exemple de définition de modèle de composant que vous pouvez utiliser pour créer un modèle de composant avec le AWS CLI et AWS IoT SiteWise SDKs. Ce modèle de composant représente un servomoteur qui peut être utilisé dans un autre équipement, tel qu'une machine à commande numérique.

Rubriques

- [Modèle de composant du servomoteur](#)

Modèle de composant du servomoteur

Le modèle de composant suivant représente un servomoteur qui peut être utilisé dans des équipements tels que des machines à commande numérique. Le servomoteur fournit diverses

mesures, telles que les températures et la résistance électrique. Ces mesures sont disponibles sous forme de propriétés sur des actifs créés à partir de modèles d'actifs faisant référence au modèle de composant du servomoteur.

```
{
  "assetModelName": "ServoMotor",
  "assetModelType": "COMPONENT_MODEL",
  "assetModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Définir les propriétés des données

Les propriétés des actifs sont les structures de chaque actif qui contiennent des données sur les actifs. Les propriétés des ressources peuvent être l'un des types suivants :

- **Attributs** : propriétés généralement statiques d'un actif, telles que le fabricant de l'appareil ou la région géographique. Pour de plus amples informations, veuillez consulter [Définir des données statiques \(attributs\)](#).
- **Mesures** — Les flux de données brutes des capteurs de l'appareil d'un actif, tels que les valeurs de vitesse de rotation horodatées ou les valeurs de température horodatées en degrés Celsius. Une mesure est définie par un alias de flux de données. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).

- Transformations : valeurs de série chronologique transformées d'un actif, telles que les valeurs de température horodatées en degrés Fahrenheit. Une transformation est définie par une expression et les variables à consommer avec cette expression. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).
- Métriques : données d'un actif agrégées sur un intervalle de temps spécifié, tel que la température moyenne horaire. Une métrique est définie par un intervalle de temps, une expression et les variables à consommer avec cette expression. Les expressions métriques peuvent saisir les propriétés métriques des actifs associés, afin que vous puissiez calculer les mesures qui représentent votre activité ou un sous-ensemble de votre activité. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).

Pour de plus amples informations, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

Pour obtenir un exemple d'utilisation des mesures, des transformations et des métriques pour calculer l'efficacité globale de l'équipement (OEE), veuillez consulter [Calculez l'OEE en AWS IoT SiteWise](#).

Rubriques

- [Définir des données statiques \(attributs\)](#)
- [Définir les flux de données provenant de l'équipement \(mesures\)](#)
- [Transformer les données \(transformations\)](#)
- [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#)
- [Utiliser des expressions de formule](#)

Définir des données statiques (attributs)

Les attributs des actifs représentent des informations généralement statiques, telles que le fabricant de l'appareil ou l'emplacement géographique. Chaque ressource que vous créez à partir d'un modèle de ressource contient les attributs de ce modèle.

Rubriques

- [Définir les attributs \(console\)](#)
- [Définir les attributs \(AWS CLI\)](#)

Définir les attributs (console)

Lorsque vous définissez un attribut pour un modèle de ressource dans la AWS IoT SiteWise console, vous spécifiez les paramètres suivants :

- Nom : nom de la propriété.
- Valeur par défaut — (Facultatif) La valeur par défaut de cet attribut. Les ressources créées à partir du modèle ont cette valeur pour l'attribut. Pour de plus amples informations sur la façon de remplacer la valeur par défaut dans une ressource créée à partir d'un modèle, veuillez consulter [Mettre à jour les valeurs des attributs](#).
- Type de données : type de données de la propriété, qui est l'un des suivants :
 - String — Chaîne de 1024 octets maximum.
 - Entier — Un entier signé de 32 bits dont la plage est comprise entre [-2 147 483 648, 2 147 483 647].
 - Double : nombre à virgule flottante avec une plage $[-10^{100}, 10^{100}]$ et une double précision IEEE 754.
 - Boolean — **true** ou **false**
- ID externe — (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actif \(console\)](#).

Définir les attributs (AWS CLI)

Lorsque vous définissez un attribut pour un modèle d'actif avec l' AWS IoT SiteWise API, vous spécifiez les paramètres suivants :

- name— Le nom de la propriété.
- defaultValue— (Facultatif) La valeur par défaut de cet attribut. Les ressources créées à partir du modèle ont cette valeur pour l'attribut. Pour de plus amples informations sur la façon de remplacer la valeur par défaut dans une ressource créée à partir d'un modèle, veuillez consulter [Mettre à jour les valeurs des attributs](#).
- dataType— Le type de données de la propriété, qui est l'un des suivants :
 - STRING— Chaîne de 1024 octets maximum.

- **INTEGER**— Un entier signé de 32 bits dont la plage est comprise entre [-2 147 483 648, 2 147 483 647].
- **DOUBLE**— Un nombre à virgule flottante avec une plage $[-10^{100}, 10^{100}]$ et une double précision IEEE 754.
- **BOOLEAN**— `true` ou `false`.
- **externalId**— (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Exemple Exemple de définition d'attribut

L'exemple suivant illustre un attribut qui représente le numéro de modèle d'une ressource avec une valeur par défaut. Cet objet est un exemple d'objet [AssetModelProperty](#) contenant un [attribut](#). Vous pouvez spécifier cet objet dans le cadre de la charge utile de la demande [CreateAssetModel](#) pour créer une propriété d'attribut. Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actifs \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    {
      "name": "Model number",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "BLT123"
        }
      }
    }
  ],
  ...
}
```

Définir les flux de données provenant de l'équipement (mesures)

Une mesure représente le flux de données brutes du capteur d'un appareil, tel que les valeurs de température horodatées ou les valeurs de rotations par minute (RPM) horodatées.

Rubriques

- [Définir les mesures \(console\)](#)
- [Définir les mesures \(AWS CLI\)](#)

Définir les mesures (console)

Lorsque vous définissez une mesure pour un modèle d'actif dans la AWS IoT SiteWise console, vous spécifiez les paramètres suivants :

- Nom : nom de la propriété.
- Unité — (Facultatif) L'unité scientifique de la propriété, telle que mm ou Celsius.
- Type de données : type de données de la propriété, qui est l'un des suivants :
 - String — Chaîne de 1024 octets maximum.
 - Entier — Un entier signé de 32 bits dont la plage est comprise entre [-2 147 483 648, 2 147 483 647].
 - Double : nombre à virgule flottante avec une plage $[-10^{100}, 10^{100}]$ et une double précision IEEE 754.
 - Boolean — **true** ou **false**
- ID externe — (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actif \(console\)](#).

Définir les mesures (AWS CLI)

Lorsque vous définissez une mesure pour un modèle d'actif à l'aide de l' AWS IoT SiteWise API, vous spécifiez les paramètres suivants :

- name— Le nom de la propriété.
- dataType— Le type de données de la propriété, qui est l'un des suivants :
 - STRING— Chaîne de 1024 octets maximum.
 - INTEGER— Un entier signé de 32 bits dont la plage est comprise entre [-2 147 483 648, 2 147 483 647].
 - DOUBLE— Un nombre à virgule flottante avec une plage $[-10^{100}, 10^{100}]$ et une double précision IEEE 754.

- **BOOLEAN**— `true` ou `false`.
- **unit**— (Facultatif) L'unité scientifique de la propriété, telle que `mm` ou `Celsius`.
- **externalId**— (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Exemple Exemple de définition de mesure

L'exemple suivant illustre une mesure qui représente les lectures du capteur de température d'une ressource. Cet objet est un exemple d'objet [AssetModelProperty](#) contenant une [mesure](#). Vous pouvez spécifier cet objet dans le cadre de la charge utile de la demande [CreateAssetModel](#) pour créer une propriété de mesure. Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actifs \(AWS CLI\)](#).

La structure [de mesure](#) est une structure vide lorsque vous définissez un modèle d'actif, car vous configurez ultérieurement chaque actif pour utiliser des flux de données d'appareils uniques. Pour plus d'informations sur la façon de connecter la propriété de mesure d'un actif au flux de données du capteur d'un appareil, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

```
{
    ...
    "assetModelProperties": [
        {
            "name": "Temperature C",
            "dataType": "DOUBLE",
            "type": {
                "measurement": {}
            },
            "unit": "Celsius"
        }
    ],
    ...
}
```

Transformer les données (transformations)

Les transformations sont des expressions mathématiques qui mappent les points de données des propriétés des actifs d'un formulaire à l'autre. Une expression de transformation comprend des variables de propriété d'actif, des littéraux, des opérateurs et des fonctions. Les points de données

transformés entretiennent une one-to-one relation avec les points de données d'entrée. AWS IoT SiteWise calcule un nouveau point de données transformé chaque fois que l'une des propriétés d'entrée reçoit un nouveau point de données.

 Note

Pour les mises à jour de propriétés avec le même horodatage, les valeurs de sortie peuvent être remplacées par des mises à jour provenant d'autres propriétés entrantes.

Par exemple, si votre ressource a un flux de mesure de température nommé `Temperature_C` avec des unités en Celsius, vous pouvez convertir chaque point de données en Fahrenheit avec la formule $Temperature\_F = 9/5 * Temperature\_C + 32$. Chaque fois qu'un point de données est AWS IoT SiteWise reçu dans le flux de `Temperature_C` mesure, la `Temperature_F` valeur correspondante est calculée en quelques secondes et disponible en tant que `Temperature_F` propriété.

Si votre transformation contient plusieurs variables, le point de données qui arrive le plus tôt lance immédiatement le calcul. Prenons l'exemple d'un fabricant de pièces utilisant une transformation pour contrôler la qualité de ses produits. En utilisant une norme différente basée sur le type de pièce, le fabricant utilise les mesures suivantes pour représenter le processus :

- `Part_Number`- Chaîne identifiant le type de pièce.
- `Good_Count`- Un entier qui augmente d'un si la pièce répond à la norme.
- `Bad_Count`- Un entier qui augmente d'un si la pièce ne répond pas à la norme.

Le fabricant crée également une transformation `Quality_Monitor`, qui équivaut à `if(eq(Part_Number, "BLT123") and (Bad_Count / (Good_Count + Bad_Count) > 0.1), "Caution", "Normal")`.

Cette transformation surveille le pourcentage de pièces défectueuses produites pour un type de pièce spécifique. Si le numéro de pièce est BLT123 et que le pourcentage de pièces défectueuses dépasse 10 % (0,1), la transformation est renvoyée "Caution". Dans le cas contraire, la transformation est renvoyée "Normal".

 Note

- Si `Part_Number` elle reçoit un nouveau point de données avant les autres mesures, la `Quality_Monitor` transformation utilise la nouvelle `Part_Number` valeur et les dernières `Bad_Count` valeurs `Good_Count` et. Pour éviter les erreurs, effectuez une réinitialisation `Good_Count` et `Bad_Count` avant le prochain cycle de fabrication.
- Utilisez [des métriques](#) si vous souhaitez évaluer les expressions uniquement lorsque toutes les variables ont reçu de nouveaux points de données.

Rubriques

- [Définir les transformations \(console\)](#)
- [Définir les transformations \(AWS CLI\)](#)

Définir les transformations (console)

Lorsque vous définissez une transformation pour un modèle d'actif dans la AWS IoT SiteWise console, vous spécifiez les paramètres suivants :

- Nom : nom de la propriété.
- Unité — (Facultatif) L'unité scientifique de la propriété, telle que mm ou Celsius.
- Type de données : type de données de la transformation, qui peut être double ou chaîne.
- ID externe — (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- Formule — L'expression de transformation. Les expressions de transformation ne peuvent pas utiliser de fonctions d'agrégation ou de fonctions temporelles. Pour ouvrir la fonction de saisie automatique, commencez à taper ou appuyez sur la flèche vers le bas. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#).

 Important

Les transformations peuvent saisir des propriétés de type entier, double, booléen ou chaîne. Les booléens sont convertis en 0 (faux) et 1 (vrai).

Les transformations doivent saisir une ou plusieurs propriétés qui ne sont pas des attributs et un certain nombre de propriétés d'attribut. AWS IoT SiteWise calcule un nouveau point

de données transformé chaque fois que la propriété d'entrée qui n'est pas un attribut reçoit un nouveau point de données. Les nouvelles valeurs d'attribut ne lancent pas les mises à jour de transformation. Le même taux de demande pour les opérations de l'API de données relatives aux propriétés des actifs s'applique aux résultats des calculs de transformation. Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actif \(console\)](#).

Définir les transformations (AWS CLI)

Lorsque vous définissez une transformation pour un modèle d'actif à l'aide de l' AWS IoT SiteWise API, vous spécifiez les paramètres suivants :

- **name**— Le nom de la propriété.
- **unit**— (Facultatif) L'unité scientifique de la propriété, telle que mm ou Celsius.
- **dataType**— Le type de données de la transformation, qui doit être DOUBLE ou STRING.
- **externalId**— (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- **expression**— L'expression de transformation. Les expressions de transformation ne peuvent pas utiliser de fonctions d'agrégation ou de fonctions temporelles. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#).
- **variables**— La liste des variables qui définit les autres propriétés de votre actif à utiliser dans l'expression. Chaque structure de variable contient un nom simple à utiliser dans l'expression et une structure **value** qui identifie la propriété à lier à cette variable. La structure **value** contient les informations suivantes :
 - **propertyId**— L'ID de la propriété à partir de laquelle les valeurs doivent être saisies. Vous pouvez utiliser le nom de la propriété au lieu de son ID.

⚠ Important

Les transformations peuvent saisir des propriétés de type entier, double, booléen ou chaîne. Les booléens sont convertis en 0 (faux) et 1 (vrai).

Les transformations doivent saisir une ou plusieurs propriétés qui ne sont pas des attributs et un certain nombre de propriétés d'attribut. AWS IoT SiteWise calcule un nouveau point de données transformé chaque fois que la propriété d'entrée qui n'est pas un attribut reçoit un nouveau point de données. Les nouvelles valeurs d'attribut ne lancent pas les mises à jour de transformation. Le même taux de demande pour les opérations de l'API de données relatives aux propriétés des actifs s'applique aux résultats des calculs de transformation. Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

Exemple définition de la transformation

L'exemple suivant illustre une propriété de transformation qui convertit les données de mesure de température d'une ressource de Celsius en Fahrenheit. Cet objet est un exemple d'objet [AssetModelProperty](#) contenant une [transformation](#). Vous pouvez spécifier cet objet dans le cadre de la charge utile de la demande [CreateAssetModel](#) pour créer une propriété de transformation. Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actifs \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "type": {
        "transform": {
          "expression": "9/5 * temp_c + 32",
          "variables": [
            {
              "name": "temp_c",
```

```

        "value": {
            "propertyId": "Temperature C"
        }
    }
]
}
},
"unit": "Fahrenheit"
}
],
...
}

```

Exemple définition de transformation contenant trois variables

L'exemple suivant illustre une propriété de transformation qui renvoie un message d'avertissement ("Caution") si plus de 10 % des BLT123 pièces ne répondent pas à la norme. Dans le cas contraire, il renvoie un message d'information ("Normal").

```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "Quality_Monitor",
            "dataType": "STRING",
            "type": {
                "transform": {
                    "expression": "if(eq(Part_Number,\"BLT123\") and (Bad_Count / (Good_Count +
Bad_Count) > 0.1), \"Caution\", \"Normal\")",
                    "variables": [
                        {
                            "name": "Part_Number",
                            "value": {
                                "propertyId": "Part Number"
                            }
                        },
                        {
                            "name": "Good_Count",
                            "value": {
                                "propertyId": "Good Count"
                            }
                        }
                    ],
                }
            }
        },
    ]
}

```

```
{
  "name": "Bad_Count",
  "value": {
    "propertyId": "Bad Count"
  }
}
]
```

Données agrégées provenant de propriétés et d'autres actifs (métriques)

Les métriques sont des expressions mathématiques qui utilisent des fonctions d'agrégation pour traiter tous les points de données en entrée et en sortie un seul point de données par intervalle de temps spécifié. Par exemple, une métrique peut calculer la température horaire moyenne à partir d'un flux de données de température.

Les métriques peuvent entrer des données à partir des métriques des ressources associées, de sorte que vous pouvez calculer des statistiques qui fournissent un aperçu de votre opération ou d'un sous-ensemble de votre opération. Par exemple, une métrique peut calculer la température horaire moyenne pour toutes les éoliennes d'un parc éolien. Pour de plus amples informations sur la définition des associations entre les ressources, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).

Les métriques peuvent également saisir des données provenant d'autres propriétés sans agréger les données sur chaque intervalle de temps. Si vous spécifiez un [attribut](#) dans une formule, AWS IoT SiteWise utilise la [dernière](#) valeur de cet attribut lors du calcul de la formule. Si vous spécifiez une métrique dans une formule, AWS IoT SiteWise utilise la [dernière](#) valeur de l'intervalle de temps pendant lequel la formule est calculée. Cela signifie que vous pouvez définir des indicateurs tels que $OEE = Availability * Quality * Performance$ AvailabilityQuality, où et quels Performance sont tous les autres indicateurs du même modèle d'actif.

AWS IoT SiteWise calcule également automatiquement un ensemble de mesures d'agrégation de base pour toutes les propriétés des actifs. Pour réduire les coûts de calcul, vous pouvez utiliser ces agrégats au lieu de définir des métriques personnalisées pour les calculs de base. Pour de plus amples informations, veuillez consulter [Interrogez les agrégats de propriétés des actifs dans AWS IoT SiteWise](#).

Rubriques

- [Définir des métriques \(console\)](#)
- [Définir les métriques \(AWS CLI\)](#)

Définir des métriques (console)

Lorsque vous définissez une métrique pour un modèle d'actif dans la AWS IoT SiteWise console, vous spécifiez les paramètres suivants :

- Nom : nom de la propriété.
- Type de données : type de données de la transformation, qui peut être double ou chaîne.
- ID externe — (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- Formule — L'expression métrique. Les expressions métriques peuvent utiliser des [fonctions d'agrégation](#) pour saisir des données à partir d'une propriété pour tous les actifs associés dans une hiérarchie. Commencez à taper ou appuyez sur la flèche vers le bas pour ouvrir la fonction de saisie automatique. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#).

Important

Les métriques ne peuvent être que des propriétés de type entier, double, booléen ou chaîne. Les booléens sont convertis en 0 (faux) et 1 (vrai).

Si vous définissez des variables d'entrée de métrique dans l'expression d'une métrique, ces entrées doivent avoir le même intervalle de temps que la métrique de sortie.

Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

- Intervalle de temps : intervalle de temps métrique. AWS IoT SiteWise prend en charge les intervalles temporels suivants, où chaque intervalle commence à la fin du précédent :
 - 1 minute à 1 minute, calculée à la fin de chaque minute (00h00, 00h01, 00h02, etc.).

- 5 minutes — 5 minutes, calculées à la fin de toutes les cinq minutes à partir de l'heure (00h00, 00h05, 00h10, etc.).
- 15 minutes — 15 minutes, calculées à la fin de toutes les quinze minutes à partir de l'heure (00h00, 00h15, 00h30, etc.).
- 1 heure — 1 heure (60 minutes), calculée à la fin de chaque heure en UTC (00:00 AM, 01:00:00 AM, 02:00:00 AM, etc.).
- 1 jour — 1 jour (24 heures), calculé à la fin de chaque journée en UTC (00h00 le lundi, 00h00 le mardi, etc.).
- 1 semaine — 1 semaine (7 jours), calculée à la fin de chaque dimanche en UTC (tous les lundis à 00h00).
- Intervalle personnalisé : vous pouvez saisir n'importe quel intervalle de temps compris entre une minute et une semaine.
- Date de décalage — (Facultatif) Date de référence à partir de laquelle agréger les données.
- Temps de décalage — (Facultatif) Heure de référence à partir de laquelle agréger les données. L'heure de décalage doit être comprise entre 00:00:00 et 23:59:59.
- Fuseau horaire du décalage — (Facultatif) Fuseau horaire du décalage. S'il n'est pas spécifié, le fuseau horaire décalé par défaut est le temps universel coordonné (UTC).

Fuseaux horaires pris en charge

- (UTC+ 00:00) Heure universelle coordonnée
- (UTC+ 01:00) Heure centrale européenne
- (UTC+ 02:00) Europe de l'Est
- (UTC+03+:00) Heure de l'Afrique de l'Est
- (UTC+ 04:00) Heure du Proche-Orient
- (UTC+ 05:00) Heure de Lahore au Pakistan
- (UTC+ 05:30) Heure normale de l'Inde
- (UTC+ 06:00) Heure normale du Bangladesh
- (UTC+ 07:00) Heure normale du Vietnam
-
- (UTC+ 09:00) Heure normale du Japon
- (UTC+ 09:30) Heure centrale de l'Australie

• (UTC+ 10:00) Heure de l'Est de l'Australie

- (UTC+ 11:00) Heure normale de Salomon
- (UTC+ 12:00) Heure normale de Nouvelle-Zélande
- (UTC- 11:00) Heure des îles Midway
- (UTC- 10:00) Heure normale d'Hawaï
- (UTC- 09:00) Heure normale de l'Alaska
- (UTC- 08:00) Heure normale du Pacifique
- (UTC- 07:00) Heure normale de Phoenix
- (UTC- 06:00) Heure normale du Centre
- (UTC- 05:00) Heure normale de l'Est
- (UTC- 04:00) Heure de Porto Rico et des Îles Vierges américaines
- (UTC- 03:00) Heure normale d'Argentine
- (UTC- 02:00) Heure de Géorgie du Sud
- (UTC- 01:00) Heure de l'Afrique centrale

Exemple intervalle de temps personnalisé avec décalage (console)

L'exemple suivant vous montre comment définir un intervalle de 12 heures avec un décalage le 20 février 2021 à 18 h 30 30 (PST).

Pour définir un intervalle personnalisé avec un décalage

1. Pour Intervalle de temps, choisissez Intervalle personnalisé.
2. Pour Intervalle de temps, effectuez l'une des opérations suivantes :
 - Entrez **12**, puis choisissez les heures.
 - Entrez **720**, puis choisissez minutes.
 - Entrez **43200**, puis choisissez secondes.



Important

L'intervalle de temps doit être un entier, quelle que soit l'unité.

3. Pour Date de décalage, choisissez 2021/02/20.
4. Pour Heure de décalage, entrez **18:30:30**.

5. Pour le fuseau horaire décalé, choisissez (UTC- 08:00) Heure normale du Pacifique.

Si vous créez la métrique le 1er juillet 2021, avant ou à 18 h 30 30 (PST), vous obtenez le premier résultat d'agrégation le 1er juillet 2021 à 18 h 30 30 (PST). Le deuxième résultat d'agrégation est le 2 juillet 2021 à 06h30 (PST), et ainsi de suite.

Définir les métriques (AWS CLI)

Lorsque vous définissez une métrique pour un modèle d'actif à l'aide de l' AWS IoT SiteWise API, vous spécifiez les paramètres suivants :

- **name**— Le nom de la propriété.
- **dataType**— Le type de données de la métrique, qui peut être `DOUBLE` ou `STRING`.
- **externalId**— (Facultatif) Il s'agit d'un identifiant défini par l'utilisateur. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- **expression**— L'expression métrique. Les expressions métriques peuvent utiliser des [fonctions d'agrégation](#) pour saisir des données à partir d'une propriété pour tous les actifs associés dans une hiérarchie. Pour de plus amples informations, veuillez consulter [Utiliser des expressions de formule](#).
- **window**— L'intervalle de temps et le décalage correspondant à la fenêtre de fluctuation de la métrique, où chaque intervalle commence à la fin du précédent :
 - **interval**— L'intervalle de temps pendant lequel la fenêtre clignote. L'intervalle de temps doit être compris entre une minute et une semaine.
 - **offsets**— Le décalage dû à la fenêtre qui bascule.

Pour plus d'informations, consultez [TumblingWindow](#) dans la Référence d'API AWS IoT SiteWise .

Exemple intervalle de temps personnalisé avec un décalage (AWS CLI)

L'exemple suivant vous montre comment définir un intervalle de 12 heures avec un décalage le 20 février 2021 à 18 h 30 30 (PST).

```
{
  "window": {
    "tumbling": {
      "interval": "12h",
```



```
        "offset": " 2021-07-23T18:30:30-08"  
      }  
    }  
  }
```

Si vous créez la métrique le 1er juillet 2021, avant ou à 18 h 30 30 (PST), vous obtenez le premier résultat d'agrégation le 1er juillet 2021 à 18 h 30 30 (PST). Le deuxième résultat d'agrégation est le 2 juillet 2021 à 06h30 (PST), et ainsi de suite.

- **variables**— La liste des variables qui définit les autres propriétés de votre actif ou de vos actifs enfants à utiliser dans l'expression. Chaque structure de variable contient un nom simple à utiliser dans l'expression et une structure `value` qui identifie la propriété à lier à cette variable. La structure `value` contient les informations suivantes :
 - **propertyId**— L'ID de la propriété à partir de laquelle les valeurs doivent être extraites. Vous pouvez utiliser le nom de la propriété au lieu de son ID si la propriété est définie dans le modèle actuel (plutôt que définie dans un modèle à partir d'une hiérarchie).
 - **hierarchyId**— (Facultatif) L'ID de la hiérarchie à partir de laquelle interroger les actifs enfants de la propriété. Vous pouvez utiliser le nom de la définition de hiérarchie au lieu de son ID. Si vous omettez cette valeur, AWS IoT SiteWise recherche la propriété dans le modèle actuel.

Important

Les métriques ne peuvent être que des propriétés de type entier, double, booléen ou chaîne. Les booléens sont convertis en 0 (faux) et 1 (vrai).

Si vous définissez des variables d'entrée de métrique dans l'expression d'une métrique, ces entrées doivent avoir le même intervalle de temps que la métrique de sortie.

Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

- **unit**— (Facultatif) L'unité scientifique de la propriété, telle que mm ou Celsius.

Exemple Exemple de définition de métrique

L'exemple suivant illustre une propriété de métrique qui agrège les données de mesure de la température d'une ressource pour calculer la température horaire maximale en Fahrenheit. Cet objet est un exemple d'objet [AssetModelProperty](#) contenant une [métrique](#). Vous pouvez spécifier cet objet dans le cadre de la charge utile de la demande [CreateAssetModel](#) pour créer une propriété de métrique. Pour de plus amples informations, veuillez consulter [Création d'un modèle d'actifs \(AWS CLI\)](#).

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Max temperature",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "max(temp_f)",
          "variables": [
            {
              "name": "temp_f",
              "value": {
                "propertyId": "Temperature F"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "1h"
            }
          }
        }
      },
      "unit": "Fahrenheit"
    }
  ],
  ...
}
```

Exemple Exemple de définition de métrique qui saisit les données des actifs associés

L'exemple suivant illustre une propriété métrique qui agrège les données de puissance moyenne de plusieurs éoliennes pour calculer la puissance moyenne totale d'un parc éolien. Cet objet est un exemple d'objet [AssetModelProperty](#) contenant une [métrique](#). Vous pouvez spécifier cet objet dans le cadre de la charge utile de la demande [CreateAssetModel](#) pour créer une propriété de métrique.

```
{
  ...
  "assetModelProperty": [
    ...
    {
      "name": "Total Average Power",
      "dataType": "DOUBLE",
      "type": {
        "metric": {
          "expression": "avg(power)",
          "variables": [
            {
              "name": "power",
              "value": {
                "propertyId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
                "hierarchyId": "Turbine Asset Model"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      },
      "unit": "kWh"
    }
  ],
  ...
}
```

Utiliser des expressions de formule

Avec les expressions de formule, vous pouvez définir les fonctions mathématiques permettant de transformer et d'agréger vos données industrielles brutes afin d'obtenir des informations sur

vos opérations. Les expressions de formule combinent des littéraux, des opérateurs, des fonctions et des variables pour traiter les données. Pour plus d'informations sur la façon de définir les propriétés des actifs qui utilisent des expressions de formule, reportez-vous [Transformer les données \(transformations\)](#) aux sections et [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#). Les transformations et les métriques sont des propriétés de formule.

Rubriques

- [Utiliser des variables dans les expressions de formule](#)
- [Utiliser des littéraux dans les expressions de formule](#)
- [Utiliser des opérateurs dans les expressions de formule](#)
- [Utiliser des constantes dans les expressions de formule](#)
- [Utiliser des fonctions dans des expressions de formule](#)
- [Tutoriels d'expression de formules](#)

Utiliser des variables dans les expressions de formule

Les variables représentent les propriétés des AWS IoT SiteWise actifs dans les expressions de formule. Utilisez des variables pour saisir des valeurs provenant d'autres propriétés d'actifs dans vos expressions, afin de pouvoir traiter les données issues de propriétés constantes ([attributs](#)), de flux de données brutes ([mesures](#)) et d'autres propriétés de formule.

Les variables peuvent représenter les propriétés des actifs à partir du même modèle d'actif ou à partir de modèles d'actifs enfants associés. Seules les formules métriques peuvent saisir des variables à partir de modèles d'actifs enfants.

Vous identifiez les variables sous différents noms dans la console et dans l'API.

- AWS IoT SiteWise console — Utilisez les noms des propriétés des actifs comme variables dans vos expressions.
- AWS IoT SiteWise API (AWS CLI, AWS SDKs) — Définissez les variables avec la [ExpressionVariable](#) structure, qui nécessite un nom de variable et une référence à une propriété d'actif. Le nom de la variable peut contenir des lettres minuscules, des chiffres et des traits de soulignement. Utilisez ensuite des noms de variables pour référencer les propriétés des actifs dans vos expressions.

Les noms de variables distinguent les majuscules et minuscules.

Pour plus d'informations, consultez les [sections Définition des transformations](#) et [Définition des métriques](#).

Utiliser des variables pour référencer les propriétés

La valeur d'une variable définit la propriété à laquelle elle fait référence. AWS IoT SiteWise propose différentes manières de le faire.

- Par identifiant de propriété : vous pouvez spécifier l'identifiant unique (UUID) de la propriété pour l'identifier.
- Par nom : si la propriété utilise le même modèle d'actif, vous pouvez spécifier son nom dans le champ ID de propriété.
- Par chemin : la valeur d'une variable peut faire référence à une propriété par son chemin. Pour de plus amples informations, veuillez consulter [Utiliser des chemins pour référencer les propriétés personnalisées du modèle composite](#).

Note

Les variables ne sont pas prises en charge par AWS IoT SiteWise la console. Ils sont utilisés par AWS IoT SiteWise l'API, notamment le AWS Command Line Interface AWS CLI) et AWS SDKs.

Une variable que vous recevez dans une réponse AWS IoT SiteWise inclut des informations complètes sur la valeur, notamment l'ID et le chemin.

Toutefois, lorsque vous transmettez une variable à AWS IoT SiteWise (par exemple, lors d'un appel « créer » ou « mettre à jour »), vous n'avez qu'à en spécifier une. Par exemple, si vous spécifiez le chemin, vous n'avez pas besoin de fournir l'ID.

Utiliser des littéraux dans les expressions de formule

AWS IoT SiteWise prend en charge l'utilisation de littéraux dans les expressions et les formules. Les littéraux sont des valeurs fixes qui représentent un type de données spécifique. Dans AWS IoT SiteWise, vous pouvez définir des nombres et des littéraux de chaîne dans les expressions de formule. Les littéraux peuvent être utilisés dans différents contextes, notamment les transformations de données, les conditions d'alarme et les calculs de visualisation.

- Chiffres

Utilisez les nombres et la notation scientifique pour définir des nombres entiers et des doubles. Vous pouvez utiliser la [notation E](#) pour exprimer des nombres en notation scientifique.

Exemples : 12.0,.9,-23.1,7.89e3, 3.4E-5

- Chaînes

Utilisez les caractères ' (guillemets) et " (guillemets doubles) pour définir les chaînes. Le type de devis pour le début et la fin doit correspondre. Pour éviter un guillemet correspondant à celui que vous utilisez pour déclarer une chaîne, incluez ce guillemet deux fois. Il s'agit du seul caractère d'échappement dans AWS IoT SiteWise les chaînes.

Exemples : 'active', "inactive", '{"temp": 52}', "{\"temp\": \"high\"}"

Utiliser des opérateurs dans les expressions de formule

Vous pouvez utiliser les opérateurs courants suivants dans les expressions de formule.

Opérateur	Description
+	Si les deux opérandes sont des nombres, cet opérateur ajoute les opérandes gauche et droit. Si l'un des opérandes est une chaîne, cet opérateur concatène les opérandes gauche et droit sous forme de chaînes. Par exemple, l'expression est 1 + 2 + " is three" évaluée à. "3 is three" La chaîne concaténée peut comporter jusqu'à 1024 caractères. Si la chaîne dépasse 1024 caractères, elle AWS IoT SiteWise ne produit aucun point de données pour ce calcul.
-	Il soustrait l'opérande droit de l'opérande gauche.

Opérateur	Description
	Vous ne pouvez utiliser cet opérateur qu'avec des opérandes numériques.
/	Il divise l'opérande gauche par l'opérande droit. Vous ne pouvez utiliser cet opérateur qu'avec des opérandes numériques.
*	Il multiplie les opérandes gauche et droit. Vous ne pouvez utiliser cet opérateur qu'avec des opérandes numériques.
^	Il élève l'opérande gauche à la puissance de l'opérande droit (élévation de la puissance). Vous ne pouvez utiliser cet opérateur qu'avec des opérandes numériques.
%	Il renvoie le reste résultant de la division de l'opérande gauche par l'opérande droit. Le résultat a le même signe que l'opérande gauche. Ce comportement est différent de celui de l'opération modulo. Vous ne pouvez utiliser cet opérateur qu'avec des opérandes numériques.
$x < y$	Renvoie 1 si la valeur x est inférieure à y, sinon 0.
$x > y$	Renvoie 1 si la valeur x est supérieure à y, sinon 0.
$x \leq y$	Renvoie 1 si x est inférieur ou égal à y, sinon 0.
$x \geq y$	Renvoie 1 si la valeur x est supérieure ou égale à y, sinon 0.

Opérateur	Description
<code>x == y</code>	Renvoie 1 si x c'est égal à y, sinon 0.
<code>x != y</code>	Renvoie 1 si x ce n'est pas égal à y, sinon 0.
<code>!x</code>	Renvoie 1 si la valeur x est évaluée à 0 (faux), sinon 0. x est évalué à faux si : • x est un opérande numérique évalué à 0 • x est évalué à une chaîne vide. • x est évalué à un tableau vide. • x est évalué à None.
<code>x and y</code>	Renvoie 0 si la valeur x est évaluée à 0 (faux). Dans le cas contraire, renvoie le résultat évalué de y. x ou y est évalué à faux si : • x ou y est un opérande numérique et il est évalué à 0 • x ou y est évalué à une chaîne vide. • x ou y est évalué selon un tableau vide. • x ou y est évalué à None.

Opérateur	Description
<code>x or y</code>	Renvoie 1 si la valeur x est évaluée à 1 (vrai). Dans le cas contraire, renvoie le résultat évalué de y. x ou y est évalué à faux si : • x ou y est un opérande numérique et il est évalué à 0 • x ou y est évalué à une chaîne vide. • x ou y est évalué selon un tableau vide. • x ou y est évalué à None.
<code>not x</code>	Renvoie 1 si la valeur x est évaluée à 0 (faux), sinon 0. x est évalué à faux si : • x est un opérande numérique évalué à 0 • x est évalué à une chaîne vide. • x est évalué à un tableau vide. • x est évalué à None.
<code>[]</code> <code>s[index]</code>	Renvoie le caractère à un index index de la chaînes. Ceci est équivalent à la syntaxe d'index en Python. Exemple Exemples • "Hello!"[1] renvoie e. • "Hello!"[-2] renvoie o.

Opérateur	Description
[] s[start:end:step]	Renvoie une tranche de la chaînes. Ceci est équivalent à la syntaxe des tranches en Python. Cet opérateur possède les arguments suivants : • <code>start</code>— (Facultatif) L'indice de début inclus de la tranche. La valeur par défaut est 0. • <code>end</code>— (Facultatif) Indice final exclusif de la tranche. La valeur par défaut est la longueur de la chaîne. • <code>step</code>— (Facultatif) Le nombre à incrémenter pour chaque étape de la tranche. Par exemple, vous pouvez spécifier 2 de renvoyer une tranche avec tous les autres caractères ou -1 d'inverser la tranche. La valeur par défaut est 1. Vous pouvez omettre l'<code>step</code> argument pour utiliser sa valeur par défaut. Par exemple, <code>s[1:4:1]</code> équivaut à <code>s[1:4]</code>. Les arguments doivent être des entiers ou la constante none. Si vous le spécifiez <code>none</code>, AWS IoT SiteWise utilise la valeur par défaut pour cet argument. Exemple Exemples • <code>"Hello!"[1:4]</code> renvoie <code>"ell"</code>. • <code>"Hello!"[:2]</code> renvoie <code>"He"</code>. • <code>"Hello!"[3:]</code> renvoie <code>"lo!"</code>. • <code>"Hello!"[:-4]</code> renvoie <code>"He"</code>. • <code>"Hello!"[::2]</code> renvoie <code>"Hlo"</code>. • <code>"Hello!"[::-1]</code> renvoie <code>"!olleH"</code>.

Utiliser des constantes dans les expressions de formule

Dans AWS IoT SiteWise, vous pouvez utiliser des constantes dans vos expressions et formules pour représenter des valeurs fixes ou des paramètres prédéfinis. Les constantes peuvent être utilisées dans différents contextes, tels que les transformations de données, les conditions d'alarme ou les calculs de visualisation. En utilisant des constantes, vous pouvez simplifier vos expressions et les rendre plus lisibles et faciles à gérer.

Vous pouvez utiliser les constantes mathématiques courantes suivantes dans vos expressions. Toutes les constantes ne distinguent pas les majuscules et minuscules.

Note

Si vous définissez une variable portant le même nom qu'une constante, la variable remplace la constante.

Constant	Description
pi	Le nombre pi (π) : 3.141592653589793
e	Le chiffre e : 2.718281828459045
true	Équivalent au chiffre 1. Dans AWS IoT SiteWise, les booléens sont convertis en équivalents numériques.
false	Équivalent au chiffre 0. Dans AWS IoT SiteWise, les booléens sont convertis en équivalents numériques.
none	C'est équivalent à aucune valeur. Vous pouvez utiliser cette constante pour ne rien afficher à la suite d'une expression conditionnelle .

Utiliser des fonctions dans des expressions de formule

Vous pouvez utiliser les fonctions suivantes pour agir sur les données de vos expressions de formule.

Les transformations et les métriques prennent en charge différentes fonctions. Le tableau suivant indique les types de fonctions compatibles avec chaque type de propriété de formule.

 Note

Vous pouvez inclure un maximum de 10 fonctions dans une expression de formule.

Type de fonction	Transformations	Métriques
Utiliser des fonctions courantes dans les expressions de formule	 Oui	 Oui
Utiliser les fonctions de comparaison dans les expressions de formule	 Oui	 Oui
Utiliser des fonctions conditionnelles dans les expressions de formule	 Oui	 Oui
Utiliser des fonctions de chaîne dans les expressions de formule	 Oui	 Oui
Utiliser les fonctions d'agrégation dans les expressions de formule	 Non	 Oui

Type de fonction	Transformations	Métriques
Utiliser des fonctions temporelles dans les expressions de formules	 Oui	 Oui
Utiliser les fonctions de date et d'heure dans les expressions de formule	 Oui	 Oui

Syntaxe des fonctions

Vous pouvez utiliser la syntaxe suivante pour créer des fonctions :

Syntaxe régulière

Avec la syntaxe normale, le nom de la fonction est suivi de parenthèses contenant zéro argument ou plus.

function_name(argument1, argument2, argument3, ...). Par exemple, les fonctions dont la syntaxe est normale peuvent ressembler à `log(x)` et `contains(s, substring)`.

Syntaxe uniforme des appels de fonction (UFCS)

L'UFCS vous permet d'appeler des fonctions en utilisant la syntaxe des appels de méthode dans la programmation orientée objet. Avec UFCS, le premier argument est suivi par point (`.`), puis le nom de la fonction et les autres arguments (le cas échéant) entre parenthèses.

argument1.function_name(argument2, argument3, ...). Par exemple, les fonctions associées à l'UFCS peuvent ressembler à `x.log()` et `s.contains(substring)`.

Vous pouvez également utiliser l'UFCS pour enchaîner les fonctions suivantes. AWS IoT SiteWise utilise le résultat de l'évaluation de la fonction en cours comme premier argument de la fonction suivante.

Par exemple, vous pouvez utiliser à la place `message.jp('$ .status').lower().contains('fail')` `decontains(lower(jp(message, '$.status')), 'fail')`.

Pour plus d'informations, consultez le site Web du [langage de programmation D](#).

 Note

Vous pouvez utiliser l'UFCS pour toutes les AWS IoT SiteWise fonctions. AWS IoT SiteWise les fonctions ne distinguent pas les majuscules et minuscules. Par exemple, vous pouvez utiliser `lower(s)` et `Lower(s)` manière interchangeable.

Utiliser des fonctions courantes dans les expressions de formule

Dans les [transformations](#) et [les métriques](#), vous pouvez utiliser les fonctions suivantes pour calculer les fonctions mathématiques courantes dans les transformations et les métriques.

Fonction	Description
<code>abs(x)</code>	Renvoie la valeur absolue de x.
<code>acos(x)</code>	Renvoie l'arc cosinus de x.
<code>asin(x)</code>	Renvoie l'arc sinus de x.
<code>atan(x)</code>	Renvoie l'arc tangent de x.
<code>cbrt(x)</code>	Renvoie la racine cubique de x.
<code>ceil(x)</code>	Renvoie l'entier le plus proche supérieur à x.
<code>cos(x)</code>	Renvoie le cosinus de x.
<code>cosh(x)</code>	Renvoie le cosinus hyperbolique de x.
<code>cot(x)</code>	Renvoie la cotangente de x.
<code>exp(x)</code>	Renvoie e à la puissance de x.
<code>expm1(x)</code>	Renvoie $\exp(x) - 1$. Utilisez cette fonction pour calculer avec plus de précision $\exp(x) - 1$ les petites valeurs de x.

Fonction	Description
<code>floor(x)</code>	Renvoie l'entier le plus proche inférieur à x.
<code>log(x)</code>	Renvoie l'élément $\log_e$ (base e) de x.
<code>log10(x)</code>	Renvoie l'élément $\log_{10}$ (base 10) de x.
<code>log1p(x)</code>	Renvoie $\log(1 + x)$. Utilisez cette fonction pour calculer avec plus de précision $\log(1 + x)$ les petites valeurs de x.
<code>log2(x)</code>	Renvoie l'élément $\log_2$ (base 2) de x.
<code>pow(x, y)</code>	Renvoie x à la puissance de y. Cela équivaut à x^y .
<code>signum(x)</code>	Renvoie le signe de x (-1 pour les entrées négatives, 0 pour les entrées nulles et +1 pour les entrées positives).
<code>sin(x)</code>	Renvoie le sinus de x.
<code>sinh(x)</code>	Renvoie le sinus hyperbolique de x.
<code>sqrt(x)</code>	Renvoie la racine carrée de x.
<code>tan(x)</code>	Renvoie la tangente de x.
<code>tanh(x)</code>	Renvoie la tangente hyperbolique de x.

Utiliser les fonctions de comparaison dans les expressions de formule

Dans les [transformations](#) et [les métriques](#), vous pouvez utiliser les fonctions de comparaison suivantes pour comparer deux valeurs et obtenir un résultat 1 (vrai) ou 0 (faux). AWS IoT SiteWise compare les chaînes par ordre [lexicographique](#).

Fonction	Description
<code>gt(x, y)</code>	Renvoie 1 si x est supérieur à y, sinon 0 ($x > y$). Cette fonction ne renvoie pas de valeur si x les types y sont incompatibles, tels qu'un nombre et une chaîne.
<code>gte(x, y)</code>	Renvoie 1 si x est supérieur ou égal à y, sinon 0 ($x \geq y$). AWS IoT SiteWise considère que les arguments sont égaux s'ils se situent dans une tolérance relative de $1E-9$. Cela se comporte de la même manière que la fonction isclose en Python. Cette fonction ne renvoie pas de valeur si x les types y sont incompatibles, tels qu'un nombre et une chaîne.
<code>eq(x, y)</code>	Renvoie 1 si x est égal à y, sinon 0 ($x == y$). AWS IoT SiteWise considère que les arguments sont égaux s'ils se situent dans une tolérance relative de $1E-9$. Cela se comporte de la même manière que la fonction isclose en Python. Cette fonction ne renvoie pas de valeur si x les types y sont incompatibles, tels qu'un nombre et une chaîne.
<code>lt(x, y)</code>	Renvoie 1 si x est inférieur à y, sinon 0 ($x < y$).

Fonction	Description
	Cette fonction ne renvoie pas de valeur si x les types y sont incompatibles, tels qu'un nombre et une chaîne.
<code>lte(x, y)</code>	Renvoie 1 si x est inférieur ou égal à y, sinon 0 ($x \leq y$). AWS IoT SiteWise considère que les arguments sont égaux s'ils se situent dans une tolérance relative de $1E-9$. Cela se comporte de la même manière que la fonction isclose en Python. Cette fonction ne renvoie pas de valeur si x les types y sont incompatibles, tels qu'un nombre et une chaîne.
<code>isnan(x)</code>	Renvoie 1 si x c'est égal à NaN, sinon 0. Cette fonction ne renvoie pas de valeur s'il s'agit d'une chaîne.

Utiliser des fonctions conditionnelles dans les expressions de formule

Dans les [transformations](#) et [les métriques](#), vous pouvez utiliser la fonction suivante pour vérifier une condition et renvoyer différents résultats, que la condition soit vraie ou fausse.

Fonction	Description
<code>if(condition, result_if_true, result_if_false)</code>	Évalue <code>condition</code> et renvoie <code>result_if_true</code> si la condition est vraie ou <code>result_if_false</code> si la condition est évaluée à <code>false</code>. <code>condition</code> doit être un chiffre. Cette fonction considère <code>0</code> une chaîne vide comme <code>false</code> et tout le reste (y compris NaN) comme <code>true</code>. Les booléens sont convertis en <code>0</code> (faux) et <code>1</code> (vrai).

Fonction	Description
	Vous pouvez renvoyer la constante none à partir de cette fonction pour supprimer la sortie pour une condition particulière. Cela signifie que vous pouvez filtrer les points de données qui ne répondent pas à une condition. Pour de plus amples informations, veuillez consulter Filtrer les points de données. Exemple Exemples • <code>if(0, x, y)</code> renvoie la variable. • <code>if(5, x, y)</code> renvoie la variable <code>x</code>. • <code>if(gt(temp, 300), x, y)</code> renvoie la variable <code>x</code> si la variable <code>temp</code> est supérieure à 300. • <code>if(gt(temp, 300), temp, none)</code> renvoie la variable <code>temp</code> si elle est supérieure ou égale à 300, ou <code>none</code> (aucune valeur) si elle <code>temp</code> est inférieure à 300. Nous vous recommandons d'utiliser l'UFCS pour les fonctions conditionnelles imbriquées dans lesquelles un ou plusieurs arguments sont des fonctions conditionnelles. Vous pouvez utiliser <code>if(condition, result_if_true)</code> pour évaluer une condition et <code>elif(condition, result_if_true, result_if_false)</code> pour évaluer des conditions supplémentaires. Par exemple, vous pouvez utiliser à la <pre>if(condition1, result1_if_true).e lif(condition2, result2_if_true, result2_if_false) place de if(condit ion1, result1_if_true, if(condit</pre>

Fonction	Description
	ion2, result2_if_true, result2_if_false)) . Vous pouvez également enchaîner des fonctions conditionnelles intermédiaires supplémentaires. Par exemple, vous pouvez utiliser plusieurs if instructions if(condition1, result1_if_true).elif(condition2, result2_if_true).elif(condition3, result3_if_true, result3_if_false) au lieu de les imbriquer, telles que if(condition1, result1_if_true, if(condition2, result2_if_true, if(condition3, result3_if_true result3_if_false))) .
	 Important Vous devez l'utiliser elif(condition, result_if_true, result_if_false) avec l'UFCS.

Utiliser des fonctions de chaîne dans les expressions de formule

Dans les [transformations](#) et [les métriques](#), vous pouvez utiliser les fonctions suivantes pour agir sur des chaînes. Pour de plus amples informations, veuillez consulter [Utiliser des chaînes dans les formules](#).

Important

Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne

doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

Fonction	Description
<code>len(s)</code>	Renvoie la longueur de la chaîne.
<code>find(s, substring)</code>	Renvoie l'index de la chaîne <code>substring</code> dans la chaîne.
<code>contains(s, substring)</code>	Renvoie 1 si la chaîne <code>s</code> contient la chaîne <code>substring</code> , sinon 0.
<code>upper(s)</code>	Renvoie la chaîne <code>s</code> en majuscules.
<code>lower(s)</code>	Renvoie la chaîne <code>s</code> en minuscules.
<code>jp(s, json_path)</code>	Évalue la chaîne <code>s</code> avec l'JsonPath expression <code>json_path</code> et renvoie le résultat. Utilisez cette fonction pour effectuer les opérations suivantes : • Extrayez une valeur, un tableau ou un objet d'une structure JSON sérialisée. • Convertit une chaîne en nombre. Par exemple, la formule est <code>jp('111', '\$')</code> renvoyée 111 sous forme de nombre. Pour extraire une valeur de chaîne d'une structure JSON et la renvoyer sous forme de nombre, vous devez utiliser plusieurs <code>jp</code> fonctions imbriquées. La <code>jp</code> fonction externe extrait la chaîne de la structure JSON et la <code>jp</code> fonction interne convertit la chaîne en nombre.

Fonction	Description
	La chaîne <code>json_path</code> doit contenir une chaîne littérale. Cela signifie qu'il ne <code>json_path</code> peut pas s'agir d'une expression évaluée en chaîne. Exemple Exemples • <code>jp('{"status":"active","value":15}', '\$.value')</code> renvoie 15. • <code>jp('{"measurement":{"reading":25,"confidence":0.95}}', '\$.measurement.reading')</code> renvoie 25. • <code>jp('[2,8,23]', '\$[2]')</code> renvoie 23. • <code>jp('{"values":[3,6,7]}', '\$.values[1]')</code> renvoie 6. • <code>jp('111', '\$')</code> renvoie 111. • <code>jp(jp('{"measurement":{"reading":25,"confidence":"0.95"}}', '\$.measurement.confidence'), '\$')</code> renvoie 0.95.
<code>join(s0, s1, s2, s3, ...)</code>	Renvoie une chaîne concaténée avec un délimiteur. Cette fonction utilise la première chaîne d'entrée comme délimiteur et réunit les chaînes d'entrée restantes. Cela se comporte de la même manière que la fonction join (CharSequence delimiter, CharSequence... elements) en Java. Exemple Exemples • <code>join("-", "aa", "bb", "cc")</code> retourne <code>aa-bb-cc</code>

Fonction	Description
<code>format(expression: "format")</code> ou <code>format("format", expression)</code>	Renvoie une chaîne au format spécifié. Cette fonction donne <code>expression</code> une valeur, puis renvoie la valeur dans le format spécifié. Cela se comporte de la même manière que la fonction format (String format, Object... args) en Java. Pour plus d'informations sur les formats pris en charge, consultez la section Conversions sous Class Formatter dans la plate-forme Java, spécification de l'API Standard Edition 7. Exemple Exemples • <code>format(100+1: "d")</code> renvoie une chaîne,101. • <code>format("The result is %d", 100+1)</code>renvoie une chaîne,The result is 101.

Fonction	Description
<code>f'expression'</code>	Renvoie une chaîne concaténée. Avec cette fonction formatée, vous pouvez utiliser une expression simple pour concaténer et formater des chaînes. Ces fonctions peuvent contenir des expressions imbriquées. Vous pouvez utiliser <code>{}</code> (accolades) pour interpoler des expressions. Cela se comporte de la même manière que les littéraux de chaîne formatés en Python. Exemple Exemples <code>f'abc{1+2: "f"}d'</code> renvoie <code>abc3.000000d</code> . Pour évaluer cet exemple d'expression, procédez comme suit : <code>format(1+2: "f")</code> renvoie un nombre à virgule flottante, <code>3.000000</code>. <code>join(' ', "abc", 1+2, 'd')</code> renvoie une chaîne, <code>abc3.000000d</code> . Vous pouvez également écrire l'expression de la manière suivante : <code>join(' ', "abc", format(1+2: "f"), 'd')</code> .

Utiliser les fonctions d'agrégation dans les expressions de formule

Dans [les métriques](#) uniquement, vous pouvez utiliser les fonctions suivantes pour agréger les valeurs d'entrée sur chaque intervalle de temps et calculer une valeur de sortie unique. Les fonctions d'agrégation peuvent agréger les données issues de ressources associées.

Les arguments des fonctions d'agrégation peuvent être [des variables](#), [des littéraux numériques](#), [des fonctions temporelles](#), des expressions imbriquées ou des fonctions d'agrégation. La formule `max(latest(x), latest(y), latest(z))` utilise une fonction d'agrégation comme argument et renvoie la plus grande valeur actuelle des `z` propriétés `xy`, et.

Vous pouvez utiliser des expressions imbriquées dans les fonctions d'agrégation. Lorsque vous utilisez des expressions imbriquées, les règles suivantes s'appliquent :

- Chaque argument ne peut comporter qu'une seule variable.

Exemple

Par exemple, $\text{avg}(x * (x - 1))$ et $\text{sum}(x/2) / \text{avg}(y^2)$ sont pris en charge.

Par exemple, $\text{min}(x/y)$ n'est pas pris en charge.

- Chaque argument peut comporter des expressions imbriquées à plusieurs niveaux.

Exemple

Par exemple, $\text{sum}(\text{avg}(x^2)/2)$ est pris en charge.

- Les différents arguments peuvent avoir des variables différentes.

Exemple

Par exemple, $\text{sum}(x/2, y^2)$ est pris en charge.

Note

- Si vos expressions contiennent des mesures, AWS IoT SiteWise utilise les dernières valeurs de l'intervalle de temps actuel pour que les mesures calculent les agrégats.
- Si vos expressions contiennent des attributs, AWS IoT SiteWise utilise les dernières valeurs des attributs pour calculer les agrégats.

Fonction	Description
$\text{avg}(x_0, \dots, x_n)$	Renvoie la moyenne des valeurs des variables données sur l'intervalle de temps actuel. Cette fonction produit un point de données uniquement si les variables données ont au moins un point de données sur l'intervalle de temps actuel.

Fonction	Description
$\text{sum}(x_0, \dots, x_n)$	Renvoie la somme des valeurs de variables données sur l'intervalle de temps actuel. Cette fonction produit un point de données uniquement si les variables données ont au moins un point de données sur l'intervalle de temps actuel.
$\text{min}(x_0, \dots, x_n)$	Renvoie la valeur minimale des valeurs des variables données sur l'intervalle de temps actuel. Cette fonction produit un point de données uniquement si les variables données ont au moins un point de données sur l'intervalle de temps actuel.
$\text{max}(x_0, \dots, x_n)$	Renvoie la valeur maximale des valeurs de variables données sur l'intervalle de temps actuel. Cette fonction produit un point de données uniquement si les variables données ont au moins un point de données sur l'intervalle de temps actuel.
$\text{count}(x_0, \dots, x_n)$	Renvoie le nombre total de points de données pour les variables données sur l'intervalle de temps actuel. Pour de plus amples informations sur la comptabilisation du nombre de points de données qui répondent à une condition, veuillez consulter Compter les points de données qui correspondent à une condition. Cette fonction calcule un point de données pour chaque intervalle de temps.

Fonction	Description
<code>stdev($x_0, \dots, x_n$)</code>	Renvoie l'écart type des valeurs des variables données sur l'intervalle de temps actuel. Cette fonction produit un point de données uniquement si les variables données ont au moins un point de données sur l'intervalle de temps actuel.

Utiliser des fonctions temporelles dans les expressions de formules

Utilisez des fonctions temporelles pour renvoyer des valeurs basées sur les horodatages des points de données.

Utiliser des fonctions temporelles dans les métriques

Dans [les métriques](#) uniquement, vous pouvez utiliser les fonctions suivantes qui renvoient des valeurs basées sur l'horodatage des points de données.

Les arguments des fonctions temporelles doivent être des propriétés du modèle d'actif local ou des expressions imbriquées. Cela signifie que vous ne pouvez pas utiliser les propriétés des modèles d'actifs enfants dans les fonctions temporelles.

Vous pouvez utiliser des expressions imbriquées dans les fonctions temporelles. Lorsque vous utilisez des expressions imbriquées, les règles suivantes s'appliquent :

- Chaque argument ne peut comporter qu'une seule variable.

Par exemple, `latest( t*9/5 + 32 )` est pris en charge.

- Les arguments ne peuvent pas être des fonctions d'agrégation.

Par exemple, `first( sum(x) )` n'est pas pris en charge.

Fonction	Description
<code>first(x)</code>	Renvoie la valeur de la variable donnée avec l'horodatage le plus ancien sur l'intervalle de temps actuel.
<code>last(x)</code>	Renvoie la valeur de la variable donnée avec l'horodatage le plus récent sur l'intervalle de temps actuel.
<code>earliest(x)</code>	Renvoie la dernière valeur de la variable donnée avant le début de l'intervalle de temps actuel. Cette fonction calcule un point de données pour chaque intervalle de temps, si la propriété d'entrée a au moins un point de données dans son historique. Consultez time-range-defintion pour plus de détails.
<code>latest(x)</code>	Renvoie la dernière valeur de la variable donnée avec le dernier horodatage avant la fin de l'intervalle de temps actuel. Cette fonction calcule un point de données pour chaque intervalle de temps, si la propriété d'entrée a au moins un point de données dans son historique. Consultez time-range-defintion pour plus de détails.
<code>statetime(x)</code>	Renvoie la durée, en secondes, pendant laquelle les variables données sont positives sur l'intervalle de temps actuel. Vous pouvez utiliser les fonctions de comparaison pour créer une propriété de transformation pour la <code>statetime</code> fonction à utiliser.

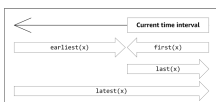
Fonction	Description
	Par exemple, si vous avez une propriété Idle qui est 0 ou 1, vous pouvez calculer le temps d'inactivité par intervalle de temps avec l'expression suivante : <code>IdleTime = statetime(Idle)</code> . Pour plus d'informations, consultez l'exemple de scénario d'heure d'état. Cette fonction ne prend pas en charge les propriétés de métriques en tant que variables d'entrée. Cette fonction calcule un point de données pour chaque intervalle de temps, si la propriété d'entrée a au moins un point de données dans son historique.

Fonction	Description
<code>TimeWeightedAvg(x, [interpolation])</code>	Renvoie la moyenne des données d'entrée pondérées par les intervalles de temps entre les points. Voir Paramètres des fonctions pondérées dans le temps pour plus de détails sur les calculs et les intervalles. L'argument facultatif <code>interpolation</code> doit être une constante de chaîne : • <code>locf</code>— Il s'agit de la valeur par défaut. Le calcul utilise l'algorithme de calcul du dernier report observé pour les intervalles entre les points de données. Dans cette approche, le point de données est calculé comme la dernière valeur observée jusqu'au prochain horodatage du point de données d'entrée. La valeur après un point de données valide est extrapolée sous forme de valeur jusqu'à l'horodatage du point de données suivant. • <code>linear</code>— Le calcul utilise l'algorithme d'interpolation linéaire pour les intervalles entre les points de données. La valeur entre deux bons points de données est extrapolée sous forme d'interpolation linéaire entre les valeurs de ces points de données. La valeur entre les bons et les mauvais points de données ou la valeur après le dernier bon point de données sera extrapolée en tant que bon point de données.

Fonction	Description
<code>TimeWeightedStDev(x, [algo])</code>	Renvoie l'écart type des données d'entrée pondéré avec les intervalles de temps entre les points. Voir Paramètres des fonctions pondérées dans le temps pour plus de détails sur les calculs et les intervalles. Le calcul utilise l'algorithme de calcul du dernier report observé pour les intervalles entre les points de données. Dans cette approche, le point de données est calculé comme la dernière valeur observée jusqu'au prochain horodatage du point de données d'entrée. Le poids est calculé sous la forme d'un intervalle de temps en secondes entre les points de données ou les limites des fenêtres. L'argument facultatif <code>algo</code> doit être une constante de chaîne : • <code>f</code>— Il s'agit de la valeur par défaut. Il renvoie une variance d'échantillon pondérée non biaisée avec des poids de fréquence, <code>TimeWeight</code> calculée en secondes. Cet algorithme est généralement supposé en fonction de l'écart type et est connu sous le nom de correction de Bessel de l'écart type pour les échantillons pondérés. • <code>p</code>— Renvoie la variance d'échantillon pondérée biaisée, également appelée variance de population. Les formules suivantes sont utilisées pour le calcul où :

Fonction	Description
	• S_p = écart type de la population • S_f = écart type de fréquence • X_i = données entrantes • ω_i = poids égal à l'intervalle de temps en secondes • μ^* = moyenne pondérée des données entrantes Équation pour l'écart type de la population : $S_p^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i}$ Équation pour l'écart type de fréquence : $S_f^2 = \frac{\sum_{i=1}^N \omega_i (x_i - \mu^*)^2}{\sum_{i=1}^N \omega_i - 1}$

Le schéma suivant montre comment AWS IoT SiteWise les fonctions temporelles `first`, `last`, `earliest` et `latest`, et sont calculées par rapport à l'intervalle de temps actuel.



Note

- L'intervalle de temps pour `first(x)`, `last(x)` est (début de la fenêtre actuelle, fin de la fenêtre actuelle].
- L'intervalle de temps pour `latest(x)` est [début du temps, fin de la fenêtre actuelle].
- La plage de temps pour `earliest(x)` est (début de l'heure, fin de la fenêtre précédente).

Paramètres des fonctions pondérés dans le temps

Les fonctions pondérées dans le temps calculées pour la fenêtre d'agrégation prennent en compte les éléments suivants :

- Points de données à l'intérieur de la fenêtre
- Intervalles de temps entre les points de données
- Dernier point de données avant la fenêtre
- Premier point de données après la fenêtre (pour certains algorithmes)

Termes :

- Point de données incorrect : tout point de données dont la qualité n'est pas bonne ou dont la valeur n'est pas numérique. Cela n'est pas pris en compte dans le calcul des résultats d'une fenêtre.
- Intervalle incorrect — Intervalle après un point de données erroné. L'intervalle avant le premier point de données connu est également considéré comme un mauvais intervalle.
- Bon point de données — Tout point de données de bonne qualité et de bonne valeur numérique.

Note

- AWS IoT SiteWise ne consomme des données GOOD de qualité que lorsqu'il calcule les transformations et les métriques. Il ignore les points UNCERTAIN de BAD données.
- L'intervalle avant le premier point de données connu est considéré comme un mauvais intervalle. Pour plus d'informations, consultez [the section called "Tutoriels d'expression de formules"](#).

L'intervalle après le dernier point de données connu se poursuit indéfiniment, affectant toutes les fenêtres suivantes. Lorsqu'un nouveau point de données arrive, la fonction recalcule l'intervalle.

Conformément aux règles ci-dessus, le résultat agrégé des fenêtres est calculé et limité aux limites des fenêtres. Par défaut, la fonction envoie le résultat de la fenêtre uniquement si l'ensemble de la fenêtre correspond à un bon intervalle.

Si l'intervalle de validité de la fenêtre est inférieur à la longueur de la fenêtre, la fonction n'envoie pas la fenêtre.

Lorsque les points de données affectant le résultat de la fenêtre changent, la fonction recalcule la fenêtre, même si les points de données se trouvent en dehors de la fenêtre.

Si la propriété d'entrée comporte au moins un point de données dans son historique et qu'un calcul a été lancé, la fonction calcule les fonctions d'agrégation pondérées dans le temps pour chaque intervalle de temps.

Exemple Exemple de scénario statetime

Prenons un exemple où vous avez une ressource avec les propriétés suivantes :

- **Idle**— Une mesure qui est 0 ou 1. Lorsque la valeur est 1, la machine est inactive.
- **Idle Time**— Mesure qui utilise la formule `statetime(Idle)` pour calculer la durée en secondes pendant laquelle la machine est inactive, par intervalle d'une minute.

La propriété **Idle** possède les points de données suivants.

Horodatage	2:00:00 PM	2:00:30 PM	2:01:15 PM	2:02:45 PM	2:04:00 PM
Idle	0	1	1	0	0

AWS IoT SiteWise calcule la **Idle Time** propriété toutes les minutes à partir des valeurs de **Idle**. Une fois ce calcul terminé, la propriété **Idle Time** possède les points de données suivants.

Horodatage	2:00:00 PM	2:01:00 PM	2:02:00 PM	2:03:00 PM	2:04:00 PM
Idle Time	N/A	30	60	45	0

AWS IoT SiteWise effectue les calculs suivants **Idle Time** à la fin de chaque minute.

- À 2:00 PM (pour 1:59 PM à 2:00 PM)
 - Il n'y a pas de données pour **Idle** avant 2:00 PM, donc aucun point de données n'est calculé.
- À 2:01 PM (pour 2:00 PM à 2:01 PM)
 - À 2:00:00 PM, la machine est active (**Idle** est égal à 0).
 - À 2:00:30 PM, la machine es inactive (**Idle** est égal à 1).

- Idle ne change plus avant la fin de l'intervalle à 2:01:00 PM, Idle Time est donc égal à 30 secondes.
- À 2:02 PM (pour 2:01 PM à 2:02 PM)
 - À 2:01:00 PM, la machine est inactive (conformément au dernier point de données à 2:00:30 PM).
 - À 2:01:15 PM, la machine est toujours inactive.
 - Idle ne change plus non plus avant la fin de l'intervalle à 2:02:00 PM, Idle Time est donc égal à 60 secondes.
- À 2:03 PM (pour 2:02 PM à 2:03 PM)
 - À 2:02:00 PM, la machine est inactive (conformément au dernier point de données à 2:01:15 PM).
 - À 2:02:45 PM, la machine est active.
 - Idle ne change plus non plus avant la fin de l'intervalle à 2:03:00 PM, Idle Time est donc égal à 45 secondes.
- À 2:04 PM (pour 2:03 PM à 2:04 PM)
 - À 2:03:00 PM, la machine est active (conformément au dernier point de données à 2:02:45 PM).
 - Idle ne change plus non plus avant la fin de l'intervalle à 2:04:00 PM, Idle Time est donc égal à 0 seconde.

Exemple Exemple TimeWeightedAvg et TimeWeightedStDev scénario

Les tableaux suivants fournissent des exemples d'entrées et de sorties pour ces métriques de fenêtre d'une minute :Avg(x), TimeWeightedAvg(x), TimeWeightedAvg(x, "linear"), stDev(x), timeWeightedStDev(x), timeWeightedStDev(x, 'p').

Exemple de saisie pour une fenêtre agrégée d'une minute :

Note

Ces points de données sont tous GOOD de qualité.

03:00:00	4.0
03:01:00	2.0

03:01:10	8.0
03:01:50	20.0
03:02:00	14,0
03:02:05	10,0
03:02:10	3.0
03:02:30	20.0
03:03:30	0.0

Résultats agrégés en sortie :

 **Note**

Aucun — Résultat non produit pour cette fenêtre.

Heure	Avg(x)	TimeWeigh tedAvg(x)	TimeWeigh tedAvg(X, "linear")	stDev(X)	timeWeigh tedStDev(x)	timeWeigh tedStDev(x, 'p')
3:00:00	4	Aucun	Aucun	0	Aucun	Aucun
3:01:00	2	4	3	0	0	0
3:02:00	14	9	13	6	5,4306100 41581775	5,3851648 07134504
3:03:00	11	13	12,875	8,5440037 4531753	7,7240544 37220943	7,6594168 62050705
3:04:00	0 USD	10	2,5	0	10,084389 681792215	10

Heure	Avg(x)	TimeWeightedAvg(x)	TimeWeightedAvg(X, "linear")	stDev(X)	timeWeightedStDev(x)	timeWeightedStDev(x, 'p')
3:05:00	Aucun	0	0	Aucun	0	0

Utiliser des fonctions temporelles dans les transformations

Dans les [transformations](#) uniquement, vous pouvez utiliser la `pretrigger()` fonction pour récupérer la valeur de GOOD qualité d'une variable avant la mise à jour de propriété qui a lancé le calcul de transformation en cours.

Prenons l'exemple d'un fabricant AWS IoT SiteWise qui surveille l'état d'une machine. Le fabricant utilise les mesures et transformations suivantes pour représenter le processus :

- Une mesure `current_state`, qui peut être 0 ou 1.
 - Si la machine est en état de nettoyage, `current_state` égal à 1.
 - Si la machine est en cours de fabrication, `current_state` est égal à 0.
- Une transformation `cleaning_state_duration`, c'est égal à `çaif(pretrigger(current_state) == 1, timestamp(current_state) - timestamp(pretrigger(current_state)), none)`. Cette transformation renvoie la durée pendant laquelle la machine est restée en état de nettoyage en secondes, au format Unix Epoch. Pour plus d'informations, consultez [Utiliser des fonctions conditionnelles dans les expressions de formule](#) et la fonction [timestamp\(\)](#).

Si la machine reste en état de nettoyage plus longtemps que prévu, le fabricant peut examiner la machine.

Vous pouvez également utiliser la `pretrigger()` fonction dans des transformations multivariées. Par exemple, vous avez deux mesures nommées `x` et `y`, et une transformation `z`, égale à `x + y + pretrigger(y)`. Le tableau suivant indique les valeurs pour `xy`, et `z` entre 9 h 00 et 9 h 15.

Note

- Cet exemple suppose que les valeurs des mesures arrivent par ordre chronologique. Par exemple, la valeur de `x` pour 9 h 00 arrive avant la valeur de `x` pour 9 h 05.

- Si les points de données de 9 h 05 arrivent avant les points de données de 9 h 00, cela z n'est pas calculé à 9 h 05.
- Si la valeur de x pour 9 h 05 arrive avant la valeur de x pour 9 h 00 et que les valeurs de y arrivent chronologiquement, z est égale $22 = 20 + 1 + 1$ à 9 h 05.

	9 H 00	9 H 05	9 H 10	9 H 15
x	10	20		30
y	1	2	3	
$z = x + y + \text{pretrigger}(y)$	yne reçoit aucun point de données avant 9h00. Par conséquent, z n'est pas calculé à 9 h 00.	$23 = 20 + 2 + 1$ pretrigger $r(y)$ est égal à 1.	$25 = 20 + 3 + 2$ xne reçoit pas de nouveau point de données. pretrigger $r(y)$ est égal à 2.	$36 = 30 + 3 + 3$ yne reçoit pas de nouveau point de données. Par conséquent, pretrigger $r(y)$ est égal à 3 à 9 h 15.

Utiliser les fonctions de date et d'heure dans les expressions de formule

Dans les [transformations](#) et [les métriques](#), vous pouvez utiliser les fonctions de date et d'heure de la manière suivante :

- Récupérez l'horodatage actuel d'un point de données en UTC ou dans le fuseau horaire local.
- Construisez des horodatages avec des arguments tels que `yearmonth`, et. `day_of_month`
- Extrayez une période telle qu'une année ou un mois avec l'`unix_time` argument.

Fonction	Description
<code>now()</code>	Renvoie la date et l'heure actuelles, en secondes, au format Unix Epoch.

Fonction	Description
<code>timestamp()</code>	- Lors des transformations, la fonction renvoie l'horodatage, en secondes, du message d'entrée au format Unix Epoch. Dans les transformations uniquement, vous pouvez effectuer l'une des opérations suivantes : - Fournissez une variable comme argument à la fonction. La <code>timestamp (<i>variable-name</i>)</code> fonction renvoie l'horodatage, en secondes, de la dernière valeur de GOOD qualité pour la variable spécifiée au format Unix Epoch. Par exemple, si votre actif possède une propriété de transformation nommée <code>Temperature_F</code> qui utilise la $9/5 * \text{Temperature\_C}$ formule pour convertir chaque point de données de température de degrés Celsius en degrés Fahrenheit, vous pouvez utiliser cette <code>timestamp (Temperature_F)</code> fonction pour obtenir l'horodatage de la dernière valeur de GOOD qualité de la propriété. <code>Temperature_F</code> - Utilisez la <code>pretrigger()</code> fonction comme argument de la fonction. La <code>timestamp(pretrigger(<i>variable-name</i>))</code> fonction renvoie l'horodatage, en secondes, de la valeur de GOOD qualité pour la variable spécifiée avant la mise à jour de la propriété qui a initié le calcul de transformation actuel au format Unix Epoch. Pour de plus amples informations,

Fonction	Description
	veuillez consulter Utiliser des fonctions temporelles dans les transformations. • En métriques, la fonction renvoie l'horodatage récupéré à la fin de la fenêtre en cours, en secondes, au format Unix Epoch.

Fonction	Description
<code>mktime(time_zone, year, month, day_of_month, hour, minute, second)</code>	Renvoie le temps d'entrée en secondes, au format Unix Epoch. Les conditions suivantes s'appliquent à l'utilisation de cette fonction : • L'argument de fuseau horaire doit être une chaîne entre guillemets (' UTC '). S'il n'est pas spécifié, le fuseau horaire par défaut est UTC. L'argument du fuseau horaire peut être le premier ou le dernier argument. • L'année, le mois, le jour du mois, l'heure, la minute et le deuxième argument doivent être en ordre. • Les arguments relatifs à l'année, au mois et à la date sont obligatoires. Les limites suivantes s'appliquent à l'utilisation de cette fonction : • <code>year</code>- Les valeurs valides sont comprises entre 1970 et 2250. • <code>month</code>- Les valeurs valides sont comprises entre 1 et 12. • <code>day-of-month</code> - Les valeurs valides sont comprises entre 1 et 31. • <code>hour</code>- Les valeurs valides sont comprises entre 0 et 23. • <code>minute</code>- Les valeurs valides sont comprises entre 0 et 59.

Fonction	Description
	• second- Les valeurs valides sont comprises entre 0 et 60. Il peut s'agir d'un nombre à virgule flottante. Exemples : • <code>mktime(2020, 2, 29)</code> • <code>mktime('UTC+3', 2021, 12, 31, 22)</code> • <code>mktime(2022, 10, 13, 2, 55, 13.68, 'PST')</code>

Fonction	Description
<code>localtime(unix_time, time_zone)</code>	Renvoie l'année, le jour du mois, le jour de la semaine, le jour de l'année, l'heure, la minute ou la seconde du fuseau horaire spécifié par rapport à l'heure Unix. Les conditions suivantes s'appliquent à l'utilisation de cette fonction : • L'argument de fuseau horaire doit être une chaîne entre guillemets (' UTC '). S'il n'est pas spécifié, le fuseau horaire par défaut est UTC. • L'argument Unix time est le temps en secondes, au format Unix Epoch. La plage valide est comprise entre 1 et 315568898 64403199. Il peut s'agir d'un nombre à virgule flottante. Exemple de réponse : 2007-12-03T10:15:30+01:00[Europe/Paris] <code>localtime(unix_time, time_zone)</code> n'est pas une fonction autonome. Les <code>sec()</code> fonctions <code>year()</code> <code>mon()</code> <code>mday</code>, <code>wday()</code>, <code>yday()</code>, <code>hour()</code> <code>minute()</code>, et prennent <code>localtime(unix_time, time_zone)</code> comme argument. Exemples : • <code>year(localtime('GMT', 1605898608.8113723))</code> • <code>now().localtime().year()</code>

Fonction	Description
	<code>timestamp().localtime('PST').year()</code> <code>localtime(1605289736, 'Europe/London').year()</code>
<code>year(localtime(unix_time, time_zone))</code>	Renvoie l'année à partir de <code>localtime(unix_time, time_zone)</code> .
<code>mon(localtime(unix_time, time_zone))</code>	Renvoie le mois à partir de <code>localtime(unix_time, time_zone)</code> .
<code>mday(localtime(unix_time, time_zone))</code>	Renvoie le jour du mois à partir de <code>localtime(unix_time, time_zone)</code> .
<code>wday(localtime(unix_time, time_zone))</code>	Renvoie le jour de la semaine à partir de <code>localtime(unix_time, time_zone)</code> .
<code>yday(localtime(unix_time, time_zone))</code>	Renvoie le jour de l'année à partir de <code>localtime(unix_time, time_zone)</code> .
<code>hour(localtime(unix_time, time_zone))</code>	Renvoie l'heure de <code>localtime(unix_time, time_zone)</code> .
<code>minute(localtime(unix_time, time_zone))</code>	Renvoie la minute de <code>localtime(unix_time, time_zone)</code> .
<code>sec(localtime(unix_time, time_zone))</code>	Renvoie le second de <code>localtime(unix_time, time_zone)</code> .

Formats de fuseau horaire pris en charge

Vous pouvez spécifier l'argument du fuseau horaire de différentes manières :

- Décalage de fuseau horaire - Spécifiez 'Z' un UTC ou un décalage ('+2' ou '-5').

- Décalage IDs - Combinez une abréviation de fuseau horaire et un décalage. Par exemple : ' GMT +2 ' et ' UTC-01:00 '. L'abréviation du fuseau horaire ne doit contenir que trois lettres.
- Basé sur la région IDs - Par exemple, ' Etc/GMT+12 ' et ' Pacific/Pago_Pago '.

Abréviations de fuseaux horaires prises en charge

Les fonctions de date et d'heure prennent en charge les abréviations de fuseau horaire à trois lettres suivantes :

- EST - 05:00
- HST - 10:00
- MARDI - 07:00
- ACT - Australie/Darwin
- AET - Australie/Sydney
- America/Argentina/BuenosAGT-_Aires
- ART - Africa/Le Caire
- AST - America/Anchorage
- BET - America/Sao_Paulo
- BST - Asia/Dhaka
- CAT - Afrique/Harare
- CET - Europe/Paris
- CNT - America/St_Johns
- CST - America/Chicago
- CTT - Asia/Shanghai
- EAT - Africa/Addis_Abeba
- JET - America/Indiana/Indianapolis
- IST - Asia/Kolkata
- JST - Asia/Tokyo
- MIT - Pacific/Apia
- NET - Asia/Yerevan
- NST - Pacific/Auckland

- PLT - Asia/Karachi
- PRT - America/Puerto_Rico
- PST - America/Los_Angeles
- SST - Pacific/Guadalcanal
- VST - Asia/Ho_Chi_Minh

Soutenu en fonction de la région IDs

Les fonctions de date et d'heure prennent en charge les fonctions suivantes basées sur les régions IDs, organisées en fonction de leur relation avec UTC+ 00:00 :

- ETC/GMT+12 (UTC- 12h00)
- Pacific/Pago_Pago (UTC- 11:00)
- Pacifique/Samoa (UTC- 11:00)
- Pacifique/Niue (UTC- 11:00)
- Etats-Unis/Samoa (UTC- 11:00)
- ETC/GMT+11 (UTC- 11:00)
- Pacifique/Midway (UTC- 11:00)
- Pacifique/Honolulu (UTC- 10:00)
- Pacifique/Rarotonga (UTC- 10:00)
- Pacifique/Tahiti (UTC- 10:00)
- Pacifique/Johnston (UTC- 10:00)
- Etats-Unis/Hawaï (UTC- 10:00)
- Système V/ HST1 0 (UTC- 10:00)
- ETC/GMT+10 (UTC- 10:00)
- Pacifique/Marquises (UTC- 09:30)
- ETC/GMT+9 (UTC- 09:00)
- Pacifique/Gambier (UTC- 09:00)
- Amérique/Atka (UTC- 09:00)
- Système V/ YST9 (UTC- 09:00)

- Amérique/Adak (UTC- 09:00)
- États-Unis/Aléoutiennes (UTC- 09:00)
- ETC/GMT+8 (UTC- 08:00)
- États-Unis/Alaska (UTC- 08:00)
- Amérique/Juneau (UTC- 08:00)
- Amérique/Metlakatla (UTC- 08:00)
- Amérique/Yakutat (UTC- 08:00)
- Pacifique/Pitcairn (UTC- 08:00)
- Amérique/Sitka (UTC- 08:00)
- Amérique/Anchorage (UTC- 08:00)
- Système V/ PST8 (UTC- 08:00)
- Amérique/Nome (UTC- 08:00)
- Système YST9 V/YDT (UTC- 08:00)
- Canada/Yukon (UTC- 07:00)
- US/Pacific-New (UTC- 07:00)
- ETC/GMT+7 (UTC- 07:00)
- États-Unis/Arizona (UTC- 07:00)
- Amérique/Dawson_Creek (UTC- 07:00)
- Canada/Pacifique (UTC- 07:00)
- PST8PDT (UTC- 07:00)
- Système V/ MST7 (UTC- 07:00)
- Amérique/Dawson (UTC- 07:00)
- Mexique/ BajaNorte (UTC- 07:00)
- Amérique/Tijuana (UTC- 07:00)
- Amérique/Creston (UTC- 07:00)
- America/Hermosillo (UTC- 07:00)
- Amérique/Santa_Isabel (UTC- 07:00)
- Amérique/Vancouver (UTC- 07:00)

- America/Ensenada (UTC- 07:00)
- Amérique/Phoenix (UTC- 07:00)
- Amérique/Whitehorse (UTC- 07:00)
- America/Fort_Nelson (UTC- 07:00)
- Système PST8 V/PDT (UTC- 07:00)
- Amérique/Los_Angeles (UTC- 07:00)
- États-Unis/Pacifique (UTC- 07:00)
- America/El Salvador (UTC- 06:00)
- Amérique/Guatemala (UTC- 06:00)
- Amérique/Belize (UTC- 06:00)
- Amérique/Managua (UTC- 06:00)
- America/Tegucigalpa (UTC- 06:00)
- ETC/GMT+6 (UTC- 06:00)
- Pacifique/Pâques (UTC- 06:00)
- Mexique/ BajaSur (UTC- 06:00)
- America/Regina (UTC- 06:00)
- Amérique/Denver (UTC- 06:00)
- Pacifique/Galapagos (UTC- 06:00)
- Amérique/Yellowknife (UTC- 06:00)
- America/Swift_Current (UTC- 06:00)
- Amérique/Inuvik (UTC- 06:00)
- Amérique/Mazatlan (UTC- 06:00)
- Amérique/Boise (UTC- 06:00)
- Amérique/Costa_Rica (UTC- 06:00)
- MST7MDT (UTC- 06:00)
- Système V/ CST6 (UTC- 06:00)
- Amérique/Chihuahua (UTC- 06:00)
- Amérique/Ojinaga (UTC- 06:00)
- Chili/ EasterIsland (UTC- 06:00)

- Etats-Unis/Montagne (UTC- 06:00)
- Amérique/Edmonton (UTC- 06:00)
- Canada/Montagne (UTC- 06:00)
- Amérique/Cambridge_Bay (UTC- 06:00)
- Navajo (UTC- 06:00)
- Système MST7 V/MDT (UTC- 06:00)
- Canada/Saskatchewan (UTC- 06:00)
- Amérique/Shiprock (UTC- 06:00)
- Amérique/Panama (UTC- 05:00)
- Amérique/Chicago (UTC- 05:00)
- America/Eirunepe (UTC- 05:00)
- ETC/GMT+5 (UTC- 05:00)
- Mexique/Général (UTC- 05:00)
- America/Porto_Acre (UTC- 05:00)
- Amérique/Guayaquil (UTC- 05:00)
- America/Rankin_Inlet (UTC- 05:00)
- US/Central (UTC- 05:00)
- Amérique/Rainy_River (UTC- 05:00)
- America/Indiana/Knox(UTC- 05:00)
- America/North_Dakota/Beulah(UTC- 05:00)
- Amérique/Monterrey (UTC- 05:00)
- Amérique/Jamaïque (UTC- 05:00)
- Amérique/Atikokan (UTC- 05:00)
- America/Coral_Harbour (UTC- 05:00)
- America/North_Dakota/Center(UTC- 05:00)
- Amérique/Cayman (UTC- 05:00)
- America/Indiana/Tell_Ville (UTC- 05:00)
- America/Mexico_City (UTC- 05:00)
- Amérique/Matamoros (UTC- 05:00)

- CST6CDT (UTC- 05:00)
- America/Knox_IN (UTC- 05:00)
- Amérique/Bogota (UTC- 05:00)
- Amérique/Menominee (UTC- 05:00)
- America/Resolute (UTC- 05:00)
- Système V/ EST5 (UTC- 05:00)
- Canada/Centre (UTC- 05:00)
- Brésil/Acre (UTC- 05:00)
- Amérique/Cancun (UTC- 05:00)
- Amérique/Lima (UTC- 05:00)
- Amérique/Bahia_Banderas (UTC- 05:00)
- US/Indiana-Starke (UTC- 05:00)
- America/Rio_Branco (UTC- 05:00)
- Système CST6 V/CDT (UTC- 05:00)
- Jamaïque (UTC- 05:00)
- Amérique/Mérida (UTC- 05:00)
- America/North_Dakota/New_Salem (UTC- 05:00)
- Amérique/Winnipeg (UTC- 05:00)
- Amérique/Cuiaba (UTC- 04:00)
- Amérique/Marigot (UTC- 04:00)
- America/Indiana/Petersburg(UTC- 04:00)
- Chili/Continental (UTC- 04:00)
- America/Grand_Turk (UTC- 04:00)
- Cuba (UTC- 04:00)
- ETC/GMT+4 (UTC- 04:00)
- Amérique/Manaus (UTC- 04:00)
- Amérique/Fort_Wayne (UTC- 04:00)
- America/St_Thomas (UTC- 04:00)
- Amérique/Anguilla (UTC- 04:00)

- Amérique/La Havane (UTC- 04:00)
- États-Unis/Michigan (UTC- 04:00)
- Amérique/Barbade (UTC- 04:00)
- Amérique/Louisville (UTC- 04:00)
- Amérique/Curaçao (UTC- 04:00)
- Amérique/Guyana (UTC- 04:00)
- Amérique/Martinique (UTC- 04:00)
- Amérique/Porto_Rico (UTC- 04:00)
- Amérique/Port_of_Spain (UTC- 04:00)
- Système V/ AST4 (UTC- 04:00)
- America/Indiana/Vevay(UTC- 04:00)
- America/Indiana/Vincennes(UTC- 04:00)
- America/Kralendijk (UTC- 04:00)
- Amérique/Antigua (UTC- 04:00)
- Amérique/Indianapolis (UTC- 04:00)
- Amérique/Iqaluit (UTC- 04:00)
- America/St_Vincent (UTC- 04:00)
- America/Kentucky/Louisville(UTC- 04:00)
- Amérique/Dominique (UTC- 04:00)
- America/Asuncion (UTC- 04:00)
- EST5HAE (UTC- 04:00)
- Amérique/Nassau (UTC- 04:00)
- America/Kentucky/Monticello(UTC- 04:00)
- Brésil/Ouest (UTC- 04:00)
- Amérique/Aruba (UTC- 04:00)
- America/Indiana/Indianapolis(UTC- 04:00)
- Amérique/Santiago (UTC- 04:00)
- America/La_Paz (UTC- 04:00)
- America/Thunder_Bay (UTC- 04:00)

- America/Indiana/Marengo(UTC- 04:00)
- Amérique/Blanc-Sablon (UTC- 04:00)
- America/Santo_Domingo (UTC- 04:00)
- Etats-Unis/Est (UTC- 04:00)
- Canada/Est (UTC- 04:00)
- Amérique/ Port-au-Prince (UTC- 04:00)
- America/Saint-Barthélemy (UTC- 04:00)
- Amérique/Nipigon (UTC- 04:00)
- Etats-Unis/Est de l'Indiana (UTC- 04:00)
- Amérique/Sainte-Lucie (UTC- 04:00)
- Amérique/Montserrat (UTC- 04:00)
- America/Lower_Princes (UTC- 04:00)
- Amérique/Détroit (UTC- 04:00)
- Amérique/Tortola (UTC- 04:00)
- America/Porto_Velho (UTC- 04:00)
- America/Campo_Grande (UTC- 04:00)
- America/Virgin (UTC- 04:00)
- America/Pangnirtung (UTC- 04:00)
- Amérique/Montréal (UTC- 04:00)
- America/Indiana/Winamac(UTC- 04:00)
- America/Boa_Vista (UTC- 04:00)
- Amérique/Grenade (UTC- 04:00)
- Amérique/New_York (UTC- 04:00)
- America/St_Kitts (UTC- 04:00)
- Amérique/Caracas (UTC- 04:00)
- Amérique/Guadeloupe (UTC- 04:00)
- Amérique/Toronto (UTC- 04:00)
- Système EST5 V/EDT (UTC- 04:00)
- America/Argentina/Catamarca(UTC- 03:00)

- Canada/Atlantique (UTC- 03:00)
- America/Argentina/Cordoba(UTC- 03:00)
- Amérique/Araguaina (UTC- 03:00)
- America/Argentina/Salta(UTC- 03:00)
- ETC/GMT+3 (UTC- 03:00)
- Amérique/Montevideo (UTC- 03:00)
- Brésil/Est (UTC- 03:00)
- America/Argentina/Mendoza(UTC- 03:00)
- America/Argentina/Rio_Gallegos (UTC- 03:00)
- Amérique/Catamarca (UTC- 03:00)
- Amérique/Cordoba (UTC- 03:00)
- America/Sao_Paulo (UTC- 03:00)
- America/Argentina/Jujuy(UTC- 03:00)
- Amérique/Cayenne (UTC- 03:00)
- Amérique/Recife (UTC- 03:00)
- America/Buenos_Aires (UTC- 03:00)
- Amérique/Paramaribo (UTC- 03:00)
- Amérique/Moncton (UTC- 03:00)
- America/Mendoza (UTC- 03:00)
- Amérique/Santarém (UTC- 03:00)
- Atlantique/Bermudes (UTC- 03:00)
- Amérique/Maceio (UTC- 03:00)
- Atlantic/Stanley (UTC- 03:00)
- Amérique/Halifax (UTC- 03:00)
- Antarctique/Rothera (UTC- 03:00)
- America/Argentina/San_Luis (UTC- 03:00)
- America/Argentina/Ushuaia(UTC- 03:00)
- Antarctique/Palmer (UTC- 03:00)
- America/Punta_Arenas (UTC- 03:00)

- Amérique/Glace_Bay (UTC- 03:00)
- America/Fortaleza (UTC- 03:00)
- Amérique/Thule (UTC- 03:00)
- America/Argentina/La_Rioja (UTC- 03:00)
- Amérique/Belém (UTC- 03:00)
- America/Jujuy (UTC- 03:00)
- Amérique/Bahia (UTC- 03:00)
- Amérique/Goose_Bay (UTC- 03:00)
- America/Argentina/San_Juan (UTC- 03:00)
- America/Argentina/ComodRivadavia(UTC- 03:00)
- America/Argentina/Tucuman(UTC- 03:00)
- America/Rosario (UTC- 03:00)
- Système AST4 V/ADT (UTC- 03:00)
- America/Argentina/Buenos_Aires (UTC- 03:00)
- America/St_Johns (UTC- 02:30)
- Canada/Terre-Neuve (UTC- 02:30)
- Amérique/Miquelon (UTC- 02:00)
- ETC/GMT+2 (UTC- 02:00)
- Amérique/Godthab (UTC- 02:00)
- America/Noronha (UTC- 02:00)
- Brésil/ DeNoronha (UTC- 02:00)
- Atlantique/Géorgie du Sud (UTC- 02:00)
- Etc/GMT+1 (UTC- 01:00)
- Atlantique/Cap-Vert (UTC- 01:00)
- Pacifique/Kiritimati (UTC+ 14:00)
- Etc/GMT-14 (UTC+ 14:00)
- Pacifique/Fakaofu (UTC+ 13:00)
- Pacifique/Enderbury (UTC+ 13:00)
- Pacifique/Apia (UTC+ 13:00)

- Pacifique/Tongatapu (UTC+ 13:00)
- Etc/GMT-13 (UTC+ 13:00)
- NZ-CHAT (UTC+ 12:45)
- Pacifique/Chatham (UTC+ 12:45)
- Pacifique/Kwajalein (UTC+ 12:00)
- Antarctique/ McMurdo (UTC+ 12:00)
- Pacifique/Wallis (UTC+ 12:00)
- Pacifique/Fidji (UTC+ 12:00)
- Pacifique/Funafuti (UTC+ 12:00)
- Pacifique/Nauru (UTC+ 12:00)
- Kwajalein (UTC+ 12:00)
- NOUVELLE-ZÉLANDE (UTC+ 12:00)
- Pacific/Wake (UTC+ 12:00)
- Antarctique/Pôle Sud (UTC+ 12:00)
- Pacifique/Tarawa (UTC+ 12:00)
- Pacifique/Auckland (UTC+ 12:00)
- Asie/Kamchatka (UTC+ 12:00)
- etc/GMT-12 (UTC+ 12:00)
- Asie/Anadyr (UTC+ 12:00)
- Pacifique/Majuro (UTC+ 12:00)
- Pacifique/Ponape (UTC+ 11:00)
- Pacifique/Bougainville (UTC+ 11:00)
- Antarctique/Macquarie (UTC+ 11:00)
- Pacifique/Pohnpei (UTC+ 11:00)
- Pacifique/Efate (UTC+ 11:00)
- Pacifique/Norfolk (UTC+ 11:00)
- Asie/Magadan (UTC+ 11:00)
- Pacifique/Kosrae (UTC+ 11:00)
- Asie/Sakhaline (UTC+ 11:00)

- Pacifique/Nouméa (UTC+ 11:00)
- Etc/GMT-11 (UTC+ 11:00)
- Asie/Srednekolymsk (UTC+ 11:00)
- Pacifique/Guadalcanal (UTC+ 11:00)
- Australie/Lord_Howe (UTC+ 10:30)
- Australie/LHI (UTC+ 10:30)
- Australie/Hobart (UTC+ 10:00)
- Pacifique/Yap (UTC+ 10:00)
- Australie/Tasmanie (UTC+ 10:00)
- Pacifique/Port_Moresby (UTC+ 10:00)
- Australie/ACT (UTC+ 10:00)
- Australie/Victoria (UTC+ 10:00)
- Pacifique/Chuuk (UTC+ 10:00)
- Australie/Queensland (UTC+ 10:00)
- Australie/Canberra (UTC+ 10:00)
- Australie/Currie (UTC+ 10:00)
- Pacifique/Guam (UTC+ 10:00)
- Pacifique/Truk (UTC+ 10:00)
- Australie/NSW (UTC+ 10:00)
- Asie/Vladivostok (UTC+ 10:00)
- Pacifique/Saipan (UTC+ 10:00)
- Antarctique/Dumont DUrville (UTC+ 10:00)
- Australie/Sydney (UTC+ 10:00)
- Australie/Brisbane (UTC+ 10:00)
- Etc/GMT-10 (UTC+ 10:00)
- Asie/Ust-Nera (UTC+ 10:00)
- Australie/Melbourne (UTC+ 10:00)
- Australie/Lindeman (UTC+ 10:00)
- Australie/Nord (UTC+ 09:30)

- Australie/Yancowinna (UTC+ 09:30)
- Australie/Adélaïde (UTC+ 09:30)
- Australie/Broken_Hill (UTC+ 09:30)
- Australie/Sud (UTC+ 09:30)
- Australie/Darwin (UTC+ 09:30)
- Etc/GMT-9 (UTC+ 09:00)
- Pacifique/Palaos (UTC+ 09:00)
- Asie/Chita (UTC+ 09:00)
- Asie/Dili (UTC+ 09:00)
- Asie/Jayapura (UTC+ 09:00)
- Asie/Yakutsk (UTC+ 09:00)
- Asie/Pyongyang (UTC+ 09:00)
- MERCREDI (UTC+ 09:00)
- Asie/Séoul (UTC+ 09:00)
- Asie/Khandyga (UTC+ 09:00)
- Japon (UTC+ 09:00)
- Asie/Tokyo (UTC+ 09:00)
- Australie/Eucla (UTC+ 08:45)
- Asie/Kuching (UTC+ 08:00)
- Asie/Chungking (UTC+ 08:00)
- Etc/GMT-8 (UTC+ 08:00)
- Australie/Perth (UTC+ 08:00)
- Asie/Macao (UTC+ 08:00)
- Asie/Macao (UTC+ 08:00)
- Asie/Choibalsan (UTC+ 08:00)
- Asie/Shanghai (UTC+ 08:00)
- Antarctique/Casey (UTC+ 08:00)
- Asie/Ulan_Bator (UTC+ 08:00)
- Asie/Chongqing (UTC+ 08:00)

- Asie/Oulan-Bator (UTC+ 08:00)
- Asie/Taipei (UTC+ 08:00)
- Asie/Manille (UTC+ 08:00)
- PRC (UTC+ 08:00)
- Asie/Ujung_Pandang (UTC+ 08:00)
- Asie/Harbin (UTC+ 08:00)
- Singapour (UTC+ 08:00)
- Asie/Brunei (UTC+ 08:00)
- Australie/Ouest (UTC+ 08:00)
- Asie/Hong_Kong (UTC+ 08:00)
- Asie/Makassar (UTC+ 08:00)
- Hong Kong (UTC+ 08:00)
- Asie/Kuala_Lumpur (UTC+ 08:00)
- Asie/Irkutsk (UTC+ 08:00)
- Asie/Singapour (UTC+ 08:00)
- Asie/Pontianak (UTC+ 07:00)
- Etc/GMT-7 (UTC+ 07:00)
- Asie/Phnom_Penh (UTC+ 07:00)
- Asie/Novossibirsk (UTC+ 07:00)
- Antarctique/Davis (UTC+ 07:00)
- Asie/Tomsk (UTC+ 07:00)
- Asie/Jakarta (UTC+ 07:00)
- Asie/Barnaoul (UTC+ 07:00)
- Indien/Noël (UTC+ 07:00)
- Asie/Ho_Chi_Minh (UTC+ 07:00)
- Asie/Hovd (UTC+ 07:00)
- Asie/Bangkok (UTC+ 07:00)
- Asie/Vientiane (UTC+ 07:00)
- Asie/Novokuznetsk (UTC+ 07:00)

- Asie/Krasnoïarsk (UTC+ 07:00)
- Asie/Saigon (UTC+ 07:00)
- Asie/Yangon (UTC+ 06:30)
- Asie/Rangoon (UTC+ 06:30)
- Indien/Cocos (UTC+ 06:30)
- Asie/Kachgar (UTC+ 06:00)
- Etc/GMT-6 (UTC+ 06:00)
- Asie/Almaty (UTC+ 06:00)
- Asie/Dacca (UTC+ 06:00)
- Asie/Omsk (UTC+ 06:00)
- Asie/Dhaka (UTC+ 06:00)
- Indien/Chagos (UTC+ 06:00)
- Asie/Qyzylorda (UTC+ 06:00)
- Asie/Bichkek (UTC+ 06:00)
- Antarctique/Vostok (UTC+ 06:00)
- Asie/Urumqi (UTC+ 06:00)
- Asie/Thimbu (UTC+ 06:00)
- Asie/Thimphou (UTC+ 06:00)
- Asie/Katmandou (UTC+ 05:45)
- Asie/Katmandou (UTC+ 05:45)
- Asie/Kolkata (UTC+ 05:30)
- Asie/Colombo (UTC+ 05:30)
- Asie/Calcutta (UTC+ 05:30)
- Asie/Aqtau (UTC+ 05:00)
- Etc/GMT-5 (UTC+ 05:00)
- Asie/Samarkand (UTC+ 05:00)
- Asie/Karachi (UTC+ 05:00)
- Asie/Ekaterinbourg (UTC+ 05:00)
- Asie/Douchanbé (UTC+ 05:00)

- Indien/Maldives (UTC+ 05:00)
- Asie/Oral (UTC+ 05:00)
- Asie/Tachkent (UTC+ 05:00)
- Antarctique/Mawson (UTC+ 05:00)
- Asie/Aqtobe (UTC+ 05:00)
- Asie/Ashkhabad (UTC+ 05:00)
- Asie/Ashgabat (UTC+ 05:00)
- Asie/Atyrau (UTC+ 05:00)
- Indien/Kerguelen (UTC+ 05:00)
- Iran (UTC+ 04:30)
- Asie/Téhéran (UTC+ 04:30)
- Asie/Kaboul (UTC+ 04:30)
- Asie/Yerevan (UTC+ 04:00)
- Etc/GMT-4 (UTC+ 04:00)
- Etc/GMT-4 (UTC+ 04:00)
- Asie/Dubaï (UTC+ 04:00)
- Indien/Réunion (UTC+ 04:00)
- Europe/Saratov (UTC+ 04:00)
- Europe/Samara (UTC+ 04:00)
- Indien/Mahé (UTC+ 04:00)
- Asie/Bakou (UTC+ 04:00)
- Asie/Muscat (UTC+ 04:00)
- Europe/Volgograd (UTC+ 04:00)
- Europe/Astrakhan (UTC+ 04:00)
- Asie/Tbilissi (UTC+ 04:00)
- Europe/Oulianovsk (UTC+ 04:00)
- Asie/Aden (UTC+ 03:00)
- Afrique/Nairobi (UTC+ 03:00)
- Europe/Istanbul (UTC+ 03:00)

- Etc/GMT-3 (UTC+ 03:00)
- Europe/Zaporijia (UTC+ 03:00)
- Israël (UTC+ 03:00)
- Indien/Comores (UTC+ 03:00)
- Antarctique/Syowa (UTC+ 03:00)
- Afrique/Mogadiscio (UTC+ 03:00)
- Europe/Bucarest (UTC+ 03:00)
- Afrique/Asmera (UTC+ 03:00)
- Europe/Mariehamn (UTC+ 03:00)
- Asie/Istanbul (UTC+ 03:00)
- Europe/Tiraspol (UTC+ 03:00)
- Europe/Moscou (UTC+ 03:00)
- Europe/Chişinău (UTC+ 03:00)
- Europe/Helsinki (UTC+ 03:00)
- Asie/Beyrouth (UTC+ 03:00)
- Asie/Tel_Aviv (UTC+ 03:00)
- Afrique/Djibouti (UTC+ 03:00)
- Europe/Simferopol (UTC+ 03:00)
- Europe/Sofia (UTC+ 03:00)
- Asie/Gaza (UTC+ 03:00)
- Afrique/Asmara (UTC+ 03:00)
- Europe/Riga (UTC+ 03:00)
- Asie/Bagdad (UTC+ 03:00)
- Asie/Damascus (UTC+ 03:00)
- AFRIQUE/Dar_es_Salaam (UTC+ 03:00)
- Afrique/Addis_Abeba (UTC+ 03:00)
- Europe/Ujgorod (UTC+ 03:00)
- Asie/Jerusalem (UTC+ 03:00)
- Asie/Riyadh (UTC+ 03:00)

- Asie/Koweït (UTC+ 03:00)
- Europe/Kirov (UTC+ 03:00)
- Afrique/Kampala (UTC+ 03:00)
- Europe/Minsk (UTC+ 03:00)
- Asie/Qatar (UTC+ 03:00)
- Europe/Kiev (UTC+ 03:00)
- Asie/Bahreïn (UTC+ 03:00)
- Europe/Vilnius (UTC+ 03:00)
- Indien/Antananarivo (UTC+ 03:00)
- Indien/Mayotte (UTC+ 03:00)
- Europe/Tallinn (UTC+ 03:00)
- Turquie (UTC+ 03:00)
- Afrique/Juba (UTC+ 03:00)
- Asie/Nicosie (UTC+ 03:00)
- Asie/Famagouste (UTC+ 03:00)
- SAMEDI (UTC+ 03:00)
- RENCONTRE (UTC+ 03:00)
- Asie/Hébron (UTC+ 03:00)
- Asie/Amman (UTC+ 03:00)
- Europe/Nicosie (UTC+ 03:00)
- Europe/Athènes (UTC+ 03:00)
- Afrique/Le Caire (UTC+ 02:00)
- Afrique/Mbabane (UTC+ 02:00)
- Europe/Bruxelles (UTC+ 02:00)
- Europe/Varsovie (UTC+ 02:00)
- HEURE DE PARIS (UTC+ 02:00)
- Europe/Luxembourg (UTC+ 02:00)
- Etc/GMT-2 (UTC+ 02:00)
- Libye (UTC+ 02:00)

- Afrique/Kigali (UTC+ 02:00)
- Afrique/Tripoli (UTC+ 02:00)
- Europe/Kaliningrad (UTC+ 02:00)
- Afrique/Windhoek (UTC+ 02:00)
- Europe/Malte (UTC+ 02:00)
- Europe/Büdingen (UTC+ 02:00)
-
- Europe/Skopje (UTC+ 02:00)
- Europe/Sarajevo (UTC+ 02:00)
- Europe/Rome (UTC+ 02:00)
- Europe/Zürich (UTC+ 02:00)
- Europe/Gibraltar (UTC+ 02:00)
- Afrique/Lubumbashi (UTC+ 02:00)
- Europe/Vaduz (UTC+ 02:00)
- Europe/Ljubljana (UTC+ 02:00)
- Europe/Berlin (UTC+ 02:00)
- Europe/Stockholm (UTC+ 02:00)
- Europe/Budapest (UTC+ 02:00)
- Europe/Zagreb (UTC+ 02:00)
- Europe/Paris (UTC+ 02:00)
- Afrique/Ceuta (UTC+ 02:00)
- Europe/Praha (UTC+ 02:00)
- Antarctique/Troll (UTC+ 02:00)
- Afrique/Gaborone (UTC+ 02:00)
- Europe/Copenhague (UTC+ 02:00)
- Europe/Vienne (UTC+ 02:00)
- Europe/Tirana (UTC+ 02:00)
- MET (UTC+ 02:00)
- Europe/Amsterdam (UTC+ 02:00)

- Afrique/Maputo (UTC+ 02:00)
- Europe/Saint-Marin (UTC+ 02:00)
- Pologne (UTC+ 02:00)
- Europe/Andorre (UTC+ 02:00)
- Europe/Oslo (UTC+ 02:00)
- Europe/Podgorica (UTC+ 02:00)
- Afrique/Bujumbura (UTC+ 02:00)
- Atlantic/Jan_Mayen (UTC+ 02:00)
- Afrique/Maseru (UTC+ 02:00)
- Europe/Madrid (UTC+ 02:00)
- Afrique/Blantyre (UTC+ 02:00)
- Afrique/Lusaka (UTC+ 02:00)
- Afrique/Harare (UTC+ 02:00)
- Afrique/Khartoum (UTC+ 02:00)
- Afrique/Johannesburg (UTC+ 02:00)
- Europe/Belgrade (UTC+ 02:00)
- Europe/Bratislava (UTC+ 02:00)
- Arctic/Longyearbyen (UTC+ 02:00)
- Égypte (UTC+ 02:00)
- Europe/Vatican (UTC+ 02:00)
- Europe/Monaco (UTC+ 02:00)
- Europe/Londres (UTC+ 01:00)
- Etc/GMT-1 (UTC+ 01:00)
- Europe/Jersey (UTC+ 01:00)
- Europe/Guernesey (UTC+ 01:00)
- Europe/Isle_of_Man (UTC+ 01:00)
- Afrique/Tunis (UTC+ 01:00)
- Afrique/Malabo (UTC+ 01:00)
- GB-Eire (UTC+ 01:00)

- Afrique/Lagos (UTC+ 01:00)
- Afrique/Alger (UTC+ 01:00)
- GB (UTC+ 01:00)
- Portugal (UTC+ 01:00)
- Afrique/Sao_Tome (UTC+ 01:00)
- Afrique/Ndjamena (UTC+ 01:00)
- Atlantique/Féroé (UTC+ 01:00)
- Irlande (UTC+ 01:00)
- Atlantique/Féroé (UTC+ 01:00)
- Europe/Dublin (UTC+ 01:00)
- Afrique/Libreville (UTC+ 01:00)
- Afrique/El_Aaiun (UTC+ 01:00)
- Afrique/El_Aaiun (UTC+ 01:00)
- Afrique/Douala (UTC+ 01:00)
- Afrique/Brazzaville (UTC+ 01:00)
- Afrique/Porto-Novo (UTC+ 01:00)
- Atlantique/Madère (UTC+ 01:00)
- Europe/Lisbonne (UTC+ 01:00)
- Atlantique/Canaries (UTC+ 01:00)
- Afrique/Casablanca (UTC+ 01:00)
- Europe/Belfast (UTC+ 01:00)
- Afrique/Luanda (UTC+ 01:00)
- Afrique/Kinshasa (UTC+ 01:00)
- Afrique/Bangui (UTC+ 01:00)
- HUMIDE (UTC+ 01:00)
- Afrique/Niamey (UTC+ 01:00)
- GMT (UTC+ 00:00)
- Etc/GMT-0 (UTC+ 00:00)
- Atlantic/Sainte-Hélène (UTC+ 00:00)

- Etc/GMT+0 (UTC+ 00:00)
- Afrique/Banjul (UTC+ 00:00)
- Etc/GMT (UTC+ 00:00)
- Afrique/Freetown (UTC+ 00:00)
- Afrique/Bamako (UTC+ 00:00)
- Afrique/Conakry (UTC+ 00:00)
- Universel (UTC+ 00:00)
- Afrique/Nouakchott (UTC+ 00:00)
- UTC (UTC+ 00:00)
- Etc/Universal (UTC+ 00:00)
- Atlantique/Açores (UTC+ 00:00)
- Afrique/Abidjan (UTC+ 00:00)
- Afrique/Accra (UTC+ 00:00)
- Etc/UCT (UTC+ 00:00)
- GMT0 (UTC+ 00:00)
- Zoulou (UTC+ 00:00) Zoulou (UTC+ 00:00)
- Afrique/Ouagadougou (UTC+ 00:00)
- Atlantique/Reykjavik (UTC+ 00:00)
- Etc/Zoulou (UTC+ 00:00)
- Islande (UTC+ 00:00)
- Afrique/Lomé (UTC+ 00:00)
- Greenwich (UTC+ 00:00)
- Etc/GMT0 (UTC+ 00:00)
- America/Danmarkshavn (UTC+ 00:00)
- Afrique/Dakar (UTC+ 00:00)
- Afrique/Bissau (UTC+ 00:00)
- Etc/Greenwich (UTC+ 00:00)
- Afrique/Tombouctou (UTC+ 00:00)
- UTC (UTC+ 00:00)

- Afrique/Monrovia (UTC+ 00:00)
- Etc/UTC (UTC+ 00:00)

Tutoriels d'expression de formules

Vous pouvez suivre ces didacticiels pour utiliser des expressions de formule dans AWS IoT SiteWise.

Rubriques

- [Utiliser des chaînes dans les formules](#)
- [Filtrer les points de données](#)
- [Compter les points de données qui correspondent à une condition](#)
- [Données tardives dans les formules](#)
- [Qualité des données dans les formules](#)
- [Valeurs non définies, infinies et en dépassement](#)

Utiliser des chaînes dans les formules

Vous pouvez opérer sur des chaînes dans vos expressions de formule. Vous pouvez également saisir des chaînes à partir de variables qui font référence à des propriétés d'attribut et de mesure.

Important

Les expressions de formule ne peuvent générer que des valeurs doubles ou des valeurs de chaîne. Les expressions imbriquées peuvent générer d'autres types de données, tels que des chaînes, mais la formule dans son ensemble doit être évaluée à un nombre ou à une chaîne. Vous pouvez utiliser la [fonction jp](#) pour convertir une chaîne en nombre. La valeur booléenne doit être 1 (vrai) ou 0 (faux). Pour de plus amples informations, veuillez consulter [Valeurs non définies, infinies et en dépassement](#).

AWS IoT SiteWise fournit les fonctionnalités d'expression de formule suivantes que vous pouvez utiliser pour agir sur des chaînes :

- [Littéraux de chaîne](#)
- L'[opérateur d'index](#) (s[index])

- L'[opérateur de tranche](#) (s[start:end:step])
- [Fonctions de comparaison](#), que vous pouvez utiliser pour comparer des chaînes par ordre [lexicographique](#)
- [Fonctions de chaîne](#), qui incluent la `jp` fonction capable d'analyser des objets JSON sérialisés et de convertir des chaînes en nombres

Filtrer les points de données

Vous pouvez utiliser la [fonction if](#) pour filtrer les points de données qui ne répondent pas à une condition. La `if` fonction évalue une condition et renvoie des valeurs `true` et des `false` résultats différents. Vous pouvez utiliser la [constante none](#) comme sortie pour un cas de `if` fonction afin de supprimer le point de données correspondant à ce cas.

Pour filtrer les points de données qui correspondent à une condition

- Créez une transformation qui utilise la `if` fonction pour définir une condition qui vérifie si une condition est remplie et renvoie `none` la `result_if_false` valeur `result_if_true` ou.

Exemple Exemple : filtrer les points de données où l'eau ne bout pas

Imaginons un scénario dans lequel vous avez une mesure qui fournit la température (en degrés Celsius) de l'eau dans une machine. `temp_c` Vous pouvez définir la transformation suivante pour filtrer les points de données où l'eau n'est pas en ébullition :

- Transformation : `boiling_temps = if(gte(temp_c, 100), temp_c, none)` — Renvoie la température si elle est supérieure ou égale à 100 degrés Celsius, sinon elle ne renvoie aucun point de données.

Compter les points de données qui correspondent à une condition

Vous pouvez utiliser [les fonctions de comparaison](#) et [sum \(\)](#) pour compter le nombre de points de données pour lesquels une condition est vraie.

Pour compter les points de données qui correspondent à une condition

1. Créez une transformation qui utilise une fonction de comparaison pour définir une condition de filtre sur une autre propriété.
2. Créez une métrique qui additionne les points de données lorsque cette condition est remplie.

Exemple Exemple : Compter le nombre de points de données où l'eau bout

Imaginons un scénario dans lequel vous avez une mesure qui fournit la température (en degrés Celsius) de l'eau dans une machine. `temp_c` Vous pouvez définir les propriétés de transformation et de métrique suivantes pour compter le nombre de points de données où l'eau bout :

- Transformation : `is_boiling = gte(temp_c, 100)` — Renvoie 1 si la température est supérieure ou égale à 100 degrés Celsius, sinon elle renvoie la valeur 0.
- Métrique : `boiling_count = sum(is_boiling)` — Renvoie le nombre de points de données où l'eau est en ébullition.

Données tardives dans les formules

AWS IoT SiteWise prend en charge l'ingestion tardive de données datant de moins de 7 jours. Lorsqu'il AWS IoT SiteWise reçoit des données tardives, il recalcule les valeurs existantes pour toute métrique qui saisit les données tardives dans une fenêtre précédente. Ces nouveaux calculs entraînent des frais de traitement des données.

Note

Lorsqu'il AWS IoT SiteWise calcule des propriétés qui entrent des données tardives, il utilise l'expression de formule actuelle de chaque propriété.

Après avoir AWS IoT SiteWise recalculé une fenêtre passée pour une métrique, elle remplace la valeur précédente pour cette fenêtre. Si vous avez activé les notifications pour cette métrique, émet AWS IoT SiteWise également une notification de valeur de propriété. Cela signifie que vous pouvez recevoir une nouvelle notification de mise à jour de valeur de propriété pour la même propriété et le même horodatage que ceux pour lesquels vous avez déjà reçu une notification. Si vos applications ou lacs de données utilisent des notifications de valeur de propriété, vous devez mettre à jour la valeur précédente avec la nouvelle valeur afin que leurs données soient exactes.

Qualité des données dans les formules

Dans AWS IoT SiteWise, chaque point de données possède un code de qualité, qui peut être l'un des suivants :

- **GOOD**— Les données ne sont affectées par aucun problème.

- BAD— Les données sont affectées par un problème tel qu'une défaillance du capteur.
- UNCERTAIN— Les données sont affectées par un problème tel que l'imprécision du capteur.

AWS IoT SiteWise ne consomme que des données GOOD de qualité lorsqu'il calcule les transformations et les métriques. AWS IoT SiteWise ne produit que des données GOOD de qualité pour des calculs réussis. Si un calcul échoue, aucun point de données AWS IoT SiteWise n'est généré pour ce calcul. Cela peut se produire si un calcul aboutit à une valeur non définie, infinie ou en dépassement.

Pour de plus amples informations sur l'interrogation des données et les filtres par qualité de données, veuillez consulter [Interrogez les données de AWS IoT SiteWise](#).

Valeurs non définies, infinies et en dépassement

Certaines expressions de formule (telles que $x / 0$, $\sqrt{-1}$, ou $\log(0)$) calculent des valeurs non définies dans un système de nombres réels, infinies ou situées en dehors de la plage prise en charge par AWS IoT SiteWise. Lorsque l'expression d'une propriété d'actif calcule une valeur indéfinie, infinie ou de dépassement, AWS IoT SiteWise ne produit aucun point de données pour ce calcul.

AWS IoT SiteWise ne produit pas non plus de point de données s'il calcule une valeur non numérique à la suite d'une expression de formule. Cela signifie que si vous définissez une formule qui calcule une chaîne, un tableau ou la [constante none](#), elle AWS IoT SiteWise ne produit aucun point de données pour ce calcul.

Exemple Exemples

Chacune des expressions de formule suivantes génère une valeur que AWS IoT SiteWise ne peut pas être représentée sous forme de nombre. AWS IoT SiteWise ne produit pas de point de données lorsqu'il calcule ces expressions de formule.

- $x / 0$ n'est pas défini.
- $\log(0)$ n'est pas défini.
- $\sqrt{-1}$ n'est pas défini dans un système de nombres réels.
- "hello" + " world" est une chaîne.
- `jp({'values':[3,6,7]}, '$.values')` est un tableau.
- `if(gte(temp, 300), temp, none)` c'est none quand temp est inférieur à 300.

Création de modèles composites personnalisés (composants)

Les modèles composites personnalisés, ou composants si vous utilisez la console, fournissent un autre niveau d'organisation pour vos modèles d'actifs et vos modèles de composants. Vous pouvez les utiliser pour structurer vos modèles en regroupant les propriétés ou en référençant d'autres modèles. Pour plus d'informations sur l'utilisation de modèles composites personnalisés, consultez [Modèles composites personnalisés \(composants\)](#).

Vous créez un modèle composite personnalisé au sein d'un modèle d'actif ou d'un modèle de composant existant. Il existe deux types de modèles composites personnalisés. Pour regrouper les propriétés associées au sein d'un modèle, vous pouvez créer un modèle composite personnalisé en ligne. Pour référencer un modèle de composant dans votre modèle d'actif ou de composant, vous pouvez créer un modèle composite component-model-based personnalisé.

Les sections suivantes décrivent comment utiliser l' AWS IoT SiteWise API pour créer des modèles composites personnalisés.

Rubriques

- [Création d'un composant en ligne \(console\)](#)
- [Création d'un modèle composite personnalisé en ligne \(AWS CLI\)](#)
- [Création d'un component-model-based composant \(console\)](#)
- [Création d'un modèle composite component-model-based personnalisé \(AWS CLI\)](#)

Création d'un composant en ligne (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour créer un composant en ligne qui définit ses propres propriétés.

Note

Comme il s'agit d'un composant intégré, ces propriétés ne s'appliquent qu'au modèle d'actif actuel et ne sont partagées nulle part ailleurs.

Si vous devez produire un modèle réutilisable (par exemple, pour le partager entre plusieurs modèles d'actifs ou pour inclure plusieurs instances au sein d'un même modèle d'actif), vous devez plutôt créer un composant basé sur un modèle de composant. Consultez la section suivante pour plus de détails.

Pour créer un composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle d'actif auquel vous souhaitez ajouter un composant.
4. Dans l'onglet Propriétés, sélectionnez Composants.
5. Choisissez Créer un composant.
6. Sur la page Créer un composant, procédez comme suit :
 - a. Entrez un nom pour le composant, tel que **ServoMotor** ou **ServoMotor Model1**. Ce nom doit être unique pour tous les composants de votre compte dans cette région.
 - b. (Facultatif) Ajoutez des définitions d'attributs pour le modèle. Les attributs représentent des informations qui changent rarement. Pour de plus amples informations, veuillez consulter [Définir des données statiques \(attributs\)](#).
 - c. (Facultatif) Ajoutez des définitions de mesures pour le modèle. Les mesures représentent des flux de données provenant de votre équipement. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).
 - d. (Facultatif) Ajoutez des définitions de transformations pour le modèle. Les transformations sont des formules qui font correspondre les données d'un formulaire à un autre. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).
 - e. (Facultatif) Ajoutez des définitions de métriques pour le modèle. Les métriques sont des formules qui regroupent les données sur des intervalles de temps. Les métriques peuvent saisir des données provenant des actifs associés, afin que vous puissiez calculer des valeurs qui représentent votre activité ou un sous-ensemble de celle-ci. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).
 - f. Choisissez Créer un composant.

Création d'un modèle composite personnalisé en ligne (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer un modèle composite personnalisé en ligne qui définit ses propres propriétés.

Utilisez cette [CreateAssetModelCompositeModel](#) opération pour créer un modèle en ligne avec des propriétés. Cette opération attend une charge utile avec la structure suivante.

Note

Comme il s'agit d'un modèle composite en ligne, ces propriétés ne s'appliquent qu'au modèle d'actif actuel et ne sont partagées nulle part ailleurs. Ce qui le rend « intégré », c'est qu'il ne fournit pas de valeur pour le `composedAssetModelId` champ.

Si vous devez produire un modèle réutilisable (par exemple, pour le partager entre plusieurs modèles d'actifs ou pour inclure plusieurs instances au sein d'un même modèle d'actif), vous devez plutôt créer un modèle `component-model-basedcomposite`. Consultez la section suivante pour plus de détails.

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "assetModelCompositeModelProperties": [
    {
      "dataType": "DOUBLE",
      "name": "Servo Motor Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "Celsius"
    },
    {
      "dataType": "DOUBLE",
      "name": "Spindle speed",
      "type": {
        "measurement": {}
      },
      "unit": "rpm"
    }
  ]
}
```

Création d'un component-model-based composant (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour créer un composant à partir d'un modèle de composant.

Pour créer un component-model-based composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle d'actif auquel vous souhaitez ajouter un composant.
4. Dans l'onglet Propriétés, sélectionnez Composants.
5. Choisissez Créer un composant.
6. Sur la page Créer un composant, procédez comme suit :
 - a. Sélectionnez le modèle de composant sur lequel vous souhaitez baser le composant.
 - b. Entrez un nom pour le composant, tel que **ServoMotor** ou **ServoMotor Model**. Ce nom doit être unique pour tous les composants de votre compte dans cette région.
 - c. Choisissez Créer un composant.

Création d'un modèle composite component-model-based personnalisé (AWS CLI)

Vous pouvez utiliser le AWS CLI pour créer un modèle composite component-model-based personnalisé au sein de votre modèle d'actif. Un modèle composite component-model-based personnalisé est une référence à un modèle de composant que vous avez déjà défini ailleurs.

Utilisez cette [CreateAssetModelCompositeModel](#) opération pour créer un modèle composite component-model-based personnalisé. Cette opération attend une charge utile avec la structure suivante.

Note

Dans cet exemple, la valeur de `composedAssetModelId` est l'ID du modèle d'actif ou l'ID externe d'un modèle de composant existant. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise . Pour un exemple de création d'un modèle de composant, consultez [Création d'un modèle de composant \(AWS CLI\)](#).

```
{
  "assetModelCompositeModelName": "CNCLathe_ServoMotorA",
  "assetModelCompositeModelType": "CUSTOM",
  "composedAssetModelId": component model ID
```

]

Comme il ne s'agit que d'une référence, un modèle composite component-model-based personnalisé ne possède aucune propriété propre, à part un nom.

Si vous souhaitez ajouter plusieurs instances du même composant à votre modèle d'actif (par exemple, une machine CNC dotée de plusieurs servomoteurs), vous pouvez ajouter plusieurs modèles composites component-model-based personnalisés portant chacun leur propre nom mais faisant tous référence au même `composedAssetModelId`.

Vous pouvez imbriquer des composants dans d'autres composants. Pour ce faire, vous pouvez ajouter un modèle component-model-based composite, comme indiqué dans cet exemple, à l'un de vos modèles de composants.

Créez des actifs pour les modèles d'actifs dans AWS IoT SiteWise

Vous pouvez créer une ressource à partir d'un modèle de ressource. Vous devez disposer d'un modèle de ressource avant de pouvoir créer une ressource. Si vous n'avez pas encore créé de modèle de ressource, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

Note

Vous pouvez uniquement créer des ressources à partir de modèles ACTIVE. Si l'état de votre modèle n'est pas ACTIVE, vous devrez peut-être attendre quelques minutes avant de pouvoir créer des ressources à partir de ce modèle. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Rubriques

- [Création d'un actif \(console\)](#)
- [Création d'un actif \(AWS CLI\)](#)
- [Configuration d'un nouvel actif](#)

Création d'un actif (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour créer un actif.

Pour créer une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez Create asset (Créer une ressource).
4. Sur la page Créer une ressource, procédez comme suit :
 - a. Pour Modèle, choisissez le modèle de ressource à partir duquel créer une ressource.

Note

Si votre modèle n'est pas ACTIF, vous devez attendre qu'il soit actif ou résoudre des problèmes s'il indique ÉCHEC.

- b. Saisissez un nom pour votre ressource.
- c. (Facultatif) Ajoutez des balises pour votre ressource. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
- d. Choisissez Create asset (Créer une ressource).

Lorsque vous créez un actif, la AWS IoT SiteWise console accède à la page du nouvel actif. Sur cette page, vous pouvez voir l'état de la ressource, qui est initialement CRÉATION. Cette page est automatiquement mise à jour, de sorte que vous pouvez attendre la mise à jour de l'état de la ressource.

Note

Le processus de création des ressources peut prendre jusqu'à une minute. Une fois que le statut est ACTIF, vous pouvez effectuer des opérations de mise à jour sur votre actif. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Après avoir créé une ressource, veuillez consulter [Configuration d'un nouvel actif](#).

Création d'un actif (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour créer un actif à partir d'un modèle d'actif.

Vous devez avoir un `assetModelId` pour créer une ressource. Si vous avez créé un modèle d'actif, mais que vous ne le connaissez pas `assetModelId`, utilisez l'[ListAssetModels](#) API pour afficher tous vos modèles d'actifs.

Pour créer un actif à partir d'un modèle d'actif, utilisez l'[CreateAsset](#) API avec les paramètres suivants :

- `assetName`— Le nom du nouvel actif. Donnez un nom à votre actif pour vous aider à l'identifier.
- `assetModelId`— L'ID de l'actif. Il s'agit de l'identifiant réel au format UUID, ou du `externalId:myExternalId` s'il en possède un. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour créer un actif (AWS CLI)

- Exécutez la commande suivante pour créer une ressource. *asset-name* Remplacez-le par le nom de l'actif et *asset-model-id* par l'ID ou l'ID externe du modèle d'actif.

```
aws iotsitewise create-asset \  
  --asset-name asset-name \  
  --asset-model-id asset-model-id
```

L'opération renvoie une réponse qui contient les détails de votre nouvelle ressource et son état au format suivant.

```
{  
  "assetId": "String",  
  "assetArn": "String",  
  "assetStatus": {  
    "state": "String",  
    "error": {  
      "code": "String",  
      "message": "String"  
    }  
  }  
}
```

L'état de la ressource est `CREATING` jusqu'à ce que la ressource soit créée.

 Note

Le processus de création des ressources peut prendre jusqu'à une minute. Pour vérifier l'état de votre actif, utilisez l'[DescribeAsset](#) opération avec l'ID de votre actif comme `assetId` paramètre. Une fois que l'actif `state` est en `ACTIVE` place, vous pouvez effectuer des opérations de mise à jour sur celui-ci. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Après avoir créé une ressource, veuillez consulter [Configuration d'un nouvel actif](#).

Configuration d'un nouvel actif

Après avoir créé un actif dans AWS IoT SiteWise, vous pouvez suivre plusieurs étapes pour utiliser pleinement l'actif et ses données. Ces étapes peuvent inclure la configuration de flux de données pour ingérer les données de l'actif, la configuration d'alarmes et de notifications pour surveiller les performances de l'actif, la création de visualisations et de tableaux de bord pour afficher les données de l'actif, et l'intégration de l'actif à d'autres AWS services ou applications tierces pour une analyse ou une automatisation plus poussées.

Terminez la configuration de votre actif à l'aide des actions facultatives suivantes :

- [Gérez les flux de données pour AWS IoT SiteWise](#) si votre ressource possède des propriétés de mesure.
- [Mettre à jour les valeurs des attributs](#) si la ressource comporte des valeurs d'attribut uniques.
- [Associer et dissocier des actifs](#) si votre ressource est une ressource parent.

Rechercher des actifs sur Console AWS IoT SiteWise

Utilisez la fonctionnalité Console AWS IoT SiteWise de recherche pour trouver des actifs en fonction des métadonnées et des filtres de valeur des propriétés en temps réel.

Prérequis

AWS IoT SiteWise nécessite des autorisations d'intégration AWS IoT TwinMaker afin de mieux organiser et modéliser les données industrielles. Si vous avez accordé des autorisations à AWS IoT SiteWise, utilisez l'[ExecuteQuery](#) API. Si vous n'avez pas accordé d'AWS IoT SiteWise autorisations

et que vous avez besoin d'aide pour démarrer, consultez [Intégrer AWS IoT SiteWise et AWS IoT TwinMaker](#).

Recherche avancée sur Console AWS IoT SiteWise

Recherche de métadonnées

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, sélectionnez Recherche avancée sous Ressources.
3. Sous Recherche avancée, choisissez l'option de recherche par métadonnées.
4. Renseignez les paramètres. Remplissez autant de champs que possible pour une recherche efficace.
 - a. Nom de la ressource : entrez le nom complet de la ressource ou un nom partiel pour une recherche étendue.
 - b. Nom de la propriété : entrez le nom complet de la propriété ou un nom partiel pour une recherche étendue.
 - c. Opérateur — Choisissez un opérateur parmi :
 - =
 - <
 - >
 - <=
 - >=
 - d. Valeur de la propriété : cette valeur est comparée à la dernière valeur de la propriété.
 - e. Type de valeur de propriété : type de données de la propriété. Choisissez parmi les options suivantes :
 - Double*
 - Integer
 - String
 - Booléen
5. Choisissez Rechercher.
6. Dans le tableau des résultats de recherche, sélectionnez l'actif dans la colonne Nom. Cela vous amène à la page détaillée de l'actif en question.

Assets

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Advanced search

Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search | Query builder

Asset name: Level-2 | Property name: power_max | Operator: > | Property value: 20 | Property value type: Double

Clear | Search

Search results (2)

Name	Asset id	Description
Level-2-asset-1	d0e9019b-9c38-4316-b574-38317aa38143	
Level-2-asset-2	b9c0d2fc-1527-42ce-8ba2-d1a4e8ff43de	Example description

Recherche partielle

Il n'est pas nécessaire de fournir tous les paramètres pour une recherche de ressources. Voici quelques exemples de recherches partielles à l'aide de l'option de recherche de métadonnées :

- Trouvez les actifs par leur nom :
 - Entrez une valeur uniquement dans le champ Nom de l'actif.
 - Les champs Nom de la propriété et Valeur de la propriété sont vides.
- Trouvez des actifs contenant des propriétés portant un nom spécifique :
 - Entrez une valeur uniquement dans le champ Nom de la propriété.
 - Les champs Nom de l'actif et Valeur de la propriété sont vides.
- Trouvez des actifs en fonction des dernières valeurs de leurs propriétés :
 - Entrez des valeurs dans les champs Nom de la propriété et Valeur de la propriété.
 - Sélectionnez un opérateur et un type de valeur de propriété.

Recherche dans le générateur de requêtes

1. Accédez à Console AWS IoT SiteWise.

2. Dans le volet de navigation, sélectionnez Recherche avancée sous Ressources.
3. Sous Recherche avancée, choisissez l'option Générateur de requêtes.
4. Dans le volet Générateur de requêtes, écrivez votre requête SQL pour récupérer un `asset_name` `asset_id` et `asset_description`.
5. Choisissez Rechercher.
6. Dans le tableau des résultats de recherche, sélectionnez l'actif dans la colonne Nom. Cela vous amène à la page détaillée de l'actif en question.

Assets

Assets represent Industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Advanced search

Use advanced search to find assets based on specific metadata. In addition, you can enter SQL queries directly in the query builder.

Metadata search | **Query builder**

Query builder

```
SELECT a.asset_id, a.asset_name, a.asset_description
FROM asset a, asset_property p, latest_value_time_series ts
WHERE a.asset_name LIKE '%asset-2%' AND a.property_name = 'temperature_f' AND ts.double_value > 50.0
```

Clear Search

Search results (2)

Name	Asset id	Description
Level-2a-asset-2	4fed596d-e903-4338-86db-34ca9301233a	Generator #3
Level-2b-asset-2	b4ac2b24-4fce-4a72-9fea-ef6d0f741e8d	Generator #2

Note

- La SELECT clause de la requête SQL doit inclure les `asset_id` champs `asset_name` et pour garantir la validité d'un actif dans la table des résultats de recherche.

- Le générateur de requêtes affiche uniquement le nom, l'identifiant de l'actif et la description dans le tableau des résultats. L'ajout de champs supplémentaires à la SELECT clause n'ajoute pas de colonnes supplémentaires à la table de résultats

Mettre à jour les valeurs des attributs

Les ressources héritent des attributs de leur modèle de ressource, y compris la valeur par défaut de l'attribut. Dans certains cas, il vaudra mieux conserver l'attribut par défaut du modèle de ressource, par exemple pour une propriété de fabricant de ressource. Dans d'autres cas, il vaudra mieux actualiser l'attribut hérité, par exemple pour la latitude et la longitude spécifiques d'une ressource.

Updating an attribute value (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour la valeur d'une propriété d'actif attributaire.

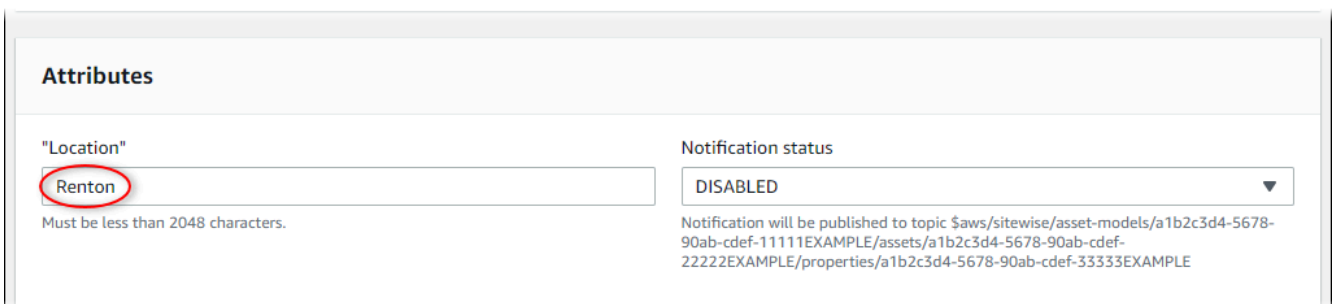
Pour mettre à jour la valeur d'un attribut (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour laquelle vous souhaitez mettre à jour un attribut.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Recherchez l'attribut à mettre à jour, puis saisissez sa nouvelle valeur.



Attributes

"Location"

Renton

Must be less than 2048 characters.

Notification status

DISABLED

Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE

6. Choisissez Enregistrer.

Updating an attribute value (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour mettre à jour la valeur d'un attribut.

Vous devez connaître l'`assetId` de votre ressource et le `propertyId` de la propriété pour effectuer cette procédure. Vous pouvez également utiliser l'identifiant externe. Si vous avez créé un actif et que vous ne le connaissez pas `assetId`, utilisez l'[ListAssets](#) API pour répertorier tous les actifs d'un modèle spécifique. Utilisez cette [DescribeAsset](#) opération pour afficher les propriétés de votre actif, y compris les propriétés IDs.

Utilisez l'opération [BatchPutAssetPropertyValue](#) pour affecter des valeurs d'attributs à votre ressource. Vous pouvez utiliser cette opération pour définir plusieurs attributs à la fois. La charge utile de cette opération contient une liste d'entrées, chacune contenant l'ID de ressource, l'ID de propriété et la valeur d'attribut.

Pour mettre à jour la valeur d'un attribut (AWS CLI)

1. Créez un fichier nommé `batch-put-payload.json` et copiez l'objet JSON suivant dans le fichier. Cet exemple de charge utile montre comment définir la latitude et la longitude d'une éolienne. Mettez à jour IDs les valeurs et les horodatages pour modifier la charge utile en fonction de votre cas d'utilisation.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-3333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
```

```

    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
    "propertyValues": [
      {
        "value": {
          "doubleValue": 122.3491
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        }
      }
    ]
  }
]
}

```

- Chaque entrée de la charge utile contient un `entryId` que vous pouvez définir sous la forme d'une chaîne unique. Si des entrées de demande échouent, chaque erreur contiendra l'`entryId` de la demande correspondante afin que vous sachiez quelles demandes réessayer.
- Pour définir une valeur d'attribut, vous pouvez inclure une structure timestamp-quality-value (TQV) dans la liste de chaque propriété `propertyValues` d'attribut. Cette structure doit contenir le nouveau `value` et le `timestamp` actuel.
 - `value`— Structure contenant l'un des champs suivants, en fonction du type de propriété défini :
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
 - `timestamp`— Une structure qui contient l'heure actuelle d'Unix en secondes,. `timeInSeconds` AWS IoT SiteWise rejette tous les points de données dont l'horodatage existait depuis plus de 7 jours ou moins de 5 minutes dans le futur.

Pour de plus amples informations sur la préparation d'une charge utile pour [BatchPutAssetPropertyValue](#), veuillez consulter [Ingérez des données avec AWS IoT SiteWise APIs](#).

2. Exécutez la commande suivante pour envoyer les valeurs d'attribut à AWS IoT SiteWise :

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

Associer et dissocier des actifs

Si le modèle de votre ressource définit des hiérarchies de modèles de ressources enfants, vous pouvez associer des ressources enfants à votre ressource. Les ressources parents peuvent accéder aux données des ressources associées et les agréger. Pour de plus amples informations sur les modèles de ressources hiérarchiques, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).

Rubriques

- [Associer et dissocier des actifs \(console\)](#)
- [Associer et dissocier des actifs \(AWS CLI\)](#)

Associer et dissocier des actifs (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour associer et dissocier des actifs.

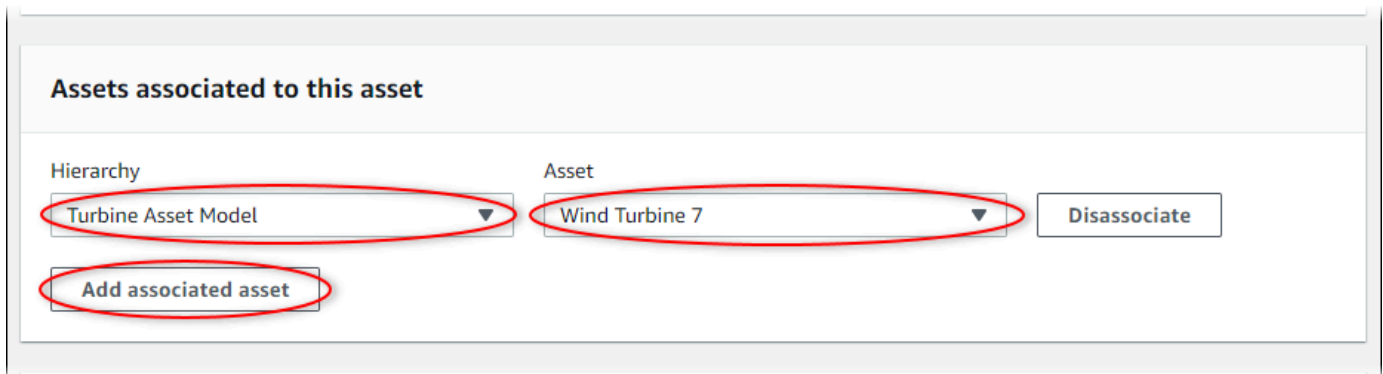
Pour associer une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource parent à laquelle vous souhaitez associer une ressource enfant.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Dans Ressources associées à cette ressource, choisissez Ajouter une ressource associée.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼

Asset: Wind Turbine 7 ▼

Disassociate

Add associated asset

6. Pour Hiérarchie, choisissez la hiérarchie qui définit la relation entre la ressource parent et la ressource enfant.
7. Pour Ressource, choisissez la ressource enfant à associer.
8. Choisissez Enregistrer.

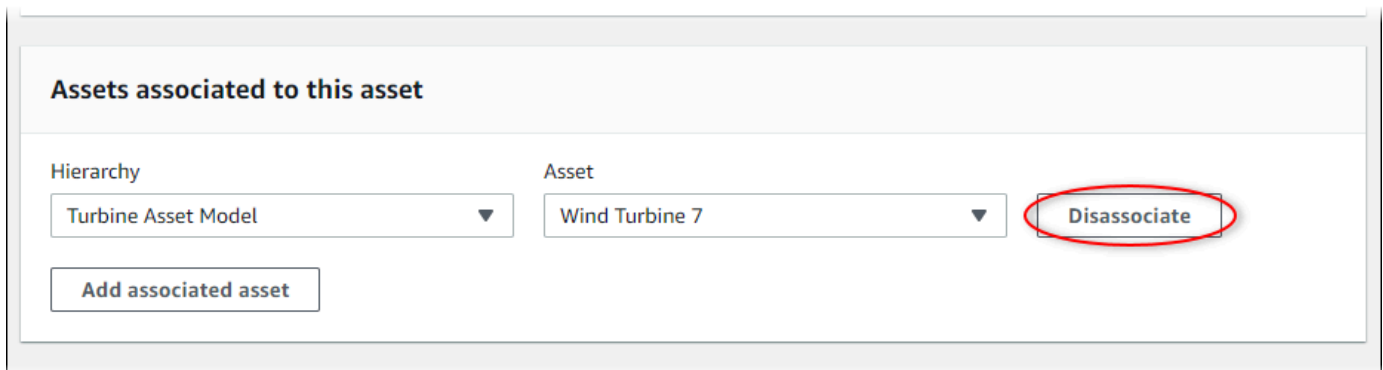
Pour dissocier une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource parent pour laquelle vous souhaitez dissocier une ressource enfant.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Dans Ressources associées à cette ressource, choisissez Dissocier pour la ressource.



Assets associated to this asset

Hierarchy: Turbine Asset Model ▼

Asset: Wind Turbine 7 ▼

Disassociate

Add associated asset

6. Choisissez Enregistrer.

Associer et dissocier des actifs ()AWS CLI

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour associer et dissocier des actifs.

Pour cette procédure, vous devez connaître l'ID de la hiérarchie (`hierarchyId`) dans le modèle de ressource parent qui définit la relation avec le modèle de ressource enfant. Utilisez l'[DescribeAsset](#) opération pour trouver l'ID de hiérarchie dans la réponse.

Pour rechercher un ID de hiérarchie

- Exécutez la commande suivante pour décrire la ressource parent. Remplacez *parent-asset-id* par l'ID de la ressource parent ou par l'ID externe.

```
aws iotsitewise describe-asset --asset-id parent-asset-id
```

L'opération renvoie une réponse qui contient les détails de la ressource. La réponse contient une `assetHierarchies` liste dont la structure est la suivante :

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

L'ID de hiérarchie est la valeur `id` d'une hiérarchie dans la liste des hiérarchies de ressources.

Une fois que vous avez l'ID de hiérarchie, vous pouvez associer ou dissocier une ressource à/de cette hiérarchie.

Pour associer un actif enfant à un actif parent, utilisez l'[AssociateAssets](#) opération. Pour dissocier un actif enfant d'un actif parent, utilisez l'[DisassociateAssets](#) opération. Spécifiez les paramètres suivants, qui sont les mêmes pour les deux opérations :

- `assetId`— L'ID de l'actif parent ou l'ID externe.

- `hierarchyId`— L'ID de hiérarchie ou l'ID externe de la ressource parent.
- `childAssetId`— L'identifiant ou l'identifiant externe de l'actif enfant.

Pour associer un actif (AWS CLI)

- Exécutez la commande suivante pour associer une ressource enfant à une ressource parent. Remplacez *parent-asset-id* *hierarchy-id*, et *child-asset-id* par le correspondant IDs :

```
aws iotsitewise associate-assets \  
  --asset-id parent-asset-id \  
  --hierarchy-id hierarchy-id \  
  --child-asset-id child-asset-id
```

Pour dissocier un actif (AWS CLI)

- Exécutez la commande suivante pour dissocier une ressource enfant d'une ressource parent. Remplacez *parent-asset-id* *hierarchy-id*, et *child-asset-id* par le correspondant IDs :

```
aws iotsitewise disassociate-assets \  
  --asset-id parent-asset-id \  
  --hierarchy-id hierarchy-id \  
  --child-asset-id child-asset-id
```

Mettre à jour les actifs et les modèles

Vous pouvez mettre à jour vos actifs, modèles d'actifs et modèles de composants AWS IoT SiteWise pour modifier leurs noms et leurs définitions. Ces opérations de mise à jour sont asynchrones et leur propagation prend du temps. AWS IoT SiteWise Vérifiez l'état de l'actif ou du modèle avant d'apporter des modifications supplémentaires. Vous devez attendre que les modifications soient propagées avant de continuer à utiliser la ressource ou le modèle mis à jour.

Rubriques

- [Mettre à jour les actifs dans AWS IoT SiteWise](#)
- [Mettre à jour les modèles d'actifs et les modèles de composants](#)

- [Mettre à jour des modèles composites personnalisés \(composants\)](#)
- [Verrouillage optimiste pour les écrits sur le modèle d'actifs](#)

Mettre à jour les actifs dans AWS IoT SiteWise

Vous pouvez utiliser la AWS IoT SiteWise console ou l'API pour mettre à jour le nom d'un actif.

Lorsque vous mettez à jour une ressource, son statut est maintenu UPDATING jusqu'à ce que les modifications se propagent. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Rubriques

- [Mettre à jour un actif \(console\)](#)
- [Mettre à jour un actif \(AWS CLI\)](#)

Mettre à jour un actif (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour les détails des actifs.

Pour mettre à jour une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource à mettre à jour.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Mettez à jour le nom de la ressource.
6. (Facultatif) Sur cette page, mettez à jour les autres informations relatives à la ressource. Pour plus d'informations, consultez les ressources suivantes :
 - [Gérez les flux de données pour AWS IoT SiteWise](#)
 - [Mettre à jour les valeurs des attributs](#)

- [Interagissez avec d'autres AWS services](#)

7. Choisissez Enregistrer.

Mettre à jour un actif (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour mettre à jour le nom d'un actif.

Utilisez cette [UpdateAsset](#) opération pour mettre à jour un actif. Spécifiez les paramètres suivants :

- `assetId`— L'ID de l'actif. Il s'agit de l'identifiant réel au format UUID, ou du `externalId:myExternalId` s'il en possède un. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .
- `assetName`— Le nouveau nom de l'actif.

Pour mettre à jour le nom d'un actif (AWS CLI)

- Exécutez la commande suivante pour mettre à jour le nom d'une ressource. Remplacez *asset-id* par l'ID ou l'ID externe de la ressource. Mettez à jour le *asset-name* avec le nouveau nom de l'actif.

```
aws iotsitewise update-asset \  
  --asset-id asset-id \  
  --asset-name asset-name
```

Mettre à jour les modèles d'actifs et les modèles de composants

Utilisez la AWS IoT SiteWise console ou l'API pour mettre à jour un modèle d'actif ou un modèle de composant.

Vous ne pouvez pas modifier le type ou le type de données d'une propriété existante, ni la fenêtre d'une métrique existante. Vous ne pouvez pas non plus modifier le type du modèle d'actif en modèle de composant, ou inversement.

Important

- Si vous supprimez une propriété d'un modèle de ressource ou d'un modèle de composant, toutes les AWS IoT SiteWise données précédentes relatives à cette propriété sont

supprimées. Pour les modèles de composants, cela concerne tous les modèles d'actifs utilisant ce modèle de composant. Veillez donc particulièrement à comprendre dans quelle mesure votre modification peut s'appliquer.

- Si vous supprimez une définition de hiérarchie d'un modèle d'actifs, AWS IoT SiteWise dissocie tous les actifs de cette hiérarchie.

Lorsque vous mettez à jour un modèle de ressource, chaque ressource basée sur ce modèle reflète toutes les modifications que vous apportez au modèle sous-jacent. Jusqu'à ce que les modifications se propagent, l'état de chaque ressource est **UPDATING**. Attendez que l'état des ressources passe à **ACTIVE** pour pouvoir interagir avec elles. Pendant ce temps, le statut du modèle de ressource mis à jour est **PROPAGATING**.

Lorsque vous mettez à jour un modèle de composant, chaque modèle d'actif qui intègre ce modèle de composant reflète les modifications. Jusqu'à ce que les modifications du modèle de composant se propagent, chaque modèle d'actif concerné possède l'**UPDATING** état, suivi **PROPAGATING** de la mise à jour de ses actifs associés, comme décrit dans le paragraphe précédent. Vous devez attendre que ces modèles d'actifs reviennent à leur **ACTIVE** état normal avant d'interagir avec eux. Pendant ce temps, le statut du modèle de composant mis à jour sera **PROPAGATING**.

Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#).

Rubriques

- [Mettre à jour un modèle d'actif ou de composant \(console\)](#)
- [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#)

Mettre à jour un modèle d'actif ou de composant (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour un modèle de ressource ou un modèle de composant.

Pour mettre à jour un modèle d'actif ou un modèle de composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez **Models (Modèles)**.
3. Choisissez le modèle d'actif ou le modèle de composant à mettre à jour.
4. Choisissez **Modifier**.

5. Sur la page Modifier le modèle, effectuez l'une des opérations suivantes :
 - Dans Informations relatives au modèle, modifiez le nom du modèle.
 - Modifiez l'une des définitions d'attribut. Vous ne pouvez pas modifier le type de données des attributs existants. Pour de plus amples informations, veuillez consulter [Définir des données statiques \(attributs\)](#).
 - Modifiez l'une des définitions de mesure. Vous ne pouvez pas modifier le type de données des mesures existantes. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).
 - Modifiez l'une des définitions de transformation. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).
 - Modifiez l'une des définitions de métrique. Vous ne pouvez pas modifier l'intervalle de temps des métriques existantes. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).
 - (Modèles d'actifs uniquement) Modifiez l'une des définitions de hiérarchie. Vous ne pouvez pas modifier le modèle de hiérarchie des hiérarchies existantes. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).
6. Choisissez Enregistrer.

 Note

Les demandes de mise à jour effectuées dans la console sont rejetées si un autre utilisateur met à jour le modèle de ressource avec succès depuis que vous avez ouvert la page Modifier le modèle pour la dernière fois. La console invite l'utilisateur à actualiser la page Modifier le modèle pour récupérer le modèle mis à jour. Vous devez effectuer à nouveau vos mises à jour, puis réessayer de sauvegarder. Pour plus d'informations, consultez [Verrouillage optimiste pour les écrits sur le modèle d'actifs](#).

Mettre à jour un modèle d'actif ou de composant (AWS CLI)

Utilisez le AWS Command Line Interface (AWS CLI) pour mettre à jour un modèle d'actif ou un modèle de composant.

Utilisez l'[UpdateAssetModel](#) API pour mettre à jour le nom, la description et les propriétés d'un modèle d'actif ou d'un modèle de composant. Pour les modèles d'actifs uniquement, vous pouvez mettre à jour les hiérarchies. Spécifiez les paramètres suivants :

- `assetModelId`— L'ID de l'actif. Il s'agit de l'identifiant réel au format UUID, ou du `externalId:myExternalId` s'il en possède un. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Spécifiez le modèle mis à jour dans la charge utile. Pour en savoir plus sur le format attendu d'un modèle d'actif ou d'un modèle de composant, voir [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

 Warning

L'[UpdateAssetModel](#) API remplace le modèle existant par le modèle que vous fournissez dans la charge utile. Pour éviter de supprimer les propriétés ou les hiérarchies de votre modèle, vous devez inclure leurs définitions IDs et leurs propriétés dans la charge utile du modèle mise à jour. Pour savoir comment interroger la structure existante de votre modèle, consultez l'[DescribeAssetModel](#) opération.

 Note

La procédure suivante ne permet de mettre à jour que les modèles composites de type `AWS/ALARM`. Si vous souhaitez mettre à jour des modèles `CUSTOM` composites, utilisez [UpdateAssetModelCompositeModel](#) plutôt. Pour de plus amples informations, veuillez consulter [Mettre à jour des modèles composites personnalisés \(composants\)](#).

Pour mettre à jour un modèle d'actif ou un modèle de composant (AWS CLI)

1. Exécutez la commande suivante pour récupérer la définition du modèle existant. `asset-model-id` Remplacez-le par l'ID ou l'ID externe du modèle d'actif ou du modèle de composant à mettre à jour.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

La commande ci-dessus renvoie la définition du modèle correspondant à la dernière version du modèle.

Dans le cas d'utilisation où un modèle d'actif est dans un `FAILED` état, récupérez la définition de modèle valide correspondant à sa version active pour créer votre demande de mise à jour.

Consultez [Versions du modèle d'actifs](#) pour plus de détails. Exécutez la commande suivante pour récupérer la définition du modèle actif :

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id --asset-model-version ACTIVE
```

L'opération renvoie une réponse contenant les détails du modèle. La réponse présente la structure suivante.

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel,
  "assetModelCompositeModelSummaries": Array of AssetModelCompositeModelSummary,
  "assetModelCreationDate": "String",
  "assetModelLastUpdateDate": "String",
  "assetModelStatus": {
    "state": "String",
    "error": {
      "code": "String",
      "message": "String"
    }
  },
  "assetModelType": "String",
  "assetModelVersion": "String",
  "eTag": "String"
}
```

Pour plus d'informations, consultez l'[DescribeAssetModel](#) opération.

2. Créez un fichier appelé `update-asset-model.json` et copiez la réponse de la commande précédente dans le fichier.
3. Supprimez les paires clé-valeur suivantes de l'objet JSON dans `update-asset-model.json` :
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCompositeModelSummaries`

- `assetModelCreationDate`
- `assetModelLastUpdateDate`
- `assetModelStatus`
- `assetModelType`
- `assetModelVersion`
- `eTag`

L'[UpdateAssetModel](#) opération attend une charge utile dont la structure est la suivante :

```
{
  "assetModelName": "String",
  "assetModelDescription": "String",
  "assetModelProperties": Array of AssetModelProperty,
  "assetModelHierarchies": Array of AssetModelHierarchyDefinition,
  "assetModelCompositeModels": Array of AssetModelCompositeModel
}
```

4. Dans `update-asset-model.json`, effectuez l'une des actions suivantes :
 - Modifiez le nom du modèle de ressource (`assetModelName`).
 - Modifiez, ajoutez ou supprimez la description du modèle de ressource (`assetModelDescription`).
 - Modifiez, ajoutez ou supprimez les propriétés du modèle de ressource (`assetModelProperties`). Vous ne pouvez pas modifier le `dataType` des propriétés existantes ou le `window` des métriques existantes. Pour de plus amples informations, veuillez consulter [Définir les propriétés des données](#).
 - Modifiez, ajoutez ou supprimez les hiérarchies du modèle de ressource (`assetModelHierarchies`). Vous ne pouvez pas modifier le `childAssetModelId` des hiérarchies existantes. Pour de plus amples informations, veuillez consulter [Définir les hiérarchies des modèles d'actifs](#).
 - Modifiez, ajoutez ou supprimez l'un des modèles composites de type AWS/ALARM (`assetModelCompositeModels`) du modèle d'actifs. Les alarmes surveillent d'autres propriétés afin que vous puissiez identifier les équipements ou les processus nécessitant une attention particulière. Chaque définition d'alarme est un modèle composite qui normalise l'ensemble de propriétés utilisées par l'alarme. Pour plus d'informations, consultez [Surveillez](#)

[les données avec des alarmes intégrées AWS IoT SiteWise](#) et [Définissez des alarmes sur les modèles d'actifs dans AWS IoT SiteWise](#).

5. Exécutez la commande suivante pour mettre à jour le modèle de ressource avec la définition stockée dans `update-asset-model.json`. Remplacez `asset-model-id` par l'ID du modèle d'actif :

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --cli-input-json file://model-payload.json
```

Important

Lorsque plusieurs utilisateurs mettent à jour un modèle d'actif en même temps, les modifications apportées par un utilisateur peuvent être annulées par inadvertance par un autre utilisateur. Pour éviter cela, vous devez définir une demande de mise à jour conditionnelle. Consultez [Verrouillage optimiste pour les écrits sur le modèle d'actifs](#).

Mettre à jour des modèles composites personnalisés (composants)

Vous pouvez utiliser l' AWS IoT SiteWise API pour mettre à jour un modèle composite personnalisé ou la AWS IoT SiteWise console pour mettre à jour les composants.

Rubriques

- [Mettre à jour un composant \(console\)](#)
- [Mettre à jour un modèle composite personnalisé \(AWS CLI\)](#)

Mettre à jour un composant (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour un composant.

Pour mettre à jour un composant (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle d'actif dans lequel se trouve le composant.

4. Dans l'onglet Propriétés, sélectionnez Composants.
5. Choisissez le composant que vous souhaitez mettre à jour.
6. Choisissez Modifier.
7. Sur la page Modifier le composant, effectuez l'une des opérations suivantes :
 - Dans Informations relatives au modèle, modifiez le nom du modèle.
 - Modifiez l'une des définitions d'attribut. Vous ne pouvez pas modifier le type de données des attributs existants. Pour de plus amples informations, veuillez consulter [Définir des données statiques \(attributs\)](#).
 - Modifiez l'une des définitions de mesure. Vous ne pouvez pas modifier le type de données des mesures existantes. Pour de plus amples informations, veuillez consulter [Définir les flux de données provenant de l'équipement \(mesures\)](#).
 - Modifiez l'une des définitions de transformation. Pour de plus amples informations, veuillez consulter [Transformer les données \(transformations\)](#).
 - Modifiez l'une des définitions de métrique. Vous ne pouvez pas modifier l'intervalle de temps des métriques existantes. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).
8. Choisissez Enregistrer.

Mettre à jour un modèle composite personnalisé (AWS CLI)

Utilisez le AWS Command Line Interface (AWS CLI) pour mettre à jour un modèle composite personnalisé.

Pour mettre à jour le nom ou la description, utilisez l'[UpdateAssetModelCompositeModel](#) opération. Pour les modèles composites personnalisés en ligne uniquement, vous pouvez également mettre à jour les propriétés. Vous ne pouvez pas mettre à jour les propriétés d'un modèle composite component-model-based personnalisé, car son modèle de composant référencé fournit les propriétés associées.

Important

Si vous supprimez une propriété d'un modèle composite personnalisé, toutes les données précédentes relatives à cette propriété sont AWS IoT SiteWise supprimées. Vous ne pouvez pas modifier le type ou le type de données d'une propriété existante.

Pour remplacer une propriété de modèle composite existante par une nouvelle propriété identiquename, procédez comme suit :

1. Soumettez une `UpdateAssetModelCompositeModel` demande en supprimant l'intégralité de la propriété existante.
2. Soumettez une deuxième `UpdateAssetModelCompositeModel` demande qui inclut la nouvelle propriété. La nouvelle propriété de l'actif sera la même name que la précédente et AWS IoT SiteWise générera une nouvelle propriété `uniqueid`.

Pour mettre à jour un modèle composite personnalisé (AWS CLI)

1. Pour récupérer la définition du modèle composite existant, exécutez la commande suivante. Remplacez-le `composite-model-id` par l'ID ou l'ID externe du modèle composite personnalisé à mettre à jour, et `asset-model-id` par le modèle d'actif auquel le modèle composite personnalisé est associé. Pour plus d'informations, consultez le Guide de l'utilisateur AWS IoT SiteWise .

- a. Exécutez la commande ci-dessous :

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id
```

- b. La commande ci-dessus renvoie la définition du modèle composite correspondant à la dernière version du modèle associé. Dans le cas d'utilisation où un modèle d'actif est dans un FAILED état, récupérez la définition de modèle valide correspondant à sa version active pour créer votre demande de mise à jour. Consultez [Versions du modèle d'actifs](#) pour plus de détails.
- c. Exécutez la commande suivante pour récupérer la définition du modèle actif :

```
aws iotsitewise describe-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--asset-model-version ACTIVE
```

- d. Pour plus d'informations, consultez l'[DescribeAssetModelCompositeModel](#) opération.
2. Créez un fichier appelé `update-custom-composite-model.json`, puis copiez la réponse de la commande précédente dans le fichier.

3. Supprimez toutes les paires clé-valeur de l'objet JSON, à `update-custom-composite-model.json` l'exception des champs suivants :
 - `assetModelCompositeModelName`
 - `assetModelCompositeModelDescription`(si présent)
 - `assetModelCompositeModelProperties`(si présent)
4. Dans `update-custom-composite-model.json`, effectuez l'une des actions suivantes :
 - Modifiez la valeur de `assetModelCompositeModelName`.
 - Ajoutez `assetModelCompositeModelDescription`, supprimez ou modifiez sa valeur.
 - Pour les modèles composites personnalisés en ligne uniquement : modifiez, ajoutez ou supprimez l'une des propriétés du modèle d'actif dans `assetModelCompositeModelProperties`.

Pour plus d'informations sur le format requis pour ce fichier, consultez la syntaxe de demande pour [UpdateAssetModelCompositeModel](#).

5. Exécutez la commande suivante pour mettre à jour le modèle composite personnalisé avec la définition stockée dans `update-custom-composite-model.json`. *composite-model-id* Remplacez-le par l'ID du modèle composite et *asset-model-id* par l'ID du modèle d'actif dans lequel il se trouve.

```
aws iotsitewise update-asset-model-composite-model \  
--asset-model-composite-model-id composite-model-id \  
--asset-model-id asset-model-id \  
--cli-input-json file://update-custom-composite-model.json
```

Important

Lorsque plusieurs utilisateurs mettent à jour un modèle d'actif en même temps, les modifications apportées par un utilisateur peuvent être annulées par inadvertance par un autre utilisateur. Pour éviter cela, vous devez définir une demande de mise à jour conditionnelle. Consultez [Verrouillage optimiste pour les écrits sur le modèle d'actifs](#).

Verrouillage optimiste pour les écrits sur le modèle d'actifs

Lors de la mise à jour d'un modèle d'actif, l'utilisateur effectue les opérations suivantes :

1. Lisez la définition actuelle du modèle d'actifs.
2. Modifiez la définition du modèle d'actif avec les modifications requises.
3. Mettez à jour le modèle d'actif avec la nouvelle définition.

Dans un scénario où deux utilisateurs mettent à jour un modèle, les opérations suivantes sont possibles :

- L'utilisateur A lit la définition du modèle d'actif X.
- L'utilisateur B lit la définition du modèle d'actif X et valide les modifications, modifiant ainsi la définition de X.
- L'utilisateur A valide et remplace les modifications apportées par l'utilisateur B pour le modèle d'actif X, sans vérifier ni intégrer les modifications apportées par l'utilisateur B.

Le verrouillage optimiste est un mécanisme utilisé AWS IoT SiteWise pour empêcher les remplacements accidentels, comme dans le scénario ci-dessus. Le verrouillage optimiste est une stratégie visant à garantir que la version actuelle du modèle d'actif en cours de mise à jour ou de suppression est la même que sa version actuelle dans AWS IoT SiteWise. Cela empêche les écritures du modèle d'actifs d'être remplacées par des mises à jour accidentelles.

Procédez comme suit pour effectuer des écritures de modèles d'actifs avec un verrouillage optimiste :

Rubriques

- [Exécution d'écritures sur le modèle d'actifs avec un verrouillage optimiste \(console\)](#)
- [Exécution d'écritures sur le modèle d'actifs avec un verrou optimiste \(AWS CLI\)](#)

Exécution d'écritures sur le modèle d'actifs avec un verrouillage optimiste (console)

La procédure ci-dessous décrit comment effectuer des écritures de modèles d'actifs avec un verrou optimiste sur la version active du modèle d'actif dans la console.

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).

3. Choisissez le modèle d'actif ou le modèle de composant à mettre à jour.
4. Choisissez Modifier.
5. Apportez des modifications sur la page Modifier le modèle.
6. Choisissez Enregistrer.

 Note

Parfois, une ou plusieurs mises à jour réussies du modèle ont eu lieu entre le moment où l'utilisateur commence à modifier le modèle et celui où il enregistre les modifications apportées au modèle.

Pour éviter que l'utilisateur ne remplace accidentellement les nouvelles mises à jour réussies, l'écriture de l'utilisateur est rejetée. La console désactive le bouton Enregistrer et invite l'utilisateur à actualiser la page Modifier le modèle. L'utilisateur doit à nouveau mettre à jour la nouvelle version active du modèle. L'utilisateur doit effectuer les étapes supplémentaires suivantes :

7. Choisissez Refresh.
8. Suivez à nouveau les étapes 5 et 6.

Exécution d'écritures sur le modèle d'actifs avec un verrou optimiste (AWS CLI)

La procédure ci-dessous décrit comment effectuer des écritures de modèles d'actifs avec un verrouillage optimiste dans le AWS CLI.

1. Récupère la ETag définition du modèle actuel associée

ETag est un jeton unique généré pour chaque nouvelle représentation d'un modèle d'actif. Appelez [DescribeAssetModel](#) l'API pour récupérer la définition du modèle d'actif actuel et l'associer à ETag partir de la réponse.

Lors de mises à jour simultanées, les utilisateurs effectuent soit des mises à jour réussies (modèle dans ACTIVE l'état), soit des mises à jour infructueuses (modèle dans FAILED l'état). Pour éviter qu'un utilisateur ne remplace accidentellement une mise à jour réussie, vous devez récupérer la version active du modèle d'[Versions du modèle d'actifs](#) actif et en obtenir la ETag valeur.

Exécutez la commande suivante :

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id \
--asset-model-version ACTIVE
```

La réponse renvoie la structure suivante :

```
{
  "assetModelId": "String",
  "assetModelArn": "String",
  "assetModelName": "String",
  ...
  "eTag": "String"
}
```

 Note

Vous devez récupérer la dernière version du modèle d'actif et sa version ETag afin de ne pas remplacer les mises à jour.

2. Exécuter les opérations UPDATE et DELETE avec des conditions d'écriture

Les modèles d'actifs suivants APIs favorisent un verrouillage optimiste :

- [UpdateAssetModel](#)
- [DeleteAssetModel](#)
- [CreateAssetModelCompositeModel](#)
- [UpdateAssetModelCompositeModel](#)
- [DeleteAssetModelCompositeModel](#)

 Note

Les scénarios ci-dessous utilisent UpdateAssetModel l'API comme référence. Les conditions s'appliquent à toutes les opérations listées ci-dessus.

Les scénarios ci-dessous décrivent les différentes conditions d'écriture en fonction des exigences de contrôle de simultanéité :

- Exécutez la commande suivante afin de ne pas remplacer les mises à jour réussies. Aucune nouvelle version active ne doit exister depuis la dernière version active lue. Remplacez `e-tag` par le ETag renvoyé dans l'opération d'API utilisée lors de la lecture de la version active.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match e-tag \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Lorsque la création d'un modèle échoue, il n'existe pas encore de version active pour ce modèle, car il est dans un FAILED état. Il est toujours possible de remplacer une nouvelle version active déjà présente, avant que vos modifications ne soient validées. Exécutez la commande suivante pour ne pas remplacer une nouvelle version active lorsqu'aucune version active n'existe lors de votre dernière lecture.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-none-match "*" \  
  --match-for-version-type ACTIVE \  
  --cli-input-json file://model-payload.json
```

- Exécutez la commande suivante pour éviter de remplacer les mises à jour réussies ou infructueuses. Cette commande définit une condition d'écriture qui garantit qu'aucune dernière version n'est créée depuis votre dernière lecture de la dernière version. Remplacez `e-tag` par le ETag renvoyé dans l'opération d'API utilisée lors de la lecture de la version active.

```
aws iotsitewise update-asset-model \  
  --asset-model-id asset-model-id \  
  --if-match eTag \  
  --match-for-version-type LATEST \  
  --cli-input-json file://model-payload.json
```

Si la condition d'écriture est évaluée à `FALSE`, la demande d'écriture échoue avec `lePreconditionFailedException`.

Supprimer des actifs et des modèles dans AWS IoT SiteWise

Vous pouvez supprimer vos actifs et vos modèles une AWS IoT SiteWise fois que vous en avez terminé avec eux. Les opérations de suppression sont asynchrones et leur propagation prend du temps. AWS IoT SiteWise

Rubriques

- [Supprimer des actifs dans AWS IoT SiteWise](#)
- [Supprimer des modèles d'actifs dans AWS IoT SiteWise](#)

Supprimer des actifs dans AWS IoT SiteWise

Vous pouvez utiliser la AWS IoT SiteWise console ou l'API pour supprimer une ressource dont vous n'avez plus besoin dans votre environnement. La suppression d'un modèle de ressource entraîne également la suppression de tous les actifs et modèles de composants associés. Cependant, il est important de noter que la suppression d'un actif ou d'un modèle est une action permanente et que toutes les données associées aux ressources supprimées sont également supprimées. Avant de supprimer des actifs ou des modèles, il est recommandé de passer en revue les dépendances ou intégrations susceptibles d'être affectées et de vous assurer que vous disposez d'une sauvegarde de toutes les données importantes.

Avant de pouvoir supprimer une ressource, vous devez d'abord dissocier ses ressources enfants et la dissocier de sa ressource parent. Pour de plus amples informations, veuillez consulter [Associer et dissocier des actifs](#). Si vous utilisez le AWS Command Line Interface (AWS CLI), vous pouvez utiliser l'[ListAssociatedAssets](#) opération pour répertorier les enfants d'un actif.

Lorsque vous supprimez une ressource, son état est DELETING jusqu'à ce que les modifications soient propagées. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#). Une fois la ressource supprimée, vous ne pouvez plus l'interroger. Si vous le faites, l'API renvoie une réponse HTTP 404.

Important

AWS IoT SiteWise supprime toutes les données de propriété des actifs supprimés.

Rubriques

- [Supprimer un actif \(console\)](#)
- [Supprimer un actif \(AWS CLI\)](#)

Supprimer un actif (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour supprimer un actif.

Pour supprimer une ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource à supprimer.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Si la ressource comporte des ressources associées, supprimez chaque ressource. Vous pouvez choisir le nom d'une ressource pour accéder à sa page, où vous pouvez la supprimer.
5. Sur la page de la ressource, choisissez Supprimer.
6. Dans la boîte de dialogue Supprimer la ressource, procédez comme suit :
 - a. Saisissez **Delete** pour confirmer la suppression.
 - b. Sélectionnez Delete (Supprimer).

Supprimer un actif (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour supprimer un actif.

Utilisez cette [DeleteAsset](#) opération pour supprimer un actif. Spécifiez le paramètre suivant :

- `assetId`— L'ID de l'actif. Il s'agit de l'identifiant réel au format UUID, ou du `externalId:myExternalId` s'il en possède un. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour supprimer un actif (AWS CLI)

1. Exécutez la commande suivante pour répertorier les hiérarchies de la ressource. Remplacez *asset-id* par l'ID ou l'ID externe de l'actif :

```
aws iotsitewise describe-asset --asset-id asset-id
```

L'opération renvoie une réponse qui contient les détails de la ressource. La réponse contient une `assetHierarchies` liste dont la structure est la suivante :

```
{
  ...
  "assetHierarchies": [
    {
      "id": "String",
      "name": "String"
    }
  ],
  ...
}
```

Pour plus d'informations, consultez l'[DescribeAsset](#) opération.

2. Pour chaque hiérarchie, exécutez la commande suivante pour répertorier les enfants de la ressource qui sont associés à cette hiérarchie. Remplacez *asset-id* par l'ID ou l'ID externe de la ressource et *hierarchy-id* par l'ID ou l'ID externe de la hiérarchie.

```
aws iotsitewise list-associated-assets \
  --asset-id asset-id \
  --hierarchy-id hierarchy-id
```

Pour plus d'informations, consultez l'[ListAssociatedAssets](#) opération.

3. Exécutez la commande suivante pour supprimer chaque ressource associée, puis pour supprimer la ressource. Remplacez *asset-id* par l'ID ou l'ID externe de la ressource.

```
aws iotsitewise delete-asset --asset-id asset-id
```

Supprimer des modèles d'actifs dans AWS IoT SiteWise

Vous pouvez utiliser la AWS IoT SiteWise console ou l'API pour supprimer un modèle de ressource.

Avant de pouvoir supprimer un modèle d'actif, vous devez d'abord supprimer tous les actifs créés à partir du modèle d'actif.

Lorsque vous supprimez un modèle de ressource, son état est DELETING jusqu'à ce que les modifications soient propagées. Pour de plus amples informations, veuillez consulter [État des ressources et des modèles](#). Une fois le modèle de ressource supprimé, vous ne pouvez plus l'interroger. Si vous le faites, l'API renvoie une réponse HTTP 404.

Rubriques

- [Supprimer un modèle de ressource \(console\)](#)
- [Supprimer un modèle de ressource \(AWS CLI\)](#)

Supprimer un modèle de ressource (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour supprimer un modèle de ressource.

Pour supprimer un modèle de ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle de ressource à supprimer.
4. Si le modèle comporte des ressources, supprimez chaque ressource. Choisissez le nom d'une ressource pour accéder à sa page, où vous pouvez la supprimer. Pour de plus amples informations, veuillez consulter [Supprimer un actif \(console\)](#).
5. Sur la page du modèle, choisissez Supprimer.
6. Dans la boîte de dialogue Supprimer le modèle, procédez comme suit :
 - a. Saisissez **Delete** pour confirmer la suppression.
 - b. Sélectionnez Delete (Supprimer).

Supprimer un modèle de ressource (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour supprimer un modèle de ressource.

Utilisez cette [DeleteAssetModel](#) opération pour supprimer un modèle de ressource. Spécifiez le paramètre suivant :

- `assetModelId`— L'ID de l'actif. Il s'agit de l'identifiant réel au format UUID, ou du `externalId:myExternalId` s'il en possède un. Pour plus d'informations, consultez [Objets de référence avec interface externe IDs](#) dans le Guide de l'utilisateur AWS IoT SiteWise .

Pour supprimer un modèle de ressource (AWS CLI)

1. Exécutez la commande suivante pour répertorier toutes les ressources créées à partir du modèle. Remplacez *asset-model-id* par l'ID ou l'ID externe du modèle d'actif.

```
aws iotsitewise list-assets --asset-model-id asset-model-id
```

Pour plus d'informations, consultez l'[ListAssets](#) opération.

2. Si la commande précédente renvoie des ressources du modèle, supprimez chaque ressource. Pour de plus amples informations, veuillez consulter [Supprimer un actif \(AWS CLI\)](#).
3. Exécutez la commande suivante pour supprimer le modèle de ressource. Remplacez *asset-model-id* par l'ID ou l'ID externe du modèle d'actif.

```
aws iotsitewise delete-asset-model --asset-model-id asset-model-id
```

Important

Pour éviter de supprimer un modèle de ressource qui a été mis à jour simultanément depuis la dernière opération de lecture, vous devez définir une demande de suppression conditionnelle. Consultez [Verrouillage optimiste pour les écrits sur le modèle d'actifs](#).

Opérations groupées avec actifs et modèles

Pour travailler avec un grand nombre de ressources ou de modèles d'actifs, utilisez des opérations groupées pour importer et exporter des ressources en masse vers un autre emplacement. Par exemple, vous pouvez créer un fichier de données qui définit les actifs ou les modèles d'actifs dans un compartiment Amazon S3, et utiliser l'importation en masse pour les créer ou les mettre à jour AWS IoT SiteWise. Sinon, si vous avez un grand nombre de ressources ou de modèles d'actifs AWS IoT SiteWise, vous pouvez les exporter vers Amazon S3.

Note

Vous effectuez des opérations AWS IoT SiteWise groupées en appelant des opérations dans l' AWS IoT TwinMaker API. Vous pouvez le faire sans configurer AWS IoT TwinMaker ni créer d' AWS IoT TwinMaker espace de travail. Tout ce dont vous avez besoin, c'est d'un compartiment Amazon S3 dans lequel vous pouvez placer votre AWS IoT SiteWise contenu.

Rubriques

- [Concepts clés et terminologie](#)
- [Fonctionnalités prises en charge](#)
- [Prérequis pour les opérations en masse](#)
- [Exécuter une tâche d'importation groupée](#)
- [Exécuter une tâche d'exportation groupée](#)
- [Suivi de l'avancement des tâches et gestion des erreurs](#)
- [Exemples d'importation de métadonnées](#)
- [Exemples de métadonnées d'exportation](#)
- [AWS IoT SiteWise schéma de tâche de transfert de métadonnées](#)

Concepts clés et terminologie

AWS IoT SiteWise les fonctionnalités d'importation et d'exportation en bloc reposent sur les concepts et la terminologie suivants :

- Importer : action qui consiste à déplacer des actifs ou des modèles d'actifs d'un fichier d'un compartiment Amazon S3 vers AWS IoT SiteWise.

- **Exporter** : action qui consiste à déplacer des actifs ou des modèles AWS IoT SiteWise d'actifs depuis un compartiment Amazon S3.
- **Source** : point de départ à partir duquel vous souhaitez déplacer le contenu.

Par exemple, un compartiment Amazon S3 est une source d'importation et AWS IoT SiteWise une source d'exportation.

- **Destination** : emplacement souhaité vers lequel vous souhaitez déplacer votre contenu.

Par exemple, un compartiment Amazon S3 est une destination d'exportation et AWS IoT SiteWise une destination d'importation.

- **AWS IoT SiteWise Schéma** : Ce schéma est utilisé pour importer et exporter des métadonnées depuis AWS IoT SiteWise.
- **Ressource de haut niveau** : AWS IoT SiteWise ressource que vous pouvez créer ou mettre à jour individuellement, telle qu'un actif ou un modèle d'actif.
- **Sous-ressource** : ressource imbriquée dans une AWS IoT SiteWise ressource de niveau supérieur. Les exemples incluent les propriétés, les hiérarchies et les modèles composites.
- **Métadonnées** : informations clés requises pour importer ou exporter des ressources avec succès. Les définitions des actifs et des modèles d'actifs sont des exemples de métadonnées.
- **metadataTransferJob**: objet créé lors de l'exécution `CreateMetadataTransferJob`.

Fonctionnalités prises en charge

Cette rubrique explique ce que vous pouvez faire lorsque vous exécutez une opération en masse. Les opérations groupées prennent en charge les fonctionnalités suivantes :

- **Création de ressources de haut niveau** : lorsque vous importez un actif ou un modèle d'actif qui ne définit pas d'identifiant, ou dont l'identifiant ne correspond pas à celui d'un actif existant, il est créé en tant que nouvelle ressource.
- **Remplacement des ressources de haut niveau** : lorsque vous importez un actif ou un modèle d'actif dont l'identifiant correspond à un actif qui existe déjà, il remplace la ressource existante.
- **Création, remplacement ou suppression de sous-ressources** : lorsque votre importation remplace une ressource de haut niveau telle qu'un actif ou un modèle d'actif, la nouvelle définition remplace toutes les sous-ressources, telles que les propriétés, les hiérarchies ou les modèles composites.

Par exemple, si vous mettez à jour un modèle de ressource lors d'une importation en bloc et que la version mise à jour définit une propriété qui n'était pas présente dans l'original, une nouvelle

propriété est créée. S'il définit une propriété qui existe déjà, la propriété existante sera mise à jour. Si le modèle d'actif mis à jour omet une propriété présente dans l'original, la propriété est supprimée.

- Aucune suppression de ressources de niveau supérieur : les opérations groupées ne suppriment pas un actif ou un modèle d'actif. Les opérations groupées ne font que les créer ou les mettre à jour.

Prérequis pour les opérations en masse

Cette section explique les conditions requises pour les opérations en masse, notamment les autorisations AWS Identity and Access Management (IAM) pour l'échange de ressources entre les AWS services et votre machine locale. Avant de démarrer une opération en bloc, remplissez les conditions préalables suivantes :

- 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

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

Pour effectuer des opérations groupées, vous avez besoin des règles suivantes.

AWS IoT SiteWise politique

Cette politique permet d'accéder aux actions d' AWS IoT SiteWise API requises pour les opérations groupées :

```
{
  "Sid": "SiteWiseApiAccess",
  "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",
    "iotsitewise>CreateAssetModelCompositeModel",
    "iotsitewise:UpdateAssetModelCompositeModel",
    "iotsitewise:DescribeAssetModelCompositeModel",
    "iotsitewise>DeleteAssetModelCompositeModel",
    "iotsitewise:ListAssetModelCompositeModels",
    "iotsitewise:ListCompositionRelationships",
    "iotsitewise:DescribeAssetCompositeModel"
  ],
  "Resource": "*"
}
```

AWS IoT TwinMaker politique

Cette politique permet d'accéder aux opérations AWS IoT TwinMaker d'API que vous utilisez pour travailler avec des opérations groupées :

```

{
  "Sid": "MetadataTransferJobApiAccess",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:CreateMetadataTransferJob",
    "iottwinmaker:CancelMetadataTransferJob",
    "iottwinmaker:GetMetadataTransferJob",
    "iottwinmaker:ListMetadataTransferJobs"
  ],
}
```

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

Politique Amazon S3

Cette politique donne accès aux compartiments Amazon S3 pour le transfert de métadonnées pour les opérations en masse.

For a specific Amazon S3 bucket

Si vous utilisez un bucket spécifique pour travailler avec les métadonnées de vos opérations groupées, cette politique permet d'accéder à ce bucket :

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

To allow any Amazon S3 bucket

Si vous devez utiliser de nombreux compartiments différents pour travailler avec les métadonnées de vos opérations groupées, cette politique permet d'accéder à n'importe quel compartiment :

```
{
  "Effect": "Allow",
  "Action": [
    "s3:PutObject",
    "s3:GetObject",
    "s3:GetBucketLocation",
    "s3:ListBucket",

```



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

Pour plus d'informations sur la résolution des problèmes liés aux opérations d'importation et d'exportation, consultez [Résoudre les problèmes d'importation et d'exportation en masse](#).

Exécuter une tâche d'importation groupée

L'importation en masse consiste à déplacer des métadonnées dans un AWS IoT SiteWise espace de travail. Par exemple, l'importation en masse peut déplacer des métadonnées d'un fichier local ou d'un fichier d'un compartiment Amazon S3 vers un AWS IoT SiteWise espace de travail.

Étape 1 : Préparation du fichier à importer

Téléchargez le fichier au format AWS IoT SiteWise natif pour importer les actifs et les modèles d'actifs. Pour plus d'informations, consultez [AWS IoT SiteWise schéma de tâche de transfert de métadonnées](#).

Étape 2 : Chargez le fichier préparé sur Amazon S3

Téléchargez le fichier sur Amazon S3. Consultez la section [Chargement d'un fichier vers Amazon S3](#) dans le guide de l'utilisateur d'Amazon Simple Storage Service pour plus de détails.

Importer des métadonnées (console)

Vous pouvez utiliser le Console AWS IoT SiteWise pour importer des métadonnées en bloc. Suivez [Étape 1 : Préparation du fichier à importer](#) et [Étape 2 : Chargez le fichier préparé sur Amazon S3](#) pour préparer un fichier prêt à être importé.

Importez des données depuis Amazon S3 vers Console AWS IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).
2. Choisissez Bulk operations New dans le volet de navigation.
3. Choisissez Nouvelle importation pour démarrer le processus d'importation.

4. Sur la page Importer des métadonnées :

- Choisissez Parcourir Amazon S3 pour afficher le compartiment et les fichiers Amazon S3.
- Accédez au compartiment Amazon S3 qui contient le fichier d'importation préparé.
- Sélectionnez le fichier à importer.
- Passez en revue le fichier sélectionné, puis choisissez Importer.

5. La page Opérations groupées sur les SiteWise métadonnées Console AWS IoT SiteWise affiche la tâche d'importation nouvellement créée dans le tableau de progression des tâches.

Importer des métadonnées (AWS CLI)

Pour effectuer une opération d'importation, procédez comme suit :

Importez des données depuis Amazon S3 vers AWS CLI

1. Créez un fichier de métadonnées qui indique les ressources que vous souhaitez importer, en suivant le [AWS IoT SiteWise schéma de tâche de transfert de métadonnées](#). Stockez ce fichier dans votre compartiment Amazon S3.

Pour des exemples de fichiers de métadonnées à importer, consultez [Exemples d'importation de métadonnées](#).

2. Créez maintenant un fichier JSON avec le corps de la requête. Le corps de la demande indique la source et la destination de la tâche de transfert. Ce fichier est distinct du fichier de l'étape précédente. Assurez-vous de spécifier votre compartiment Amazon S3 comme source et `iotsitewise` comme destination.

L'exemple suivant montre le corps de la demande :

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3::amzn-s3-demo-bucket/"
    }
  }],
  "destination": {
    "type": "iotsitewise"
  }
}
```

```
}  
}
```

3. Appelez le `CreateMetadataTransferJob` en exécutant la AWS CLI commande suivante. Dans cet exemple, le corps de la demande de l'étape précédente est nommé `createMetadataTransferJobExport.json`.

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

Cela créera une tâche de transfert de métadonnées et lancera le processus de transfert des ressources que vous avez sélectionnées.

Exécuter une tâche d'exportation groupée

L'exportation en masse consiste à déplacer des métadonnées d'un AWS IoT SiteWise espace de travail vers un compartiment Amazon S3.

Lorsque vous exportez votre AWS IoT SiteWise contenu en masse vers Amazon S3, vous pouvez définir des filtres pour limiter les modèles d'actifs et les actifs spécifiques que vous souhaitez exporter.

Les filtres doivent être spécifiés dans une `iotSiteWiseConfiguration` section de la section sources de votre requête JSON.

Note

Vous pouvez inclure plusieurs filtres dans votre demande. L'opération groupée exportera les modèles d'actifs et les actifs correspondant à l'un des filtres.

Si vous ne fournissez aucun filtre, l'opération groupée exporte tous vos modèles et actifs d'actifs.

Exemple corps de la demande avec filtres

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

```

    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID"
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeAssets": true
          }
        },
        {
          "filterByAssetModel": {
            "assetModelId": "asset model ID",
            "includeOffspring": true
          }
        }
      ]
    },
    "destination": {
      "type": "s3",
      "s3Configuration": {
        "location": "arn:aws:s3:::amzn-s3-demo-bucket"
      }
    }
  }
}

```

Exporter les métadonnées (console)

La procédure suivante explique l'action d'exportation de la console :

Créez une tâche d'exportation dans Console AWS IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).
2. Choisissez Bulk operations New dans le volet de navigation.
3. Choisissez Nouvelle exportation pour démarrer le processus d'exportation.
4. Sur la page Exporter les métadonnées :

- Entrez le nom de la tâche d'exportation. Il s'agit du nom utilisé pour le fichier exporté dans votre compartiment Amazon S3.
 - Choisissez les ressources à exporter, ce qui définit les filtres pour le travail :
 - Exportez tous les actifs et modèles d'actifs. Utilisez des filtres sur les actifs et les modèles d'actifs.
 - Exportez des actifs. Filtrez en fonction de vos actifs.
 - Sélectionnez la ressource à utiliser pour le filtre d'exportation.
 - (Facultatif) Ajoutez la descendance ou le modèle d'actif associé.
 - Exportez des modèles d'actifs. Filtrez en fonction de vos modèles d'actifs.
 - Sélectionnez le modèle de ressource à utiliser pour le filtre d'exportation.
 - (Facultatif) Ajoutez la descendance, l'actif associé ou les deux.
 - Choisissez Suivant.
 - Accédez au compartiment Amazon S3 :
 - Choisissez Parcourir Amazon S3 pour afficher le compartiment et les fichiers Amazon S3.
 - Accédez au compartiment Amazon S3 dans lequel le fichier doit être placé.
 - Choisissez Suivant.
 - Passez en revue la tâche d'exportation et choisissez Exporter.
5. La page Opérations groupées sur les SiteWise métadonnées Console AWS IoT SiteWise affiche la tâche d'importation nouvellement créée dans le tableau de progression des tâches.

Pour connaître les différentes manières d'utiliser les filtres lors de l'exportation de métadonnées, consultez [Exemples de métadonnées d'exportation](#).

Exporter les métadonnées (AWS CLI)

La procédure suivante explique l'opération AWS CLI d'exportation :

Exporter des données depuis AWS IoT SiteWise Amazon S3

1. Créez un fichier JSON avec le corps de votre requête. Le corps de la demande indique la source et la destination de la tâche de transfert. L'exemple suivant montre un exemple de corps de demande :

```
{
  "metadataTransferJobId": "your-transfer-job-Id",
  "sources": [{
    "type": "iotsitewise"
  }],
  "destination": {
    "type": "s3",
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Assurez-vous de spécifier votre compartiment Amazon S3 comme destination de la tâche de transfert de métadonnées.

 Note

Cet exemple exportera tous vos modèles d'actifs et actifs. Pour limiter l'exportation à des modèles d'actifs ou à des actifs spécifiques, vous pouvez inclure des filtres dans le corps de votre demande. Pour plus d'informations sur l'application de filtres d'exportation, consultez [Exemples de métadonnées d'exportation](#).

2. Enregistrez le corps du fichier de votre demande pour l'utiliser à l'étape suivante. Dans cet exemple, le fichier est nommé `createMetadataTransferJobExport.json`.
3. Appelez le `CreateMetadataTransferJob` en exécutant la AWS CLI commande suivante :

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

Remplacez le fichier JSON d'entrée `createMetadataTransferJobExport.json` par votre propre nom de fichier de transfert.

Suivi de l'avancement des tâches et gestion des erreurs

Le traitement d'une tâche en masse prend du temps. Chaque tâche est traitée dans l'ordre de AWS IoT SiteWise réception de la demande. Il est traité one-at-a-time pour chaque compte. Lorsqu'une tâche est terminée, la suivante en file d'attente démarre automatiquement le traitement. AWS IoT

SiteWise résout les tâches de manière asynchrone et met à jour le statut de chacune au fur et à mesure de son avancement. Chaque tâche possède un champ d'état qui contient l'état de la ressource et un message d'erreur, le cas échéant.

L'état peut avoir l'une des valeurs suivantes :

- **VALIDATING**— Validation de la tâche, y compris le format de fichier soumis et son contenu.
- **PENDING**— La tâche est dans une file d'attente. Vous pouvez annuler des tâches dans cet état depuis la AWS IoT SiteWise console, mais tous les autres états continueront jusqu'à la fin.
- **RUNNING**— Traitement de la tâche. Il s'agit de créer et de mettre à jour des ressources telles que définies par le fichier d'importation, ou d'exporter des ressources en fonction des filtres de tâches d'exportation choisis. En cas d'annulation, aucune ressource importée par cette tâche n'est supprimée. Pour plus d'informations, consultez [Consulter la progression et les détails de la tâche \(console\)](#).
- **CANCELLING**— L'offre d'emploi est en cours d'annulation.
- **ERROR**— Une ou plusieurs ressources n'ont pas pu être traitées. Consultez le rapport de travail détaillé pour plus d'informations. Pour plus d'informations, consultez [Inspecter les détails de l'erreur \(console\)](#).
- **COMPLETED**— Job terminé sans erreur.
- **CANCELLED**— La tâche est annulée et n'est pas mise en file d'attente. Si vous avez annulé une **RUNNING** tâche, les ressources déjà importées par cette tâche au moment de l'annulation ne sont pas supprimées de AWS IoT SiteWise.

Rubriques

- [Suivi de l'avancement des tâches](#)
- [Inspectez les erreurs pour AWS IoT SiteWise](#)

Suivi de l'avancement des tâches

Consulter la progression et les détails de la tâche (console)

Consultez [Importer des métadonnées \(console\)](#) ou [Exporter les métadonnées \(console\)](#) pour démarrer une tâche groupée.

Vue d'ensemble de l'avancement des tâches dans la AWS IoT SiteWise console :

1. Accédez à la [console AWS IoT SiteWise](#).
2. Choisissez Bulk operations New dans le volet de navigation.
3. Le tableau de progression des tâches de la AWS IoT SiteWise console affiche la liste des tâches exécutées en bloc.
4. La colonne Type de tâche indique s'il s'agit d'une tâche d'exportation ou d'importation. Les colonnes Date d'importation indiquent la date de début de la tâche.
5. La colonne Status affiche le statut de la tâche. Vous pouvez sélectionner une tâche pour en voir les détails.
6. La tâche sélectionnée affiche Success en cas de réussite, ou une liste d'échecs en cas d'échec de la tâche. Une description de l'erreur est également affichée pour chaque type de ressource.

Vue d'ensemble des détails du job dans la AWS IoT SiteWise console :

Le tableau de progression des tâches de la AWS IoT SiteWise console affiche la liste des tâches exécutées en bloc.

1. Choisissez un poste pour obtenir plus de détails.
2. Pour une tâche d'importation, le Data source ARN représente l'emplacement du fichier d'importation sur Amazon S3.
3. Pour une tâche d'exportation, le Data destination ARN représente l'emplacement du fichier sur Amazon S3 après l'exportation.
4. Les Status et Status reason, fournissent des détails supplémentaires sur le travail en cours. Pour plus d'informations, consultez [Suivi de l'avancement des tâches et gestion des erreurs](#).
5. Le Queued position représente la position de la tâche dans la file d'attente des processus. Les tâches sont traitées une par une. Une position en file d'attente de 1 indique que le travail sera traité ensuite.
6. La page de détails des tâches affiche également le nombre d'avancement des tâches.
 - Les types de comptage de l'avancement des tâches sont les suivants :
 - i. Total resources— Indique le nombre total d'actifs dans le processus de transfert.
 - ii. Succeeded— Indique le nombre d'actifs transférés avec succès au cours du processus.

- iii. **Failed**— Indique le nombre d'actifs défaillants au cours du processus.
 - iv. **Skipped**— Indique le nombre d'actifs ignorés au cours du processus.
7. Un statut de tâche **PENDING** égal ou **VALIDATING** affiche la progression de toutes les tâches—. Cela indique que le nombre de progrès des tâches est en cours d'évaluation.
 8. Un statut de tâche de **RUNNING** affiche le `Total resources` nombre de tâches soumises pour traitement. Les chiffres détaillés (`SucceededFailed`, et `Skipped`) s'appliquent aux ressources traitées. La somme des dénombrements détaillés est inférieure au `Total resources` nombre, jusqu'à ce que le statut du poste soit **COMPLETED** ou **ERROR**.
 9. Si le statut d'une tâche est **COMPLETED** ou **ERROR**, le `Total resources` nombre est égal à la somme des dénombrements détaillés (`SucceededFailed`, et `Skipped`).
 10. Si le statut d'une tâche est défini comme **ERROR** tel, consultez le tableau des échecs des tâches pour plus de détails sur les erreurs et les échecs spécifiques. Pour plus d'informations, consultez [Inspecter les détails de l'erreur \(console\)](#).

Examiner l'avancement et les détails du travail (AWS CLI)

Après avoir démarré une opération groupée, vous pouvez vérifier ou mettre à jour son statut à l'aide des actions d'API suivantes :

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

Récupérez des informations à l'aide de **GetMetadataTransferJob** l'API :

1. Créez et exécutez une tâche de transfert. Appelez l'API `GetMetadataTransferJob`.

Exemple AWS CLI commande :

```
aws iottwinmaker get-metadata-transfer-job \  
  --metadata-transfer-job-id your_metadata_transfer_job_id \  
  --region your_region
```

2. L'`GetMetadataTransferJobAPI` renvoie un `MetadataTransferJobProgress` objet avec les paramètres suivants :
 - **SucceededCount** — Indique le nombre d'actifs transférés avec succès au cours du processus.

- **FailedCount** — Indique le nombre d'actifs défaillants au cours du processus.
- **SkippedCount** — Indique le nombre de ressources ignorées au cours du processus.
- **TotalCount** — Indique le nombre total d'actifs dans le processus de transfert.

Ces paramètres indiquent l'état d'avancement de la tâche. Si le statut est le cas **RUNNING**, ils permettent de suivre le nombre de ressources restant à traiter.

Si vous rencontrez des erreurs de validation du schéma, ou si **FailedCount** est supérieur ou égal à 1, l'état d'avancement de la tâche passe à **ERROR**. Un rapport d'erreur complet relatif à la tâche est placé dans votre compartiment Amazon S3. Pour plus d'informations, consultez [Inspectez les erreurs pour AWS IoT SiteWise](#).

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

Utilisez un fichier JSON pour filtrer les tâches renvoyées en fonction de leur état actuel. Consultez la procédure suivante :

1. Pour spécifier les filtres que vous souhaitez utiliser, créez un fichier JSON AWS CLI d'entrée. Vous souhaitez utiliser :

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

Pour obtenir la liste des `state` valeurs valides, consultez le Guide [ListMetadataTransferJobsFilter](#) de référence de l'AWS IoT TwinMaker API.

2. Utilisez le fichier JSON comme argument dans l' AWS CLI exemple de commande suivant :

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

- Pour annuler une tâche, utilisez l'action [CancelMetadataTransferJob](#) API. Cette API annule la tâche de transfert de métadonnées spécifique, sans affecter les ressources déjà exportées ou importées :

```
aws iottwinmaker cancel-metadata-transfer-job \  
  --region your_region \  
  --metadata-transfer-job-id job-to-cancel-id
```

Inspectez les erreurs pour AWS IoT SiteWise

Inspecter les détails de l'erreur (console)

Détails de l'erreur dans la AWS IoT SiteWise console :

1. Accédez à la [console AWS IoT SiteWise](#).
2. Consultez le tableau de progression des tâches Console AWS IoT SiteWise pour obtenir la liste des tâches effectuées en bloc.
3. Sélectionnez une tâche pour afficher les détails de la tâche.
4. Si le statut d'une tâche est COMPLETED ou ERROR, le Total resources nombre est égal à la somme des dénombrements détaillés (SucceededFailed, etSkipped).
5. Si le statut d'une tâche est défini comme ERROR tel, consultez le tableau des échecs des tâches pour plus de détails sur les erreurs et les échecs spécifiques.
6. Le tableau des échecs de tâches affiche le contenu du rapport de tâche. Le Resource type champ indique l'emplacement de l'erreur ou des défaillances, par exemple :
 - Par exemple, une erreur de validation Bulk operations template dans le Resource type champ indique que le modèle d'importation et le format de fichier du schéma de métadonnées ne correspondent pas. Pour plus d'informations, consultez [AWS IoT SiteWise schéma de tâche de transfert de métadonnées](#).
 - Un échec Asset dans le Resource type champ indique que l'actif n'a pas été créé en raison d'un conflit avec un autre actif. Consultez la section [Erreurs courantes](#) pour plus d'informations sur les erreurs et les conflits liés aux AWS IoT SiteWise ressources.

Inspecter les détails de l'erreur (AWS CLI)

Pour gérer et diagnostiquer les erreurs produites lors d'une tâche de transfert, consultez la procédure suivante concernant l'utilisation de l'action GetMetadataTransferJob API :

1. Après avoir créé et exécuté une tâche de transfert, appelez [GetMetadataTransferJob](#):

```
aws iottwinmaker get-metadata-transfer-job \  
  --region your_region \  
  --metadata-transfer-job-id job-to-cancel-id
```

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

2. Une fois que l'état de la tâche est passé à `zéroCOMPLETED`, vous pouvez commencer à vérifier les résultats de la tâche.
3. Lorsque vous appelez `GetMetadataTransferJob`, il renvoie un objet appelé [MetadataTransferJobProgress](#).

L' `MetadataTransferJobProgress` objet contient les paramètres suivants :

- `FailedCount` : indique le nombre d'actifs défaillants pendant le processus de transfert.
 - `SkippedCount` : indique le nombre d'actifs ignorés pendant le processus de transfert.
 - `SucceededCount` : indique le nombre d'actifs qui ont réussi pendant le processus de transfert.
 - `TotalCount` : indique le nombre total d'actifs impliqués dans le processus de transfert.
4. En outre, l'appel d'API renvoie un élément `reportUrl` contenant une URL présignée. Si votre tâche de transfert présente des problèmes que vous devez approfondir, consultez cette URL.

Exemples d'importation de métadonnées

Cette section explique comment créer des fichiers de métadonnées pour importer des modèles d'actifs et des actifs en une seule opération d'importation en bloc.

Exemple d'importation en masse

Vous pouvez importer de nombreux modèles et actifs en une seule opération d'importation en bloc. L'exemple suivant montre comment créer un fichier de métadonnées à cette fin.

Dans cet exemple de scénario, plusieurs sites de travail contiennent des robots industriels dans des cellules de travail.

L'exemple définit deux modèles d'actifs :

- `RobotModel1`: Ce modèle d'actif représente un type particulier de robot que vous avez sur vos sites de travail. Le robot possède une propriété de mesure, `Temperature`.
- `WorkCell`: Ce modèle d'actif représente un ensemble de robots au sein de l'un de vos sites de travail. Le modèle d'actifs définit une hiérarchie pour représenter la relation entre une cellule de travail et des robots. `robotHierarchyOEM1`

L'exemple définit également certains actifs :

- WorkCell11: une cellule de travail au sein de votre site de Boston
- RobotArm123456: un robot au sein de cette cellule de travail
- RobotArm987654: un autre robot au sein de cette cellule de travail

Le fichier de métadonnées JSON suivant définit ces modèles et actifs d'actifs. L'exécution d'une importation groupée avec ces métadonnées crée les modèles d'actifs et les actifs qu'ils contiennent AWS IoT SiteWise, y compris leurs relations hiérarchiques.

Fichier de métadonnées à importer

```
{
  "assetModels": [
    {
      "assetModelExternalId": "Robot.OEM1.3536",
      "assetModelName": "RobotModel1",
      "assetModelProperties": [
        {
          "dataType": "DOUBLE",
          "externalId": "Temperature",
          "name": "Temperature",
          "type": {
            "measurement": {
              "processingConfig": {
                "forwardingConfig": {
                  "state": "ENABLED"
                }
              }
            }
          },
          "unit": "fahrenheit"
        }
      ],
    },
    {
      "assetModelExternalId": "ISA95.WorkCell",
      "assetModelName": "WorkCell",
      "assetModelProperties": [],
      "assetModelHierarchies": [
        {
          "externalId": "workCellHierarchyWith0EM1Robot",
```

```

        "name": "robotHierarchyOEM1",
        "childAssetModelExternalId": "Robot.OEM1.3536"
      }
    ]
  },
  "assets": [
    {
      "assetExternalId": "Robot.OEM1.3536.123456",
      "assetName": "RobotArm123456",
      "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
      "assetExternalId": "Robot.OEM1.3536.987654",
      "assetName": "RobotArm987654",
      "assetModelExternalId": "Robot.OEM1.3536"
    },
    {
      "assetExternalId": "BostonSite.Area1.Line1.WorkCell1",
      "assetName": "WorkCell1",
      "assetModelExternalId": "ISA95.WorkCell",
      "assetHierarchies": [
        {
          "externalId": "workCellHierarchyWithOEM1Robot",
          "childAssetExternalId": "Robot.OEM1.3536.123456"
        },
        {
          "externalId": "workCellHierarchyWithOEM1Robot",
          "childAssetExternalId": "Robot.OEM1.3536.987654"
        }
      ]
    }
  ]
}

```

Exemple d'intégration initiale de modèles et d'actifs

Dans cet exemple de scénario, vous avez plusieurs sites de travail qui contiennent des robots industriels dans une entreprise.

L'exemple définit plusieurs modèles d'actifs :

- **Sample_Enterprise**— Ce modèle d'actifs représente l'entreprise dont les sites font partie. Le modèle d'actifs définit une hiérarchie pour représenter la relation entre les sites et l'entreprise. **Enterprise to Site**
- **Sample_Site**— Ce modèle d'actifs représente les sites de fabrication de l'entreprise. Le modèle d'actifs définit une hiérarchie pour représenter la relation entre les lignes et le site. **Site to Line**
- **Sample_Welding Line**— Ce modèle d'actifs représente une chaîne de montage sur les sites de travail. Le modèle d'actifs définit une hiérarchie pour représenter la relation entre les robots et la ligne. **Line to Robot**
- **Sample_Welding Robot**— Ce modèle d'actifs représente un type particulier de robot sur vos sites de travail.

L'exemple définit également les actifs en fonction des modèles d'actifs.

- **Sample_AnyCompany Motor**— Cet actif est créé à partir du modèle **Sample_Enterprise** d'actif.
- **Sample_Chicago**— Cet actif est créé à partir du modèle **Sample_Site** d'actif.
- **Sample_Welding Line 1**— Cet actif est créé à partir du modèle **Sample_Welding Line** d'actif.
- **Sample_Welding Robot 1**— Cet actif est créé à partir du modèle **Sample_Welding Robot** d'actif.
- **Sample_Welding Robot 2**— Cet actif est créé à partir du modèle **Sample_Welding Robot** d'actif.

Le fichier de métadonnées JSON suivant définit ces modèles et actifs d'actifs. L'exécution d'une importation groupée avec ces métadonnées crée les modèles d'actifs et les actifs qu'ils contiennent AWS IoT SiteWise, y compris leurs relations hiérarchiques.

Fichier JSON pour intégrer les actifs et les modèles à importer

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
```

```

        "dataType": "STRING",
        "externalId": "External_Id_Welding_Robot_Serial_Number",
        "name": "Serial Number",
        "type": {
            "attribute": {
                "defaultValue": "-"
            }
        },
        "unit": "-"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Cycle_Count",
        "name": "CycleCount",
        "type": {
            "measurement": {}
        },
        "unit": "EA"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Current",
        "name": "Joint 1 Current",
        "type": {
            "measurement": {}
        },
        "unit": "Amps"
    },
    {
        "dataType": "DOUBLE",
        "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
        "name": "Max Joint 1 Current",
        "type": {
            "metric": {
                "expression": "max(joint1current)",
                "variables": [
                    {
                        "name": "joint1current",
                        "value": {
                            "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
                        }
                    }
                ]
            }
        }
    },
    ],

```



```

        "window": {
            "tumbling": {
                "interval": "5m"
            }
        }
    },
    "unit": "Amps"
}
],
{
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetModelName": "Sample_Welding Line",
    "assetModelProperties": [
        {
            "dataType": "DOUBLE",
            "externalId": "External_Id_Welding_Line_Availability",
            "name": "Availability",
            "type": {
                "measurement": {}
            },
            "unit": "%"
        }
    ],
    "assetModelHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "name": "Line to Robot",
            "childAssetModelExternalId": "External_Id_Welding_Robot"
        }
    ]
},
{
    "assetModelExternalId": "External_Id_Site",
    "assetModelName": "Sample_Site",
    "assetModelProperties": [
        {
            "dataType": "STRING",
            "externalId": "External_Id_Site_Street_Address",
            "name": "Street Address",
            "type": {
                "attribute": {
                    "defaultValue": "-"
                }
            }
        }
    ]
}
]

```

```

        }
      },
      "unit": "-"
    }
  ],
  "assetModelHierarchies": [
    {
      "externalId": "External_Id_Site_T0_Line",
      "name": "Site to Line",
      "childAssetModelExternalId": "External_Id_Welding_Line"
    }
  ]
},
{
  "assetModelExternalId": "External_Id_Enterprise",
  "assetModelName": "Sample_Enterprise",
  "assetModelProperties": [
    {
      "dataType": "STRING",
      "name": "Company Name",
      "externalId": "External_Id_Enterprise_Company_Name",
      "type": {
        "attribute": {
          "defaultValue": "-"
        }
      },
      "unit": "-"
    }
  ],
  "assetModelHierarchies": [
    {
      "externalId": "External_Id_Enterprise_T0_Site",
      "name": "Enterprise to Site",
      "childAssetModelExternalId": "External_Id_Site"
    }
  ]
}
],
"assets": [
  {
    "assetExternalId": "External_Id_Welding_Robot_1",
    "assetName": "Sample_Welding Robot 1",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [

```

```

        {
            "externalId": "External_Id_Welding_Robot_Serial_Number",
            "attributeValue": "S1000"
        },
        {
            "externalId": "External_Id_Welding_Robot_Cycle_Count",
            "alias": "AnyCompany/Chicago/Welding Line/S1000/Count"
        },
        {
            "externalId": "External_Id_Welding_Robot_Joint_1_Current",
            "alias": "AnyCompany/Chicago/Welding Line/S1000/1/Current"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Robot_2",
    "assetName": "Sample_Welding Robot 2",
    "assetModelExternalId": "External_Id_Welding_Robot",
    "assetProperties": [
        {
            "externalId": "External_Id_Welding_Robot_Serial_Number",
            "attributeValue": "S2000"
        },
        {
            "externalId": "External_Id_Welding_Robot_Cycle_Count",
            "alias": "AnyCompany/Chicago/Welding Line/S2000/Count"
        },
        {
            "externalId": "External_Id_Welding_Robot_Joint_1_Current",
            "alias": "AnyCompany/Chicago/Welding Line/S2000/1/Current"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetProperties": [
        {
            "externalId": "External_Id_Welding_Line_Availability",
            "alias": "AnyCompany/Chicago/Welding Line/Availability"
        }
    ],
    "assetHierarchies": [

```

```

        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_1"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        }
    ]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",
            "childAssetExternalId": "External_Id_Welding_Line_1"
        }
    ]
},
{
    "assetExternalId": "External_Id_Enterprise_AnyCompany",
    "assetName": "Sample_AnyEnterprise Motor",
    "assetModelExternalId": "External_Id_Enterprise",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Enterprise_T0_Site",
            "childAssetExternalId": "External_Id_Site_Chicago"
        }
    ]
}
]
}

```

La capture d'écran suivante montre les modèles qui s'affichent Console AWS IoT SiteWise après l'exécution de l'exemple de code précédent.

IoT SiteWise > Models

Models (4) Refresh Create component model Create asset model

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter instances

Name	Status	Model type	Date created	Date modified
Sample_Enterprise	ACTIVE	Asset model	November 10, 2023 at 11:22:13 (UT...)	November 10, 202...
Sample_Site	ACTIVE	Asset model	November 10, 2023 at 11:21:57 (UT...)	November 10, 202...
Sample_Welding Line	ACTIVE	Asset model	November 10, 2023 at 11:21:40 (UT...)	November 10, 202...
Sample_Welding Robot	ACTIVE	Asset model	November 10, 2023 at 11:21:24 (UT...)	November 10, 202...

La capture d'écran suivante montre les modèles, les actifs et les hiérarchies qui s'affichent Console AWS IoT SiteWise après l'exécution de l'exemple de code précédent.

IoT SiteWise > Assets

Assets (1) Refresh Create asset

Assets represent industrial devices and processes that send data streams to SiteWise. Models are structures that enforce a specific model of properties and hierarchies for all instances of each asset. You must create every asset from a model.

Filter top level assets

Name	Description	Status	Date created	Date modified
☒ Sample_AnyEnterprise Motor		ACTIVE	November 10, 2023 at 11:23:06 (UTC-5:00)	November 10, 2023 at 11:23:06 (UTC-...
☒ Sample_Chicago		ACTIVE	November 10, 2023 at 11:22:57 (UTC-5:00)	November 10, 2023 at 11:22:57 (UTC-...
☒ Sample_Welding Line 1		ACTIVE	November 10, 2023 at 11:22:48 (UTC-5:00)	November 10, 2023 at 11:22:48 (UTC-...
☐ Sample_Welding Robot 1		ACTIVE	November 10, 2023 at 11:22:39 (UTC-5:00)	November 10, 2023 at 11:22:39 (UTC-...
☐ Sample_Welding Robot 2		ACTIVE	November 10, 2023 at 11:22:30 (UTC-5:00)	November 10, 2023 at 11:22:30 (UTC-...

Exemple d'intégration d'actifs supplémentaires

Cet exemple définit les actifs supplémentaires à importer dans un modèle d'actif existant dans votre compte :

- **Sample_Welding Line 2**— Cet actif est créé à partir du modèle **Sample_Welding Line** d'actif.
- **Sample_Welding Robot 3**— Cet actif est créé à partir du modèle **Sample_Welding Robot** d'actif.
- **Sample_Welding Robot 4**— Cet actif est créé à partir du modèle **Sample_Welding Robot** d'actif.

Pour créer les actifs initiaux pour cet exemple, reportez-vous à [Exemple d'intégration initiale de modèles et d'actifs](#).

Le fichier de métadonnées JSON suivant définit ces modèles et actifs d'actifs. L'exécution d'une importation groupée avec ces métadonnées crée les modèles d'actifs et les actifs qu'ils contiennent AWS IoT SiteWise, y compris leurs relations hiérarchiques.

Fichier JSON pour intégrer des ressources supplémentaires

```
{
  "assets": [
    {
      "assetExternalId": "External_Id_Welding_Robot_3",
      "assetName": "Sample_Welding Robot 3",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S3000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/Count"
        },
        {
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "alias": "AnyCompany/Chicago/Welding Line/S3000/1/Current"
        }
      ]
    },
    {
      "assetExternalId": "External_Id_Welding_Robot_4",
      "assetName": "Sample_Welding Robot 4",
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetProperties": [
        {
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "attributeValue": "S4000"
        },
        {
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "alias": "AnyCompany/Chicago/Welding Line/S4000/Count"
        }
      ]
    }
  ]
}
```

```

        },
        {
            "externalId": "External_Id_Welding_Robot_Joint_1_Current",
            "alias": "AnyCompany/Chicago/Welding Line/S4000/1/Current"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_1",
    "assetName": "Sample_Welding Line 1",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_1"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_2"
        },
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_3"
        }
    ]
},
{
    "assetExternalId": "External_Id_Welding_Line_2",
    "assetName": "Sample_Welding Line 2",
    "assetModelExternalId": "External_Id_Welding_Line",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Welding_Line_T0_Robot",
            "childAssetExternalId": "External_Id_Welding_Robot_4"
        }
    ]
},
{
    "assetExternalId": "External_Id_Site_Chicago",
    "assetName": "Sample_Chicago",
    "assetModelExternalId": "External_Id_Site",
    "assetHierarchies": [
        {
            "externalId": "External_Id_Site_T0_Line",

```

```

        "childAssetExternalId": "External_Id_Welding_Line_1"
      },
      {
        "externalId": "External_Id_Site_T0_Line",
        "childAssetExternalId": "External_Id_Welding_Line_2"
      }
    ]
  }
}

```

La capture d'écran suivante montre les modèles, les actifs et les hiérarchies qui s'affichent Console AWS IoT SiteWise après l'exécution de l'exemple de code précédent.

The screenshot shows the AWS IoT SiteWise console with the 'Assets' tab selected. It displays a list of assets under the heading 'Assets (1)'. A search bar is present with the text 'Filter top level assets'. The assets are organized into a tree structure:

- Sample_AnyCompany Motor** (ACTIVE)
 - Sample_Chicago** (ACTIVE)
 - Sample_Welding Line 1** (ACTIVE)
 - Sample_Welding Robot 2 (ACTIVE)
 - Sample_Welding Robot 3 (ACTIVE)
 - Sample_Welding Robot 1 (ACTIVE)
 - Sample_Welding Line 2** (ACTIVE)
 - Sample_Welding Robot 4 (ACTIVE)

The table columns are: Name, Description, Status, Date created, and Date modified. All assets are in an 'ACTIVE' state.

Exemple d'intégration de nouvelles propriétés

Cet exemple définit de nouvelles propriétés sur les modèles d'actifs existants. Veuillez [Exemple d'intégration d'actifs supplémentaires](#) à intégrer des actifs et des modèles supplémentaires.

- **Joint 1 Temperature**— Cette propriété est ajoutée au modèle **Sample_Welding Robot** d'actif. Cette nouvelle propriété se propagera également à chaque actif créé à partir du modèle **Sample_Welding Robot** d'actif.

Pour ajouter une nouvelle propriété à un modèle d'actif existant, consultez l'exemple de fichier de métadonnées JSON suivant. Comme indiqué dans le JSON, la définition complète du modèle

Sample_Welding Robot d'actif existant doit être fournie avec la nouvelle propriété. Si la liste complète des propriétés de la définition existante n'est pas fournie, AWS IoT SiteWise supprime les propriétés omises.

Fichier JSON pour intégrer de nouvelles propriétés

Cet exemple ajoute une nouvelle propriété Joint 1 Temperature au modèle d'actif.

```
{
  "assetModels": [
    {
      "assetModelExternalId": "External_Id_Welding_Robot",
      "assetModelName": "Sample_Welding Robot",
      "assetModelProperties": [
        {
          "dataType": "STRING",
          "externalId": "External_Id_Welding_Robot_Serial_Number",
          "name": "Serial Number",
          "type": {
            "attribute": {
              "defaultValue": "-"
            }
          },
          "unit": "-"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Cycle_Count",
          "name": "CycleCount",
          "type": {
            "measurement": {}
          },
          "unit": "EA"
        },
        {
          "dataType": "DOUBLE",
          "externalId": "External_Id_Welding_Robot_Joint_1_Current",
          "name": "Joint 1 Current",
          "type": {
            "measurement": {}
          },
          "unit": "Amps"
        }
      ]
    }
  ]
}
```

```

    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Max_Current",
      "name": "Max Joint 1 Current",
      "type": {
        "metric": {
          "expression": "max(joint1current)",
          "variables": [
            {
              "name": "joint1current",
              "value": {
                "propertyExternalId":
"External_Id_Welding_Robot_Joint_1_Current"
              }
            }
          ],
          "window": {
            "tumbling": {
              "interval": "5m"
            }
          }
        }
      },
      "unit": "Amps"
    },
    {
      "dataType": "DOUBLE",
      "externalId": "External_Id_Welding_Robot_Joint_1_Temperature",
      "name": "Joint 1 Temperature",
      "type": {
        "measurement": {}
      },
      "unit": "degC"
    }
  ]
}

```

Exemple de gestion de flux de données

Cet exemple montre deux manières de gérer les flux de données associés à une propriété d'actif. Lorsque vous renommez un alias de propriété d'actif, il existe deux options pour les données historiques actuellement stockées dans le flux de données de la propriété d'actif.

- Option 1 — Conserver le flux de données actuel et renommer uniquement l'alias, afin que les données historiques soient accessibles avec le nouvel alias.

Dans l'exemple de fichier de métadonnées JSON, la propriété de l'actif avec ID `External_Id_Welding_Robot_Cycle_Count` change son alias en `AnyCompany/Chicago/Welding Line/S3000/Count-Updated`. Les données historiques de cette propriété d'actif restent les mêmes après cette modification.

- Option 2 — Attribuez un nouveau flux de données à la propriété de l'actif accessible avec le nouvel alias. L'ancien flux de données ainsi que ses données historiques sont toujours accessibles avec l'ancien alias, mais ils ne sont associés à aucune propriété d'actif.

Dans l'exemple de fichier de métadonnées JSON, la propriété de l'actif avec ID `External_Id_Welding_Robot_Joint_1_Current` change son alias en `AnyCompany/Chicago/Welding Line/S4999/1/Current`. Cette fois, la valeur supplémentaire `retainDataOnAliasChange` est présente et définie sur `False`. Avec ce paramètre, le flux de données d'origine est dissocié de la propriété de l'actif et un nouveau flux de données ne contenant aucune donnée historique est créé.

Pour accéder à l'ancien flux de données contenant les données historiques d'origine AWS Console Home, rendez-vous sur la page Flux de données et recherchez l'ancien alias `AnyCompany/Chicago/Welding Line/S3000/1/Current`.

Fichier JSON pour mettre à jour les alias de propriété

```
{
  "assetExternalId": "External_Id_Welding_Robot_3",
  "assetName": "Sample_Welding Robot 3",
  "assetModelExternalId": "External_Id_Welding_Robot",
  "assetProperties": [
    {
      "externalId": "External_Id_Welding_Robot_Serial_Number",
      "attributeValue": "S3000"
    }
  ]
}
```

```

    },
    {
      "externalId": "External_Id_Welding_Robot_Cycle_Count",
      "alias": "AnyCompany/Chicago/Welding Line/S3000/Count-Updated"
    },
    {
      "externalId": "External_Id_Welding_Robot_Joint_1_Current",
      "alias": "AnyCompany/Chicago/Welding Line/S4999/1/Current",
      "retainDataOnAliasChange": "FALSE"
    }
  ]
}

```

Exemples de métadonnées d'exportation

Lorsque vous exportez votre AWS IoT SiteWise contenu en masse vers Amazon S3, vous pouvez définir des filtres pour limiter les modèles d'actifs et les actifs spécifiques que vous souhaitez exporter.

Vous spécifiez les filtres dans une `iotSiteWiseConfiguration` section de la `sources` section du corps de votre demande.

Note

Vous pouvez inclure plusieurs filtres. L'opération groupée exportera tout modèle de ressource ou ressource correspondant à l'un des filtres.

Si vous ne fournissez aucun filtre, l'opération exportera tous vos modèles et actifs d'actifs.

```

{
  "metadataTransferJobId": "your-transfer-job-id",
  "sources": [{
    "type": "iotsitewise",
    "iotSiteWiseConfiguration": {
      "filters": [{
        list of filters
      }]
    }
  ]},
  "destination": {
    "type": "s3",

```

```
    "s3Configuration": {
      "location": "arn:aws:s3:::amzn-s3-demo-bucket"
    }
  }
}
```

Filtrer par modèle d'actif

Vous pouvez filtrer un modèle d'actif spécifique. Vous pouvez également inclure tous les actifs utilisant ce modèle, ou tous les modèles d'actifs au sein de sa hiérarchie. Vous ne pouvez pas inclure à la fois les actifs et la hiérarchie.

Pour plus d'informations sur les hiérarchies, consultez [Définir les hiérarchies des modèles d'actifs](#).

Asset model

Ce filtre inclut le modèle d'actif spécifié :

```
"filterByAssetModel": {
  "assetModelId": "asset model ID"
}
```

Asset model and its assets

Ce filtre inclut le modèle d'actif spécifié, ainsi que tous les actifs utilisant ce modèle d'actif :

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeAssets": true
}
```

Asset model and its hierarchy

Ce filtre inclut le modèle d'actif spécifié, ainsi que tous les modèles d'actifs associés dans sa hiérarchie :

```
"filterByAssetModel": {
  "assetModelId": "asset model ID",
  "includeOffspring": true
}
```

Filtrer par actif

Vous pouvez filtrer un actif spécifique. Vous pouvez également inclure son modèle d'actif ou tous les actifs associés dans sa hiérarchie. Vous ne pouvez pas inclure à la fois le modèle et la hiérarchie des actifs.

Pour plus d'informations sur les hiérarchies, consultez [Définir les hiérarchies des modèles d'actifs](#).

Asset

Ce filtre inclut l'actif spécifié :

```
"filterByAsset": {  
  "assetId": "asset ID"  
}
```

Asset and its asset model

Ce filtre inclut l'actif spécifié, ainsi que le modèle d'actif qu'il utilise :

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeAssetModel": true  
}
```

Asset and its hierarchy

Ce filtre inclut l'actif spécifié, ainsi que tous les actifs associés dans sa hiérarchie :

```
"filterByAsset": {  
  "assetId": "asset ID",  
  "includeOffspring": true  
}
```

AWS IoT SiteWise schéma de tâche de transfert de métadonnées

Utilisez le schéma des tâches de transfert de AWS IoT SiteWise métadonnées à titre de référence lorsque vous effectuez vos propres opérations d'importation et d'exportation en bloc :

```
{  
  "$schema": "https://json-schema.org/draft/2020-12/schema",  
}
```

```

"title": "IoTSiteWise",
"description": "Metadata transfer job resource schema for IoTSiteWise",
"definitions": {
  "Name": {
    "type": "string",
    "minLength": 1,
    "maxLength": 256,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "Description": {
    "type": "string",
    "minLength": 1,
    "maxLength": 2048,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "ID": {
    "type": "string",
    "minLength": 36,
    "maxLength": 36,
    "pattern": "^[0-9a-f]{8}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{4}-[0-9a-f]{12}$"
  },
  "ExternalId": {
    "type": "string",
    "minLength": 2,
    "maxLength": 128,
    "pattern": "[a-zA-Z0-9_][a-zA-Z_\\-0-9.:]*[a-zA-Z0-9_]+"
  },
  "AttributeValue": {
    "description": "The value of the property attribute.",
    "type": "string",
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "PropertyUnit": {
    "description": "The unit of measure (such as Newtons or RPM) of the asset property.",
    "type": "string",
    "minLength": 1,
    "maxLength": 256,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "PropertyAlias": {
    "description": "The property alias that identifies the property.",
    "type": "string",
    "minLength": 1,

```

```

    "maxLength": 1000,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "AssetProperty": {
    "description": "The asset property's definition, alias, unit, and notification
state.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "id"
        ]
      },
      {
        "required": [
          "externalId"
        ]
      }
    ],
    "properties": {
      "id": {
        "description": "The ID of the asset property.",
        "$ref": "#/definitions/ID"
      },
      "externalId": {
        "description": "The ExternalID of the asset property.",
        "$ref": "#/definitions/ExternalId"
      },
      "alias": {
        "$ref": "#/definitions/PropertyAlias"
      },
      "unit": {
        "$ref": "#/definitions/PropertyUnit"
      },
      "attributeValue": {
        "$ref": "#/definitions/AttributeValue"
      },
      "retainDataOnAliasChange": {
        "type": "string",
        "default": "TRUE",
        "enum": [
          "TRUE",
          "FALSE"
        ]
      }
    }
  }
}

```



```

    ]
  },
  "propertyNotificationState": {
    "description": "The MQTT notification state (ENABLED or DISABLED) for this
asset property.",
    "type": "string",
    "enum": [
      "ENABLED",
      "DISABLED"
    ]
  }
},
"AssetHierarchy": {
  "description": "A hierarchy specifies allowed parent/child asset relationships.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetId"
      ]
    },
    {
      "required": [
        "id",
        "childAssetExternalId"
      ]
    },
    {
      "required": [
        "externalId",
        "childAssetExternalId"
      ]
    }
  ],
  "properties": {

```

```
    "id": {
      "description": "The ID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ID"
    },
    "externalId": {
      "description": "The ExternalID of a hierarchy in the parent asset's model.",
      "$ref": "#/definitions/ExternalId"
    },
    "childAssetId": {
      "description": "The ID of the child asset to be associated.",
      "$ref": "#/definitions/ID"
    },
    "childAssetExternalId": {
      "description": "The ExternalID of the child asset to be associated.",
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Tag": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "key",
    "value"
  ],
  "properties": {
    "key": {
      "type": "string"
    },
    "value": {
      "type": "string"
    }
  }
},
"AssetModelType": {
  "type": "string",
  "default": null,
  "enum": [
    "ASSET_MODEL",
    "COMPONENT_MODEL"
  ]
},
"AssetModelCompositeModel": {
```

```
"description": "Contains a composite model definition in an asset model. This composite model definition is applied to all assets created from the asset model.",
"type": "object",
"additionalProperties": false,
"anyOf": [
  {
    "required": [
      "id"
    ]
  },
  {
    "required": [
      "externalId"
    ]
  }
],
"required": [
  "name",
  "type"
],
"properties": {
  "id": {
    "description": "The ID of the asset model composite model.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of the asset model composite model.",
    "$ref": "#/definitions/ExternalId"
  },
  "parentId": {
    "description": "The ID of the parent asset model composite model.",
    "$ref": "#/definitions/ID"
  },
  "parentExternalId": {
    "description": "The ExternalID of the parent asset model composite model.",
    "$ref": "#/definitions/ExternalId"
  },
  "composedAssetModelId": {
    "description": "The ID of the composed asset model.",
    "$ref": "#/definitions/ID"
  },
  "composedAssetModelExternalId": {
    "description": "The ExternalID of the composed asset model.",
    "$ref": "#/definitions/ExternalId"
  }
}
```

```

    },
    "description": {
      "description": "A description for the asset composite model.",
      "$ref": "#/definitions/Description"
    },
    "name": {
      "description": "A unique, friendly name for the asset composite model.",
      "$ref": "#/definitions/Name"
    },
    "type": {
      "description": "The type of the composite model. For alarm composite models,
this type is AWS/ALARM.",
      "$ref": "#/definitions/Name"
    },
    "properties": {
      "description": "The property definitions of the asset model.",
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetModelProperty"
      }
    }
  },
  "AssetModelProperty": {
    "description": "Contains information about an asset model property.",
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "id"
        ]
      },
      {
        "required": [
          "externalId"
        ]
      }
    ],
    "required": [
      "name",
      "dataType",
      "type"
    ]
  },

```

```

"properties": {
  "id": {
    "description": "The ID of the asset model property.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of the asset model property.",
    "$ref": "#/definitions/ExternalId"
  },
  "name": {
    "description": "The name of the asset model property.",
    "$ref": "#/definitions/Name"
  },
  "dataType": {
    "description": "The data type of the asset model property.",
    "$ref": "#/definitions/DataType"
  },
  "dataTypeSpec": {
    "description": "The data type of the structure for this property.",
    "$ref": "#/definitions/Name"
  },
  "unit": {
    "description": "The unit of the asset model property, such as Newtons or
RPM.",
    "type": "string",
    "minLength": 1,
    "maxLength": 256,
    "pattern": "[^\\u0000-\\u001F\\u007F]+"
  },
  "type": {
    "description": "The property type",
    "$ref": "#/definitions/PropertyType"
  }
},
"DataType": {
  "type": "string",
  "enum": [
    "STRING",
    "INTEGER",
    "DOUBLE",
    "BOOLEAN",
    "STRUCT"
  ]
}

```

```

    },
    "PropertyType": {
      "description": "Contains a property type, which can be one of attribute,
measurement, metric, or transform.",
      "type": "object",
      "additionalProperties": false,
      "properties": {
        "attribute": {
          "$ref": "#/definitions/Attribute"
        },
        "transform": {
          "$ref": "#/definitions/Transform"
        },
        "metric": {
          "$ref": "#/definitions/Metric"
        },
        "measurement": {
          "$ref": "#/definitions/Measurement"
        }
      }
    },
  },
  "Attribute": {
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "defaultValue": {
        "type": "string",
        "pattern": "^[^\\u0000-\\u001F\\u007F]+"
      }
    }
  },
  "Transform": {
    "type": "object",
    "additionalProperties": false,
    "required": [
      "expression",
      "variables"
    ],
    "properties": {
      "expression": {
        "description": "The mathematical expression that defines the transformation
function.",
        "type": "string",
        "minLength": 1,

```

```

        "maxLength": 1024
    },
    "variables": {
        "description": "The list of variables used in the expression.",
        "type": "array",
        "items": {
            "$ref": "#/definitions/ExpressionVariable"
        }
    },
    "processingConfig": {
        "$ref": "#/definitions/TransformProcessingConfig"
    }
},
"TransformProcessingConfig": {
    "description": "The processing configuration for the given transform property.",
    "type": "object",
    "additionalProperties": false,
    "required": [
        "computeLocation"
    ],
    "properties": {
        "computeLocation": {
            "description": "The compute location for the given transform property.",
            "$ref": "#/definitions/ComputeLocation"
        },
        "forwardingConfig": {
            "description": "The forwarding configuration for a given property.",
            "$ref": "#/definitions/ForwardingConfig"
        }
    }
},
"Metric": {
    "type": "object",
    "additionalProperties": false,
    "required": [
        "expression",
        "variables",
        "window"
    ],
    "properties": {
        "expression": {
            "description": "The mathematical expression that defines the metric aggregation function.",

```

```

    "type": "string",
    "minLength": 1,
    "maxLength": 1024
  },
  "variables": {
    "description": "The list of variables used in the expression.",
    "type": "array",
    "items": {
      "$ref": "#/definitions/ExpressionVariable"
    }
  },
  "window": {
    "description": "The window (time interval) over which AWS IoT SiteWise
computes the metric's aggregation expression",
    "$ref": "#/definitions/MetricWindow"
  },
  "processingConfig": {
    "$ref": "#/definitions/MetricProcessingConfig"
  }
}
},
"MetricProcessingConfig": {
  "description": "The processing configuration for the metric.",
  "type": "object",
  "additionalProperties": false,
  "required": [
    "computeLocation"
  ],
  "properties": {
    "computeLocation": {
      "description": "The compute location for the given metric property.",
      "$ref": "#/definitions/ComputeLocation"
    }
  }
},
"ComputeLocation": {
  "type": "string",
  "enum": [
    "EDGE",
    "CLOUD"
  ]
},
"ForwardingConfig": {
  "type": "object",

```



```

    "additionalProperties": false,
    "required": [
      "state"
    ],
    "properties": {
      "state": {
        "type": "string",
        "enum": [
          "ENABLED",
          "DISABLED"
        ]
      }
    }
  },
  "MetricWindow": {
    "description": "Contains a time interval window used for data aggregate computations (for example, average, sum, count, and so on).",
    "type": "object",
    "additionalProperties": false,
    "properties": {
      "tumbling": {
        "description": "The tumbling time interval window.",
        "type": "object",
        "additionalProperties": false,
        "required": [
          "interval"
        ],
        "properties": {
          "interval": {
            "description": "The time interval for the tumbling window.",
            "type": "string",
            "minLength": 2,
            "maxLength": 23
          },
          "offset": {
            "description": "The offset for the tumbling window.",
            "type": "string",
            "minLength": 2,
            "maxLength": 25
          }
        }
      }
    }
  }
},

```

```

"ExpressionVariable": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "name",
    "value"
  ],
  "properties": {
    "name": {
      "description": "The friendly name of the variable to be used in the
expression.",
      "type": "string",
      "minLength": 1,
      "maxLength": 64,
      "pattern": "^[a-z][a-z0-9_]*$"
    },
    "value": {
      "description": "The variable that identifies an asset property from which to
use values.",
      "$ref": "#/definitions/VariableValue"
    }
  }
},
"VariableValue": {
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "propertyId"
      ]
    },
    {
      "required": [
        "propertyExternalId"
      ]
    }
  ],
  "properties": {
    "propertyId": {
      "$ref": "#/definitions/ID"
    },
    "propertyExternalId": {
      "$ref": "#/definitions/ExternalId"
    }
  }
}

```

```

    },
    "hierarchyId": {
      "$ref": "#/definitions/ID"
    },
    "hierarchyExternalId": {
      "$ref": "#/definitions/ExternalId"
    }
  }
},
"Measurement": {
  "type": "object",
  "additionalProperties": false,
  "properties": {
    "processingConfig": {
      "$ref": "#/definitions/MeasurementProcessingConfig"
    }
  }
},
"MeasurementProcessingConfig": {
  "type": "object",
  "additionalProperties": false,
  "required": [
    "forwardingConfig"
  ],
  "properties": {
    "forwardingConfig": {
      "description": "The forwarding configuration for the given measurement
property.",
      "$ref": "#/definitions/ForwardingConfig"
    }
  }
},
"AssetModelHierarchy": {
  "description": "Contains information about an asset model hierarchy.",
  "type": "object",
  "additionalProperties": false,
  "anyOf": [
    {
      "required": [
        "id",
        "childAssetModelId"
      ]
    }
  ],
  {

```

```

    "required": [
      "id",
      "childAssetModelExternalId"
    ],
  },
  {
    "required": [
      "externalId",
      "childAssetModelId"
    ],
  },
  {
    "required": [
      "externalId",
      "childAssetModelExternalId"
    ],
  }
],
"required": [
  "name"
],
"properties": {
  "id": {
    "description": "The ID of the asset model hierarchy.",
    "$ref": "#/definitions/ID"
  },
  "externalId": {
    "description": "The ExternalID of the asset model hierarchy.",
    "$ref": "#/definitions/ExternalId"
  },
  "name": {
    "description": "The name of the asset model hierarchy.",
    "$ref": "#/definitions/Name"
  },
  "childAssetModelId": {
    "description": "The ID of the asset model. All assets in this hierarchy must
be instances of the child AssetModelId asset model.",
    "$ref": "#/definitions/ID"
  },
  "childAssetModelExternalId": {
    "description": "The ExternalID of the asset model. All assets in this
hierarchy must be instances of the child AssetModelId asset model.",
    "$ref": "#/definitions/ExternalId"
  }
}

```

```

    }
  },
  "AssetModel": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "assetModelId"
        ]
      },
      {
        "required": [
          "assetModelExternalId"
        ]
      }
    ],
    "required": [
      "assetModelName"
    ],
    "properties": {
      "assetModelId": {
        "description": "The ID of the asset model.",
        "$ref": "#/definitions/ID"
      },
      "assetModelExternalId": {
        "description": "The ID of the asset model.",
        "$ref": "#/definitions/ExternalId"
      },
      "assetModelName": {
        "description": "A unique, friendly name for the asset model.",
        "$ref": "#/definitions/Name"
      },
      "assetModelDescription": {
        "description": "A description for the asset model.",
        "$ref": "#/definitions/Description"
      },
      "assetModelType": {
        "description": "The type of the asset model.",
        "$ref": "#/definitions/AssetModelType"
      },
      "assetModelProperties": {
        "description": "The property definitions of the asset model.",
        "type": "array",

```

```

      "items": {
        "$ref": "#/definitions/AssetModelProperty"
      }
    },
    "assetModelCompositeModels": {
      "description": "The composite asset models that are part of this asset model. Composite asset models are asset models that contain specific properties.",
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetModelCompositeModel"
      }
    },
    "assetModelHierarchies": {
      "description": "The hierarchy definitions of the asset model. Each hierarchy specifies an asset model whose assets can be children of any other assets created from this asset model.",
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetModelHierarchy"
      }
    },
    "tags": {
      "description": "A list of key-value pairs that contain metadata for the asset model.",
      "type": "array",
      "items": {
        "$ref": "#/definitions/Tag"
      }
    }
  },
  "Asset": {
    "type": "object",
    "additionalProperties": false,
    "anyOf": [
      {
        "required": [
          "assetId",
          "assetModelId"
        ]
      },
      {
        "required": [
          "assetExternalId",

```

```

        "assetModelId"
      ]
    },
    {
      "required": [
        "assetId",
        "assetModelExternalId"
      ]
    },
    {
      "required": [
        "assetExternalId",
        "assetModelExternalId"
      ]
    }
  ],
  "required": [
    "assetName"
  ],
  "properties": {
    "assetId": {
      "description": "The ID of the asset",
      "$ref": "#/definitions/ID"
    },
    "assetExternalId": {
      "description": "The external ID of the asset",
      "$ref": "#/definitions/ExternalId"
    },
    "assetModelId": {
      "description": "The ID of the asset model from which to create the asset.",
      "$ref": "#/definitions/ID"
    },
    "assetModelExternalId": {
      "description": "The ExternalID of the asset model from which to create the
asset.",
      "$ref": "#/definitions/ExternalId"
    },
    "assetName": {
      "description": "A unique, friendly name for the asset.",
      "$ref": "#/definitions/Name"
    },
    "assetDescription": {
      "description": "A description for the asset",
      "$ref": "#/definitions/Description"
    }
  }
}

```

```

    },
    "assetProperties": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetProperty"
      }
    },
    "assetHierarchies": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AssetHierarchy"
      }
    },
    "tags": {
      "description": "A list of key-value pairs that contain metadata for the
asset.",
      "type": "array",
      "uniqueItems": false,
      "items": {
        "$ref": "#/definitions/Tag"
      }
    }
  }
},
"additionalProperties": false,
"properties": {
  "assetModels": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/AssetModel"
    }
  },
  "assets": {
    "type": "array",
    "uniqueItems": false,
    "items": {
      "$ref": "#/definitions/Asset"
    }
  }
}
}

```


Surveillez les données avec des alarmes intégrées AWS IoT SiteWise

Vous pouvez configurer des alarmes pour vos données afin d'avertir votre équipe lorsque l'équipement ou les processus ne fonctionnent pas de manière optimale. Les performances optimales d'une machine ou d'un procédé signifient que les valeurs de certaines métriques doivent se situer dans une plage de limites élevées et basses. Lorsque ces mesures sont en dehors de leur plage de fonctionnement, les opérateurs d'équipement doivent être avertis afin qu'ils puissent résoudre le problème. Utilisez les alarmes pour identifier rapidement les problèmes et avertir les opérateurs afin d'optimiser les performances de votre équipement et de vos processus.

Rubriques

- [Types d'alarmes](#)
- [États d'alarme](#)
- [Propriétés de l'état de l'alarme](#)
- [Définissez des alarmes sur les modèles d'actifs dans AWS IoT SiteWise](#)
- [Configurer les alarmes sur les actifs dans AWS IoT SiteWise](#)
- [Répondez aux alarmes dans AWS IoT SiteWise](#)
- [Ingérer un état d'alarme externe dans AWS IoT SiteWise](#)

Types d'alarmes

Vous pouvez définir des alarmes qui sont détectées dans le AWS cloud et des alarmes que vous détectez à l'aide de processus externes. AWS IoT SiteWise prend en charge les types d'alarmes suivants :

- AWS IoT Events alarmes

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 Events les alarmes sont des alarmes qui détectent AWS IoT Events. AWS IoT SiteWise envoie les valeurs des propriétés des actifs à un modèle d'alarme dans AWS IoT Events. AWS IoT Events Envoie ensuite l'état de l'alarme à AWS IoT SiteWise. Vous pouvez configurer des options telles que le moment où l'alarme est détectée et les personnes à avertir lorsque l'état de l'alarme change. Vous pouvez également définir les [AWS IoT Events actions](#) qui se produisent lorsque l'état de l'alarme change.

Les alarmes in AWS IoT Events sont des instances de modèles d'alarme. Le modèle d'alarme indique le seuil et la gravité de l'alarme, ce qu'il faut faire lorsque l'état de l'alarme change, etc. Lorsque vous configurez chaque caractéristique du modèle d'alarme, vous spécifiez une propriété d'attribut à partir du modèle d'actif surveillé par l'alarme. Tous les actifs basés sur le modèle d'actif utilisent la valeur de l'attribut lors de l' AWS IoT Events évaluation de cette caractéristique de l'alarme. Pour plus d'informations, consultez la section [Utilisation des alarmes](#) dans le Guide du AWS IoT Events développeur.

Vous pouvez répondre à une AWS IoT Events alarme lorsqu'elle change d'état. Par exemple, vous pouvez accuser réception d'une alarme ou la suspendre lorsqu'elle devient active. Vous pouvez également activer, désactiver et réinitialiser les alarmes.

SiteWise Les utilisateurs de Monitor peuvent visualiser, configurer et répondre aux AWS IoT Events alarmes dans les portails SiteWise Monitor. Pour plus d'informations, consultez la section [Surveillance par alarmes](#) dans le guide AWS IoT SiteWise Monitor d'application.

 Note

AWS IoT Events des frais s'appliquent pour évaluer ces alarmes et transférer des données entre AWS IoT SiteWise et AWS IoT Events. Pour en savoir plus, consultez [Pricing AWS IoT Events](#) (Tarification).

- Alarmes externes

Les alarmes externes sont des alarmes que vous évaluez en dehors de celles-ci AWS IoT SiteWise. Utilisez des alarmes externes si vous disposez d'une source de données signalant l'état des alarmes. L'alarme externe contient une propriété de mesure dans laquelle vous ingérez les données d'état de l'alarme.

Vous ne pouvez pas accuser réception ou suspendre une alarme externe lorsqu'elle change d'état.

SiteWise Les utilisateurs de Monitor peuvent voir l'état des alarmes externes dans les portails SiteWise Monitor, mais ils ne peuvent ni configurer ces alarmes ni y répondre.

AWS IoT SiteWise n'évalue pas l'état des alarmes externes.

États d'alarme

Les alarmes industrielles incluent des informations sur l'état de l'équipement ou du processus qu'elles surveillent et des informations (facultatives) sur la réponse de l'opérateur à l'état d'alarme.

Lorsque vous définissez une AWS IoT Events alarme, vous indiquez s'il faut activer ou non le flux d'accusé de réception. Le flux d'accusé de réception est activé par défaut. Lorsque vous activez cette option, les opérateurs peuvent accuser réception de l'alarme et laisser une note contenant des détails sur l'alarme ou les mesures qu'ils ont prises pour y remédier. Si un opérateur n'accuse pas réception d'une alarme active avant qu'elle ne devienne inactive, l'alarme se verrouille. L'état verrouillé indique que l'alarme est devenue active et n'a pas été confirmée. L'opérateur doit donc vérifier l'équipement ou le processus et accuser réception de l'alarme verrouillée.

Les alarmes présentent les états suivants :

- **Normal (Normal)** — L'alarme est activée mais inactive. Le procédé ou l'équipement industriel fonctionne comme prévu.
- **Active (Active)** — L'alarme est active. Le procédé ou l'équipement industriel se situe en dehors de sa plage de fonctionnement et nécessite une attention particulière.
- **Reconnu (Acknowledged)** — Un opérateur a confirmé l'état de l'alarme.

Cet état s'applique uniquement aux alarmes pour lesquelles vous activez le flux d'accusé de réception.

- **Verrouillé (Latched)** — L'alarme est revenue à la normale mais elle était active et aucun opérateur ne l'a reconnue. Le processus ou l'équipement industriel nécessite l'attention d'un opérateur pour remettre l'alarme à la normale.

Cet état s'applique uniquement aux alarmes pour lesquelles vous activez le flux d'accusé de réception.

- **Snoozed (SnoozeDisabled)** — L'alarme est désactivée car un opérateur l'a mise en veille. L'opérateur définit la durée pendant laquelle l'alarme se met en veille. Après cette durée, l'alarme revient à l'état normal.

- Désactivé (Disabled) — L'alarme est désactivée et ne sera pas détectée.

Propriétés de l'état de l'alarme

AWS IoT SiteWise stocke les données d'état d'alarme sous forme d'objet JSON sérialisé sous forme de chaîne. Cet objet contient l'état et des informations supplémentaires sur l'alarme, telles que les actions de réponse de l'opérateur et la règle évaluée par l'alarme.

Vous identifiez la propriété d'état de l'alarme par son nom et son type de structure AWS/ALARM_STATE. Pour de plus amples informations, veuillez consulter [Définissez des alarmes sur les modèles d'actifs dans AWS IoT SiteWise](#).

L'objet de données sur l'état de l'alarme contient les informations suivantes :

`stateName`

État de l'alarme. Pour de plus amples informations, veuillez consulter [États d'alarme](#).

Type de données : STRING

`customerAction`

(Facultatif) Objet contenant des informations sur la réponse de l'opérateur à l'alarme. Les opérateurs peuvent activer, désactiver, accuser réception et suspendre les alarmes. Lorsqu'ils le font, les données d'état de l'alarme incluent leur réponse et la note qu'ils peuvent laisser lorsqu'ils répondent. Cet objet contient les informations suivantes :

`actionName`

Nom de l'action entreprise par l'opérateur pour répondre à l'alarme. Cette valeur contient l'une des chaînes suivantes :

- ENABLE
- DISABLE
- SNOOZE
- ACKNOWLEDGE
- RESET

Type de données : STRING

enable

(Facultatif) Objet présent `customerAction` lorsque l'opérateur active l'alarme. Lorsqu'un opérateur active l'alarme, l'état de l'alarme passe à `Normal`. Cet objet contient les informations suivantes :

note

(Facultatif) La note que le client quitte lorsqu'il active l'alarme.

Type de données : `STRING`

Longueur maximale : 128 caractères

disable

(Facultatif) Objet présent `customerAction` lorsque l'opérateur désactive l'alarme. Lorsqu'un opérateur active l'alarme, l'état de l'alarme passe à `Disabled`. Cet objet contient les informations suivantes :

note

(Facultatif) La note que le client quitte lorsqu'il désactive l'alarme.

Type de données : `STRING`

Longueur maximale : 128 caractères

acknowledge

(Facultatif) Objet présent `customerAction` lorsque l'opérateur confirme l'alarme. Lorsqu'un opérateur active l'alarme, l'état de l'alarme passe à `Acknowledged`. Cet objet contient les informations suivantes :

note

(Facultatif) La note que le client quitte lorsqu'il accuse réception de l'alarme.

Type de données : `STRING`

Longueur maximale : 128 caractères

snooze

(Facultatif) Objet présent `customerAction` lorsque l'opérateur déclenche l'alarme. Lorsqu'un opérateur active l'alarme, l'état de l'alarme passe à `SnoozeDisabled`. Cet objet contient les informations suivantes :

snoozeDuration

Durée en secondes pendant laquelle l'opérateur met l'alarme en veille. L'alarme passe à Normal l'état après cette durée.

Type de données : INTEGER

note

(Facultatif) La note que le client quitte lorsqu'il déclenche l'alarme.

Type de données : STRING

Longueur maximale : 128 caractères

ruleEvaluation

(Facultatif) Objet contenant des informations sur la règle qui évalue l'alarme. Cet objet contient les informations suivantes :

simpleRule

Objet contenant des informations relatives à une règle simple, qui compare la valeur d'une propriété à une valeur de seuil à l'aide d'un opérateur de comparaison. Cet objet contient les informations suivantes :

inputProperty

Valeur de la propriété évaluée par cette alarme.

Type de données : DOUBLE

operator

L'opérateur de comparaison utilisé par cette alarme pour comparer la propriété au seuil. Cette valeur contient l'une des chaînes suivantes :

- <— Inférieur à
- <=— Inférieur ou égal
- ==— Égal
- !=— Pas égal
- >=— Supérieur ou égal

- >— Supérieur à

Type de données : STRING

threshold

La valeur de seuil à laquelle cette alarme compare la valeur de la propriété.

Type de données : DOUBLE

Définissez des alarmes sur les modèles d'actifs dans AWS IoT SiteWise

Les modèles d'actifs favorisent la standardisation de vos données industrielles et de vos alarmes. Vous pouvez définir des définitions d'alarmes sur les modèles d'actifs afin de standardiser les alarmes pour tous les actifs en fonction d'un modèle d'actif.

Vous utilisez des modèles d'actifs composites pour définir des alarmes sur les modèles d'actifs. Les modèles d'actifs composites sont des modèles d'actifs qui normalisent un ensemble spécifique de propriétés sur un autre modèle d'actif. Les modèles d'actifs composites garantissent la présence de certaines propriétés sur un modèle d'actif. Les alarmes ont des propriétés de type, d'état et (facultatives) de source, de sorte que le modèle composite d'alarme garantit l'existence de ces propriétés.

Chaque modèle d'actif composite possède un type qui définit les propriétés de ce modèle composite. Les modèles composites d'alarme définissent les propriétés du type d'alarme, de l'état de l'alarme et de la source d'alarme (en option). Lorsque vous créez un actif à partir d'un modèle d'actif avec des modèles composites, l'actif inclut les propriétés du modèle composite ainsi que les propriétés que vous spécifiez dans le modèle d'actif.

Chaque propriété d'un modèle composite doit porter le nom qui l'identifie pour son type de modèle composite. Les propriétés du modèle composite prennent en charge les propriétés contenant des types de données complexes. Ces propriétés ont le type de STRUCT données et un dataTypeSpec trait qui spécifie le type de données complexe de la propriété. Les propriétés de type de données complexes contiennent des données JSON sérialisées sous forme de chaînes.

Les modèles composites d'alarme présentent les propriétés suivantes. Chaque propriété doit porter le nom qui l'identifie pour ce type de modèle composite.

Type d'alarme

Type d'alarme. Spécifiez l'un des éléments suivants :

- **IOT_EVENTS**— Une AWS IoT Events alarme. AWS IoT SiteWise envoie des données AWS IoT Events à pour évaluer l'état de cette alarme. Vous devez spécifier la propriété de la source d'alarme pour définir le modèle AWS IoT Events d'alarme pour cette définition d'alarme.

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](#).

- **EXTERNAL**— Une alarme externe. Vous ingérez l'état de l'alarme sous forme de mesure.

Nom de la propriété : `AWS/ALARM_TYPE`

Type de propriété : [attribut](#)

Type de données : `STRING`

État de l'alarme

Les données chronologiques relatives à l'état de l'alarme. Il s'agit d'un objet sérialisé sous la forme d'une chaîne contenant l'état et d'autres informations relatives à l'alarme. Pour de plus amples informations, veuillez consulter [Propriétés de l'état de l'alarme](#).

Nom de la propriété : `AWS/ALARM_STATE`

Type de propriété : [mesure](#)

Type de données : `STRUCT`

Type de structure de données : `AWS/ALARM_STATE`

Source d'alarme

(Facultatif) Le nom de ressource Amazon (ARN) de la ressource qui évalue l'état de l'alarme. Pour les AWS IoT Events alarmes, il s'agit de l'ARN du modèle d'alarme.

Nom de la propriété : `AWS/ALARM_SOURCE`

Type de propriété : [attribut](#)

Type de données : STRING

Exemple Exemple de modèle composite d'alarme

Le modèle d'actif suivant représente une chaudière dotée d'une alarme pour surveiller sa température. AWS IoT SiteWise envoie les données de température AWS IoT Events pour détecter l'alarme.

```
{
  "assetModelName": "Boiler",
  "assetModelDescription": "A boiler that alarms when its temperature exceeds its
limit.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "High Temperature",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ],
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
```

```

        "attribute": {
            "defaultValue": "IOT_EVENTS"
        }
    },
    {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
            "measurement": {}
        }
    },
    {
        "name": "AWS/ALARM_SOURCE",
        "dataType": "STRING",
        "type": {
            "attribute": {}
        }
    }
]
}
]
}

```

Rubriques

- [Exigences relatives aux notifications d'alarme dans AWS IoT SiteWise](#)
- [Définissez des AWS IoT Events alarmes pour AWS IoT SiteWise](#)
- [Définissez les alarmes externes dans AWS IoT SiteWise](#)

Exigences relatives aux notifications d'alarme dans AWS IoT SiteWise

AWS IoT Events utilise une AWS Lambda fonction de votre AWS compte pour envoyer des notifications d'alarme. Vous devez créer cette fonction Lambda dans la même AWS région que vos alarmes pour activer les notifications d'alarme. Cette fonction Lambda utilise [Amazon Simple Notification Service \(Amazon SNS\) pour envoyer des notifications par SMS et Amazon Simple Email Service \(Amazon SES\) pour envoyer des notifications par e-mail](#). Lorsque vous créez l' AWS IoT Events alarme, vous configurez les protocoles et les paramètres utilisés par l'alarme pour envoyer des notifications.

 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 Events fournit un modèle de AWS CloudFormation pile que vous pouvez utiliser pour créer cette fonction Lambda dans votre compte. Pour plus d'informations, consultez la section [Fonction Lambda de notification d'alarme](#) dans le Guide du AWS IoT Events développeur.

Définissez des AWS IoT Events alarmes pour 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](#).

Lorsque vous créez une AWS IoT Events alarme, AWS IoT SiteWise envoie les valeurs des propriétés des actifs AWS IoT Events à pour évaluer l'état de l'alarme. AWS IoT Events les définitions des alarmes dépendent du modèle d'alarme dans lequel vous les définissez AWS IoT Events. Pour définir une AWS IoT Events alarme sur un modèle d'actif, vous définissez un modèle composite d'alarme qui spécifie le modèle AWS IoT Events d'alarme comme propriété de source d'alarme.

AWS IoT Events les alarmes dépendent d'entrées telles que les seuils d'alarme et les paramètres de notification des alarmes. Vous définissez ces entrées sous forme d'attributs dans le modèle d'actif. Vous pouvez ensuite personnaliser ces entrées pour chaque ressource en fonction du modèle. La AWS IoT SiteWise console peut créer ces attributs pour vous. Si vous définissez des alarmes avec l'API AWS CLI or, vous devez définir manuellement ces attributs sur le modèle d'actif.

Vous pouvez également définir d'autres actions qui se produisent lorsque votre alarme est détectée, telles que des actions de notification d'alarme personnalisées. Par exemple, vous pouvez configurer une action qui envoie une notification push à une rubrique Amazon SNS. Pour plus d'informations sur les actions que vous pouvez définir, consultez la section [Utilisation d'autres AWS services](#) dans le Guide du AWS IoT Events développeur.

Lorsque vous mettez à jour ou supprimez un modèle d'actif, vous AWS IoT SiteWise pouvez vérifier si un modèle d'alarme AWS IoT Events surveille une propriété d'actif associée à ce modèle d'actif. Cela vous empêche de supprimer une propriété de ressource actuellement AWS IoT Events utilisée par une alarme. Pour activer cette fonctionnalité dans AWS IoT SiteWise, vous devez avoir `lotevents:ListInputRoutingsautorisation`. Cette autorisation permet AWS IoT SiteWise d'appeler l'opération d'[ListInputRoutings](#)API prise en charge par AWS IoT Events. Pour de plus amples informations, veuillez consulter [ListInputRoutings Autorisation \(facultative\)](#).

 Note

La fonction de notification d'alarme n'est pas disponible dans la région Chine (Pékin).

Rubriques

- [Définir une AWS IoT Events alarme \(AWS IoT SiteWise console\)](#)
- [Définir une AWS IoT Events alarme \(AWS IoT Events console\)](#)
- [Définir une AWS IoT Events alarme \(AWS CLI\)](#)

Définir une AWS IoT Events alarme (AWS IoT SiteWise console)

Vous pouvez utiliser la AWS IoT SiteWise console pour définir une AWS IoT Events alarme sur un modèle d'actif existant. Pour définir une AWS IoT Events alarme sur un nouveau modèle d'actif, créez le modèle d'actif, puis procédez comme suit. Pour de plus amples informations, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

 Important

Chaque alarme nécessite un attribut qui spécifie la valeur de seuil à comparer pour l'alarme. Vous devez définir l'attribut de valeur seuil sur le modèle d'actif avant de pouvoir définir une alarme.

Prenons un exemple où vous souhaitez définir une alarme qui détecte lorsqu'une éolienne dépasse sa vitesse nominale maximale de 50 mi/h. Avant de définir l'alarme, vous devez définir un attribut (Vitesse maximale du vent) avec une valeur par défaut de 50.

Pour définir une AWS IoT Events alarme sur un modèle d'actif

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle d'actif pour lequel vous souhaitez définir une alarme.
4. Choisissez l'onglet Alarme.
5. Choisissez Ajouter une alarme.
6. Dans la section Options du type d'alarme, choisissez AWS IoT Events alarme.
7. Dans la section Détails de l'alarme, procédez comme suit :
 - a. Saisissez un nom pour votre alarme.
 - b. (Facultatif) Entrez une description pour votre alarme.
8. Dans la section Définitions des seuils, vous définissez le moment où l'alarme est détectée et la gravité de l'alarme. Procédez comme suit :
 - a. Sélectionnez la propriété sur laquelle l'alarme est détectée. Chaque fois que cette propriété reçoit une nouvelle valeur, elle AWS IoT SiteWise envoie la valeur AWS IoT Events à pour évaluer l'état de l'alarme.
 - b. Sélectionnez l'opérateur à utiliser pour comparer la propriété à la valeur de seuil. Sélectionnez parmi les options suivantes :
 - < inférieur à
 - <= inférieur ou égal
 - == égal
 - != différent
 - >= supérieur ou égal
 - > supérieur à
 - c. Pour Valeur, sélectionnez la propriété d'attribut à utiliser comme valeur de seuil. AWS IoT Events compare la valeur de la propriété avec la valeur de cet attribut.
 - d. Entrez le niveau de gravité de l'alarme. Utilisez un chiffre que votre équipe comprend pour refléter la gravité de cette alarme.
9. (Facultatif) Dans la section Paramètres de notification - facultatif, procédez comme suit :
 - a. Choisissez Active.

 Note

Si vous choisissez Inactif, vous et votre équipe ne recevrez aucune notification d'alarme.

- b. Dans Destinataire, choisissez le destinataire.

 Important

Vous pouvez envoyer des notifications d'alarme aux AWS IAM Identity Center utilisateurs. Pour utiliser cette fonctionnalité, vous devez activer IAM Identity Center. Vous ne pouvez activer IAM Identity Center que dans une seule AWS région à la fois. Cela signifie que vous ne pouvez définir des notifications d'alarme que dans la région où vous activez IAM Identity Center. Pour plus d'informations, consultez [Démarrer](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

- c. Pour Protocole, choisissez l'une des options suivantes :
- Courrier électronique et texto : l'alarme avertit les utilisateurs de l'IAM Identity Center par SMS et e-mail.
 - E-mail : l'alarme avertit les utilisateurs de l'IAM Identity Center par un message électronique.
 - Texte — L'alarme avertit les utilisateurs de l'IAM Identity Center par un message SMS.
- d. Pour Expéditeur, choisissez l'expéditeur.

 Important

Vous devez vérifier l'adresse e-mail de l'expéditeur dans Amazon Simple Email Service (Amazon SES). Pour plus d'informations, consultez [Vérifier l'identité d'une adresse e-mail](#) dans le manuel Amazon Simple Email Service Developer Guide.

10. Dans la section État de l'actif par défaut, vous pouvez définir l'état par défaut des alarmes créées à partir de ce modèle d'actif.

 Note

Vous activez ou désactivez cette alarme pour les actifs que vous créez à partir de ce modèle d'actif lors d'une étape ultérieure.

11. Dans la section Paramètres avancés, vous pouvez configurer les autorisations, les paramètres de notification supplémentaires, les actions relatives à l'état de l'alarme, le modèle d'alarme dans SiteWise Monitor et le flux d'accusé de réception.

 Note

AWS IoT Events les alarmes nécessitent les rôles de service suivants :

- Rôle censé envoyer AWS IoT Events des valeurs d'état d'alarme à AWS IoT SiteWise.
- Rôle censé envoyer AWS IoT Events des données à Lambda. Vous n'avez besoin de ce rôle que si votre alarme envoie des notifications.

Dans la section Autorisations, procédez comme suit :

- a. Pour AWS IoT Events le rôle, utilisez un rôle existant ou créez-en un avec les autorisations requises. Ce rôle nécessite `iotsitewise:BatchPutAssetPropertyValue` autorisation et une relation de confiance qui permettent à `iotevents.amazonaws.com` d'assumer le rôle.
 - b. Pour le rôle AWS IoT Events Lambda, utilisez un rôle existant ou créez-en un avec les autorisations requises. Ce rôle nécessite les `sso-directory:DescribeUser` autorisations `lambda:InvokeFunction` et une relation de confiance qui permet `iotevents.amazonaws.com` d'assumer le rôle.
12. (Facultatif) Dans la section Paramètres de notification supplémentaires, procédez comme suit :
 - a. Pour l'attribut Recipient, vous définissez un attribut dont la valeur indique le destinataire de la notification. Vous pouvez choisir les utilisateurs d'IAM Identity Center comme destinataires.

Vous pouvez créer un attribut ou utiliser un attribut existant dans le modèle d'actif.

- Si vous choisissez Créer un nouvel attribut de destinataire, spécifiez le nom de l'attribut de destinataire et la valeur par défaut du destinataire (facultatif) pour l'attribut.

- Si vous choisissez Utiliser un attribut de destinataire existant, sélectionnez l'attribut dans Nom de l'attribut du destinataire. L'alarme utilise la valeur par défaut de l'attribut que vous avez choisi.

Vous pouvez remplacer la valeur par défaut de chaque actif que vous créez à partir de ce modèle d'actif.

- b. Pour l'attribut de message personnalisé, vous définissez un attribut dont la valeur indique le message personnalisé à envoyer en plus du message de changement d'état par défaut. Par exemple, vous pouvez spécifier un message qui aidera votre équipe à comprendre comment traiter cette alarme.

Vous pouvez choisir de créer un attribut ou d'utiliser un attribut existant dans le modèle d'actif.

- Si vous choisissez de créer un nouvel attribut de message personnalisé, spécifiez le nom de l'attribut de message personnalisé et la valeur par défaut du message personnalisé (facultatif) pour l'attribut.
- Si vous choisissez Utiliser un attribut de message personnalisé existant, sélectionnez l'attribut dans Nom de l'attribut de message personnalisé. L'alarme utilise la valeur par défaut de l'attribut que vous avez choisi.

Vous pouvez remplacer la valeur par défaut de chaque actif que vous créez à partir de ce modèle d'actif.

- c. Pour Gérer votre fonction Lambda, effectuez l'une des opérations suivantes :
 - Pour AWS IoT SiteWise créer une nouvelle fonction Lambda, choisissez Create a new Lambda from a AWS managed template.
 - Pour utiliser une fonction Lambda existante, choisissez Utiliser une fonction Lambda existante et choisissez le nom de la fonction.

Pour plus d'informations, consultez [la section Gestion des notifications d'alarme](#) dans le Guide du AWS IoT Events développeur.

13. (Facultatif) Dans la section Définir l'action d'état, procédez comme suit :

- a. Choisissez l'action Modifier.

- b. Sous Ajouter des actions d'état d'alarme, ajoutez des actions, puis choisissez Enregistrer.

Vous pouvez ajouter jusqu'à 10 actions.

AWS IoT Events peut effectuer des actions lorsque l'alarme est active. Vous pouvez définir des actions intégrées pour utiliser un temporisateur ou définir une variable, ou envoyer des données à d'autres AWS ressources. Pour plus d'informations, consultez la section [Actions prises en charge](#) dans le Guide du AWS IoT Events développeur.

14. (Facultatif) Sous Gérer le modèle d'alarme dans le SiteWise moniteur - facultatif, choisissez Actif ou Inactif.

Utilisez cette option pour mettre à jour le modèle d'alarme dans SiteWise Monitorss. Cette option est activée par défaut.

15. Sous Accusation du flux, sélectionnez Actif ou Inactif. Pour plus d'informations sur le flux d'accusé de réception, consultez [États d'alarme](#).
16. Choisissez Ajouter une alarme.

Note

La AWS IoT SiteWise console envoie plusieurs demandes d'API pour ajouter l'alarme au modèle d'actif. Lorsque vous choisissez Ajouter une alarme, la console ouvre une boîte de dialogue qui indique la progression de ces demandes d'API. Restez sur cette page jusqu'à ce que chaque demande d'API aboutisse ou jusqu'à ce qu'une demande d'API échoue. Si une demande échoue, fermez la boîte de dialogue, corrigez le problème et choisissez Ajouter une alarme pour réessayer.

Définir une AWS IoT Events alarme (AWS IoT Events console)

Vous pouvez utiliser la AWS IoT Events console pour définir une AWS IoT Events alarme sur un modèle d'actif existant. Pour définir une AWS IoT Events alarme sur un nouveau modèle d'actif, créez le modèle d'actif, puis procédez comme suit. Pour de plus amples informations, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

 Important

Chaque alarme nécessite un attribut qui spécifie la valeur de seuil à comparer pour l'alarme. Vous devez définir l'attribut de valeur seuil sur le modèle d'actif avant de pouvoir définir une alarme.

Prenons un exemple où vous souhaitez définir une alarme qui détecte lorsqu'une éolienne dépasse sa vitesse nominale maximale de 50 mi/h. Avant de définir l'alarme, vous devez définir un attribut (Vitesse maximale du vent) avec une valeur par défaut de 50.

Pour définir une AWS IoT Events alarme sur un modèle d'actif

1. Accédez à la [console AWS IoT Events](#).
2. Dans le volet de navigation, sélectionnez Modèles d'alarme.
3. Choisissez Créer un modèle d'alarme.
4. Saisissez un nom pour votre alarme.
5. (Facultatif) Entrez une description pour votre alarme.
6. Dans la section Cible de l'alarme, procédez comme suit :
 - a. Pour les options Target, choisissez la propriété de l'AWS IoT SiteWise actif.
 - b. Choisissez le modèle d'actif pour lequel vous souhaitez ajouter l'alarme.
7. Dans la section Définitions des seuils, vous définissez le moment où l'alarme est détectée et la gravité de l'alarme. Procédez comme suit :
 - a. Sélectionnez la propriété sur laquelle l'alarme est détectée. Chaque fois que cette propriété reçoit une nouvelle valeur, elle AWS IoT SiteWise envoie la valeur AWS IoT Events à pour évaluer l'état de l'alarme.
 - b. Sélectionnez l'opérateur à utiliser pour comparer la propriété à la valeur de seuil. Sélectionnez parmi les options suivantes :
 - < inférieur à
 - <= inférieur ou égal
 - == égal
 - != différent
 - >= supérieur ou égal

- > supérieur à
 - c. Pour Valeur, sélectionnez la propriété d'attribut à utiliser comme valeur de seuil. AWS IoT Events compare la valeur de la propriété avec la valeur de cet attribut.
 - d. Entrez le niveau de gravité de l'alarme. Utilisez un chiffre que votre équipe comprend pour refléter la gravité de cette alarme.
8. (Facultatif) Dans la section Paramètres de notification - facultatif, procédez comme suit :
- a. Pour Protocole, choisissez l'une des options suivantes :
 - Courrier électronique et texto : l'alarme avertit les utilisateurs de l'IAM Identity Center par SMS et e-mail.
 - E-mail : l'alarme avertit les utilisateurs de l'IAM Identity Center par un message électronique.
 - Texte — L'alarme avertit les utilisateurs de l'IAM Identity Center par un message SMS.
 - b. Pour Expéditeur, choisissez l'expéditeur.
-
- Important
- Vous devez vérifier l'adresse e-mail de l'expéditeur dans Amazon Simple Email Service (Amazon SES). Pour plus d'informations, consultez la section [Vérification des adresses e-mail dans Amazon SES](#), dans le manuel Amazon Simple Email Service Developer Guide.
- c. Choisissez l'attribut dans Attribut du destinataire (facultatif). L'alarme utilise la valeur par défaut de l'attribut que vous avez choisi.
 - d. Choisissez l'attribut dans Attribut de message personnalisé - facultatif. L'alarme utilise la valeur par défaut de l'attribut que vous avez choisi.
9. Dans la section Instance, spécifiez l'état par défaut pour cette alarme. Vous pouvez activer ou désactiver cette alarme pour tous les actifs que vous créez à partir de ce modèle d'actif lors d'une étape ultérieure.
10. Dans les paramètres avancés, vous pouvez configurer les autorisations, les paramètres de notification supplémentaires, les actions relatives à l'état de l'alarme, le modèle d'alarme dans SiteWise Monitor et le flux d'accusé de réception.

 Note

AWS IoT Events les alarmes nécessitent les rôles de service suivants :

- Rôle censé envoyer AWS IoT Events des valeurs d'état d'alarme à AWS IoT SiteWise.
- Rôle censé envoyer AWS IoT Events des données à Lambda. Vous n'avez besoin de ce rôle que si votre alarme envoie des notifications.

- a. Dans la section Accusation de réception, choisissez Activé ou Désactivé. Pour plus d'informations sur le flux d'accusé de réception, consultez [États d'alarme](#).
- b. Dans la section Autorisations, procédez comme suit :
 - i. Pour AWS IoT Events le rôle, utilisez un rôle existant ou créez-en un avec les autorisations requises. Ce rôle nécessite `l:otsitewise:BatchPutAssetPropertyValue` autorisation et une relation de confiance qui permettent à `iotevents.amazonaws.com` d'assumer le rôle.
 - ii. Pour le rôle Lambda, utilisez un rôle existant ou créez-en un avec les autorisations requises. Ce rôle nécessite les `sso-directory:DescribeUser` autorisations `lambda:InvokeFunction` et une relation de confiance qui permet `iotevents.amazonaws.com` d'assumer le rôle.
- c. (Facultatif) Dans le volet Paramètres de notification supplémentaires, procédez comme suit :
 - Pour Gérer votre fonction Lambda, effectuez l'une des opérations suivantes :
 - Pour AWS IoT Events créer une nouvelle fonction Lambda, choisissez Create a new Lambda function.
 - Pour utiliser une fonction Lambda existante, choisissez Utiliser une fonction Lambda existante et choisissez le nom de la fonction.

Pour plus d'informations, consultez [la section Gestion des notifications d'alarme](#) dans le Guide du AWS IoT Events développeur.

- d. (Facultatif) Dans la section Définir l'action d'état - facultatif, procédez comme suit :
 - Sous Actions relatives à l'état d'alarme, ajoutez des actions, puis choisissez Enregistrer.

Vous pouvez ajouter jusqu'à 10 actions.

AWS IoT Events peut effectuer des actions lorsque l'alarme est active. Vous pouvez définir des actions intégrées pour utiliser un temporisateur ou définir une variable, ou envoyer des données à d'autres AWS ressources. Pour plus d'informations, consultez la section [Actions prises en charge](#) dans le Guide du AWS IoT Events développeur.

11. Choisissez Créer.

Note

La AWS IoT Events console envoie plusieurs demandes d'API pour ajouter l'alarme au modèle d'actif. Lorsque vous choisissez Ajouter une alarme, la console ouvre une boîte de dialogue qui indique la progression de ces demandes d'API. Restez sur cette page jusqu'à ce que chaque demande d'API aboutisse ou jusqu'à ce qu'une demande d'API échoue. Si une demande échoue, fermez la boîte de dialogue, corrigez le problème et choisissez Ajouter une alarme pour réessayer.

Définir une AWS IoT Events alarme (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour définir une AWS IoT Events alarme qui surveille la propriété d'un actif. Vous pouvez définir l'alarme sur un modèle d'actif nouveau ou existant. Après avoir défini l'alarme sur le modèle d'actif, vous créez une alarme AWS IoT Events et vous la connectez au modèle d'actif. Au cours de ce processus, vous devez effectuer les opérations suivantes :

Étapes

- [Étape 1 : définir une alarme sur un modèle d'actif](#)
- [Étape 2 : Définition d'un modèle AWS IoT Events d'alarme](#)
- [Étape 3 : Activer le flux de données entre AWS IoT SiteWise et AWS IoT Events](#)

Étape 1 : définir une alarme sur un modèle d'actif

Ajoutez une définition d'alarme et les propriétés associées à un modèle d'actif nouveau ou existant.

Pour définir une alarme sur un modèle d'actif (CLI)

1. Créez un fichier, appelé `asset-model-payload.json`. Suivez les étapes décrites dans ces autres sections pour ajouter les détails de votre modèle d'actif au fichier, mais ne soumettez pas de demande de création ou de mise à jour du modèle d'actif. Dans cette section, vous ajoutez une définition d'alarme aux détails du modèle d'actif dans le `asset-model-payload.json` fichier.
 - Pour plus d'informations sur la création d'un modèle d'actifs, consultez [Création d'un modèle d'actifs \(AWS CLI\)](#).
 - Pour plus d'informations sur la mise à jour d'un modèle d'actif existant, consultez [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#).

Note

Votre modèle d'actif doit définir au moins une propriété d'actif, y compris la propriété d'actif à surveiller avec l'alarme.

2. Ajoutez un modèle composite d'alarme (`assetModelCompositeModels`) au modèle d'actif. Un modèle composite AWS IoT Events d'alarme indique le `IOT_EVENTS` type et spécifie une propriété de source d'alarme. Vous ajoutez la propriété source d'alarme après avoir créé le modèle d'alarme dans AWS IoT Events.

Important

Le modèle composite d'alarme doit porter le même nom que le modèle AWS IoT Events d'alarme que vous créerez ultérieurement. Les noms des modèles d'alarme ne peuvent contenir que des caractères alphanumériques. Spécifiez un nom alphanumérique unique afin de pouvoir utiliser le même nom pour le modèle d'alarme.

```
{  
  ...  
  "assetModelCompositeModels": [  
    {  
      "name": "BoilerTemperatureHighAlarm",  
      "type": "AWS/ALARM",  
      "properties": [  
        {
```

```

        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
            "attribute": {
                "defaultValue": "IOT_EVENTS"
            }
        }
    },
    {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
            "measurement": {}
        }
    }
]
}
]
}

```

3. Ajoutez un attribut de seuil d'alarme au modèle d'actif. Spécifiez la valeur par défaut à utiliser pour ce seuil. Vous pouvez remplacer cette valeur par défaut pour chaque actif en fonction de ce modèle.

Note

L'attribut du seuil d'alarme doit être a INTEGER ou DOUBLE a.

```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "Temperature Max Threshold",
            "dataType": "DOUBLE",
            "type": {
                "attribute": {
                    "defaultValue": "105.0"
                }
            }
        }
    ]
}

```

```
    }
  ]
}
```

4. (Facultatif) Ajoutez des attributs de notification d'alarme au modèle d'actif. Ces attributs indiquent le destinataire du centre d'identité IAM et les autres entrées AWS IoT Events utilisées pour envoyer des notifications lorsque l'alarme change d'état. Vous pouvez remplacer ces valeurs par défaut pour chaque actif en fonction de ce modèle.

Important

Vous pouvez envoyer des notifications d'alarme aux AWS IAM Identity Center utilisateurs. Pour utiliser cette fonctionnalité, vous devez activer IAM Identity Center. Vous ne pouvez activer IAM Identity Center que dans une seule AWS région à la fois. Cela signifie que vous ne pouvez définir des notifications d'alarme que dans la région où vous activez IAM Identity Center. Pour plus d'informations, consultez [Démarrer](#) dans le Guide de l'utilisateur AWS IAM Identity Center .

Procédez comme suit :

- a. Ajoutez un attribut qui spécifie l'ID de votre banque d'identités IAM Identity Center. Vous pouvez utiliser l'opération d'[ListInstances](#) API IAM Identity Center pour répertorier vos magasins d'identités. Cette opération ne fonctionne que dans la région où vous activez IAM Identity Center.

```
aws sso-admin list-instances
```

Spécifiez ensuite l'ID du magasin d'identités (par exemple, d-123EXAMPLE) comme valeur par défaut pour l'attribut.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "identityStoreId",
      "dataType": "STRING",
      "type": {
```



```

        "attribute": {
            "defaultValue": "d-123EXAMPLE"
        }
    }
}
]
}

```

- b. Ajoutez un attribut qui spécifie l'ID de l'utilisateur IAM Identity Center qui reçoit les notifications. Pour définir un destinataire de notification par défaut, ajoutez un ID utilisateur IAM Identity Center comme valeur par défaut. Procédez de l'une des manières suivantes pour obtenir un ID utilisateur IAM Identity Center :

- i. Vous pouvez utiliser l'[ListUsers](#) API IAM Identity Center pour obtenir l'identifiant d'un utilisateur dont vous connaissez le nom d'utilisateur. Remplacez *d-123EXAMPLE* par l'ID de votre banque d'identités et remplacez *Name* par le nom d'utilisateur de l'utilisateur.

```

aws identitystore list-users \
  --identity-store-id d-123EXAMPLE \
  --filters AttributePath=UserName,AttributeValue=Name

```

- ii. Utilisez la [console IAM Identity Center](#) pour parcourir vos utilisateurs et trouver un ID utilisateur.

Spécifiez ensuite l'ID utilisateur (par exemple, *123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE*) comme valeur par défaut pour l'attribut, ou définissez l'attribut sans valeur par défaut.

```

{
    ...
    "assetModelProperties": [
        ...
        {
            "name": "userId",
            "dataType": "STRING",
            "type": {
                "attribute": {
                    "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
                }
            }
        }
    ]
}

```

```

    }
  ]
}

```

- c. (Facultatif) Ajoutez un attribut qui spécifie l'ID d'expéditeur par défaut pour les notifications par SMS (texte). L'identifiant de l'expéditeur s'affiche en tant qu'expéditeur du message sur les messages envoyés par Amazon Simple Notification Service (Amazon SNS). Pour plus d'informations, consultez la section [Demander un identifiant d'expéditeur AWS End User Messaging SMS dans](#) le guide de AWS End User Messaging SMS l'utilisateur.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "senderId",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "MyFactory"
        }
      }
    }
  ]
}

```

- d. (Facultatif) Ajoutez un attribut qui spécifie l'adresse e-mail par défaut à utiliser comme adresse d'expéditeur dans les notifications par e-mail.

```

{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "fromAddress",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "my.factory@example.com"
        }
      }
    }
  ]
}

```

```
]
}
```

- e. (Facultatif) Ajoutez un attribut qui indique le sujet par défaut à utiliser dans les notifications par e-mail.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "emailSubject",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "[ALERT] High boiler temperature"
        }
      }
    }
  ]
}
```

- f. (Facultatif) Ajoutez un attribut qui spécifie un message supplémentaire à inclure dans les notifications. Par défaut, les messages de notification incluent des informations sur l'alarme. Vous pouvez également inclure un message supplémentaire fournissant plus d'informations à l'utilisateur.

```
{
  ...
  "assetModelProperties": [
    ...
    {
      "name": "additionalMessage",
      "dataType": "STRING",
      "type": {
        "attribute": {
          "defaultValue": "Turn off the power before you check the alarm."
        }
      }
    }
  ]
}
```

5. Créez le modèle d'actif ou mettez à jour le modèle d'actif existant. Effectuez l'une des actions suivantes :

- Pour créer le modèle d'actif, exécutez la commande suivante.

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Pour mettre à jour le modèle d'actif existant, exécutez la commande suivante. Remplacez *asset-model-id* par l'ID du modèle d'actif.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Après avoir exécuté la commande, notez le `assetModelId` dans la réponse.

Exemple : modèle d'actif de chaudière

Le modèle d'actif suivant représente une chaudière qui fournit des données de température. Ce modèle d'équipement définit une alarme qui détecte la surchauffe de la chaudière.

```
{
  "assetModelName": "Boiler Model",
  "assetModelDescription": "Represents a boiler.",
  "assetModelProperties": [
    {
      "name": "Temperature",
      "dataType": "DOUBLE",
      "unit": "C",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature Max Threshold",
      "dataType": "DOUBLE",
      "type": {
        "attribute": {
          "defaultValue": "105.0"
        }
      }
    }
  ]
}
```

```
},
{
  "name": "identityStoreId",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "d-123EXAMPLE"
    }
  }
},
{
  "name": "userId",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "123EXAMPLE-a1b2c3d4-5678-90ab-cdef-33333EXAMPLE"
    }
  }
},
{
  "name": "senderId",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "MyFactory"
    }
  }
},
{
  "name": "fromAddress",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "my.factory@example.com"
    }
  }
},
{
  "name": "emailSubject",
  "dataType": "STRING",
  "type": {
    "attribute": {
      "defaultValue": "[ALERT] High boiler temperature"
    }
  }
}
```

```

    }
  },
  {
    "name": "additionalMessage",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "Turn off the power before you check the alarm."
      }
    }
  }
],
"assetModelHierarchies": [

],
"assetModelCompositeModels": [
  {
    "name": "BoilerTemperatureHighAlarm",
    "type": "AWS/ALARM",
    "properties": [
      {
        "name": "AWS/ALARM_TYPE",
        "dataType": "STRING",
        "type": {
          "attribute": {
            "defaultValue": "IOT_EVENTS"
          }
        }
      },
      {
        "name": "AWS/ALARM_STATE",
        "dataType": "STRUCT",
        "dataTypeSpec": "AWS/ALARM_STATE",
        "type": {
          "measurement": {}
        }
      }
    ]
  }
]
}

```

Étape 2 : Définition d'un modèle AWS IoT Events d'alarme

Créez le modèle d'alarme dans AWS IoT Events. Dans AWS IoT Events, vous utilisez des expressions pour spécifier des valeurs dans les modèles d'alarme. Vous pouvez utiliser des expressions pour spécifier des valeurs de AWS IoT SiteWise à évaluer et à utiliser comme entrées pour l'alarme. Lorsque les valeurs des propriétés de l'actif sont AWS IoT SiteWise envoyées au modèle d'alarme, il AWS IoT Events évalue l'expression pour obtenir la valeur de la propriété ou l'ID de l'actif. Vous pouvez utiliser les expressions suivantes dans le modèle d'alarme :

- Valeurs des propriétés des actifs

Pour obtenir la valeur d'une propriété d'actif, utilisez l'expression suivante. Remplacez *assetModelId* par l'ID du modèle d'actif et remplacez *propertyId* par l'ID de la propriété.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.propertyValue.value
```

- Actif IDs

Pour obtenir l'ID de la ressource, utilisez l'expression suivante. Remplacez *assetModelId* par l'ID du modèle d'actif et remplacez *propertyId* par l'ID de la propriété.

```
$sitewise.assetModel.`assetModelId`.`propertyId`.assetId
```

Note

Lorsque vous créez le modèle d'alarme, vous pouvez définir des littéraux plutôt que des expressions dont l'évaluation est basée sur des AWS IoT SiteWise valeurs. Cela peut réduire le nombre d'attributs que vous définissez dans votre modèle d'actifs. Toutefois, si vous définissez une valeur au sens littéral, vous ne pouvez pas personnaliser cette valeur sur les actifs en fonction du modèle d'actif. Vos AWS IoT SiteWise Monitor utilisateurs ne peuvent pas non plus personnaliser l'alarme, car ils ne peuvent configurer les paramètres d'alarme que sur les actifs.

Pour créer un modèle AWS IoT Events d'alarme (CLI)

1. Lorsque vous créez le modèle d'alarme dans AWS IoT Events, vous devez spécifier l'ID de chaque propriété utilisée par l'alarme, notamment les éléments suivants :

- La propriété d'état d'alarme dans le modèle d'actif composite
- La propriété surveillée par l'alarme
- L'attribut de seuil
- (Facultatif) L'attribut ID de la banque d'identités IAM Identity Center
- (Facultatif) L'attribut d'ID utilisateur IAM Identity Center
- (Facultatif) L'attribut ID de l'expéditeur du SMS
- (Facultatif) L'attribut d'adresse e-mail
- (Facultatif) L'attribut d'objet de l'e-mail
- (Facultatif) L'attribut de message supplémentaire

Exécutez la commande suivante pour récupérer IDs ces propriétés sur le modèle d'actif. Remplacez *asset-model-id* par l'ID du modèle d'actif de l'étape précédente.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

L'opération renvoie une réponse qui contient les détails du modèle de ressource. Notez l'ID de chaque propriété utilisée par l'alarme. Vous les utiliserez IDs lorsque vous créerez le modèle AWS IoT Events d'alarme à l'étape suivante.

2. Créez le modèle d'alarme dans AWS IoT Events. Procédez comme suit :
 - a. Créez un fichier, appelé `alarm-model-payload.json`.
 - b. Copiez l'objet JSON suivant dans le fichier.
 - c. Entrez le nom (`alarmModelName`), la description (`alarmModelDescription`) et la gravité (`severity`) de votre alarme. Pour ce qui est de la gravité, spécifiez un entier qui reflète les niveaux de gravité de votre entreprise.

 Important

Le modèle d'alarme doit porter le même nom que le modèle composite d'alarme que vous avez défini précédemment dans votre modèle d'équipement.
Les noms des modèles d'alarme ne peuvent contenir que des caractères alphanumériques.


```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3
}
```

d. Ajoutez la règle de comparaison (alarmRule) à l'alarme. Cette règle définit la propriété à surveiller (inputProperty), la valeur de seuil à comparer (threshold) et l'opérateur de comparaison à utiliser (comparisonOperator).

- Remplacez *assetModelId* par l'ID du modèle d'actif.
- Remplacez *alarmPropertyId* par l'identifiant de la propriété surveillée par l'alarme.
- Remplacez *thresholdAttributeId* par l'ID de la propriété de l'attribut threshold.
- Remplacez *GREATER* par l'opérateur à utiliser pour comparer les valeurs des propriétés avec le seuil. Sélectionnez parmi les options suivantes :
 - LESS
 - LESS_OR_EQUAL
 - EQUAL
 - NOT_EQUAL
 - GREATER_OR_EQUAL
 - GREATER

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  }
}
```

```
}

```

- e. Ajoutez une action (`alarmEventActions`) pour envoyer l'état de l'alarme au AWS IoT SiteWise moment où l'alarme change d'état.

 Note

Pour une configuration avancée, vous pouvez définir des actions supplémentaires à effectuer lorsque l'alarme change d'état. Par exemple, vous pouvez appeler une AWS Lambda fonction ou publier sur un sujet MQTT. Pour plus d'informations, consultez la section [Utilisation d'autres AWS services](#) dans le Guide du AWS IoT Events développeur.

- Remplacez *assetModelId* par l'ID du modèle d'actif.
- Remplacez *alarmPropertyId* par l'identifiant de la propriété surveillée par l'alarme.
- Remplacez *alarmStatePropertyId* par l'ID de la propriété d'état de l'alarme dans le modèle composite d'alarme.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  }
}
```

```

    }
  ]
}
}

```

- f. (Facultatif) Configurez les paramètres de notification d'alarme. L'action de notification d'alarme utilise une fonction Lambda de votre compte pour envoyer des notifications d'alarme. Pour de plus amples informations, veuillez consulter [Exigences relatives aux notifications d'alarme dans AWS IoT SiteWise](#). Dans les paramètres de notification d'alarme, vous pouvez configurer les notifications par SMS et e-mail à envoyer aux utilisateurs d'IAM Identity Center. Procédez comme suit :
 - i. Ajoutez la configuration de notification d'alarme (alarmNotification) à la charge utile dans alarm-model-payload.json.
 - *alarmNotificationFunctionArn* Remplacez-le par l'ARN de la fonction Lambda qui gère les notifications d'alarme.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  }
}

```

```

    },
    "alarmNotification": {
      "notificationActions": [
        {
          "action": {
            "lambdaAction": {
              "functionArn": "alarmNotificationFunctionArn"
            }
          }
        }
      ]
    }
  }
}

```

- ii. (Facultatif) Configurez les notifications par SMS (smsConfigurations) à envoyer à un utilisateur du IAM Identity Center lorsque l'alarme change d'état.

- *identityStoreIdAttributeId* Remplacez-le par l'ID de l'attribut qui contient l'ID de la banque d'identités IAM Identity Center.
- *userIdAttributeId* Remplacez-le par l'ID de l'attribut qui contient l'ID de l'utilisateur IAM Identity Center.
- *senderIdAttributeId* Remplacez-le par l'ID de l'attribut qui contient l'identifiant de l'expéditeur Amazon SNS, ou supprimez-le senderId de la charge utile.
- *additionalMessageAttributeId* Remplacez-le par l'ID de l'attribut qui contient le message supplémentaire, ou supprimez-le additionalMessage de la charge utile.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId` .propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId` .propertyValue.value"
    }
  },
  "alarmEventActions": {

```

```

    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ],
    "alarmNotification": {
      "notificationActions": [
        {
          "action": {
            "lambdaAction": {
              "functionArn": "alarmNotificationFunctionArn"
            }
          },
          "smsConfigurations": [
            {
              "recipients": [
                {
                  "ssoIdentity": {
                    "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                    "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                  }
                }
              ],
              "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
              "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
            }
          ]
        }
      ]
    }
  }
}

```

- iii. (Facultatif) Configurez les notifications par e-mail (emailConfigurations) à envoyer à un utilisateur du IAM Identity Center lorsque l'alarme change d'état.

- *identityStoreIdAttributeId* Remplacez-le par l'ID de la propriété d'attribut ID de la banque d'identités IAM Identity Center.
- Remplacez *userIdAttributeId* par l'ID de la propriété d'attribut ID utilisateur d'IAM Identity Center.
- *fromAddressAttributeId* Remplacez-le par l'ID de la propriété d'attribut d'adresse « from » ou supprimez-le from de la charge utile.
- *emailSubjectAttributeId* Remplacez-le par l'ID de la propriété d'attribut de l'objet de l'e-mail ou subject supprimez-le de la charge utile.
- *additionalMessageAttributeId* Remplacez-le par l'ID de la propriété d'attribut de message supplémentaire ou supprimez-le additionalMessage de la charge utile.

```
{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "'alarmStatePropertyId'"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
```

```

    "action": {
      "lambdaAction": {
        "functionArn": "alarmNotificationFunctionArn"
      }
    },
    "smsConfigurations": [
      {
        "recipients": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
            }
          }
        ],
        "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId` .propertyValue.value",
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
      }
    ],
    "emailConfigurations": [
      {
        "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId` .propertyValue.value",
        "recipients": {
          "to": [
            {
              "ssoIdentity": {
                "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId` .propertyValue.value",
                "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId` .propertyValue.value"
              }
            }
          ]
        },
        "content": {
          "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId` .propertyValue.value",
          "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId` .propertyValue.value"
        }
      }
    ]
  }
}

```

```

    }
  }
]
}
}

```

- g. (Facultatif) Ajoutez les capacités d'alarme (`alarmCapabilities`) à la charge utile `alarm-model-payload.json`. Dans cet objet, vous pouvez spécifier si le flux d'accusé de réception est activé et l'état d'activation par défaut pour les actifs en fonction du modèle d'actif. Pour plus d'informations sur le flux d'accusé de réception, consultez [États d'alarme](#).

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
"$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {
          "assetId":
"$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
          "propertyId": "`alarmStatePropertyId`"
        }
      }
    ]
  },
  "alarmNotification": {
    "notificationActions": [
      {
        "action": {
          "lambdaAction": {
            "functionArn": "alarmNotificationFunctionArn"
          }
        }
      }
    ]
  }
}

```



```

    }
  },
  "smsConfigurations": [
    {
      "recipients": [
        {
          "ssoIdentity": {
            "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
            "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
          }
        }
      ],
      "senderId":
"$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
      "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
    }
  ],
  "emailConfigurations": [
    {
      "from":
"$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
      "recipients": {
        "to": [
          {
            "ssoIdentity": {
              "identityStoreId":
"$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
              "userId":
"$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
            }
          }
        ]
      },
      "content": {
        "subject":
"$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
        "additionalMessage":
"$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
      }
    }
  ]
}

```

```

    }
  ]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": true
  }
}
}
}

```

- h. Ajoutez le rôle de service IAM (roleArn) qui AWS IoT Events peut assumer d'envoyer des données à AWS IoT SiteWise. Ce rôle nécessite l'`iotsitewise:BatchPutAssetPropertyValue` autorisation et une relation de confiance qui permettent `iotevents.amazonaws.com` d'assumer le rôle. Pour envoyer des notifications, ce rôle nécessite également les `sso-directory:DescribeUser` autorisations `lambda:InvokeFunction` et. Pour plus d'informations, consultez la section [Rôles du service Alarm](#) dans le guide du AWS IoT Events développeur.
- Remplacez le `roleArn` par l'ARN du rôle qui AWS IoT Events peut assumer l'exécution de ces actions.

```

{
  "alarmModelName": "BoilerTemperatureHighAlarm",
  "alarmModelDescription": "Detects when the boiler temperature is high.",
  "severity": 3,
  "alarmRule": {
    "simpleRule": {
      "inputProperty":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.propertyValue.value",
      "comparisonOperator": "GREATER",
      "threshold":
        "$sitewise.assetModel.`assetModelId`.`thresholdAttributeId`.propertyValue.value"
    }
  },
  "alarmEventActions": {
    "alarmActions": [
      {
        "iotSiteWise": {

```

```

        "assetId":
        "$sitewise.assetModel.`assetModelId`.`alarmPropertyId`.assetId",
        "propertyId": "`alarmStatePropertyId`"
    }
}
],
},
"alarmNotification": {
    "notificationActions": [
        {
            "action": {
                "lambdaAction": {
                    "functionArn": "alarmNotificationFunctionArn"
                }
            },
            "smsConfigurations": [
                {
                    "recipients": [
                        {
                            "ssoIdentity": {
                                "identityStoreId":
                                "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value",
                                "userId":
                                "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
                            }
                        }
                    ],
                    "senderId":
                    "$sitewise.assetModel.`assetModelId`.`senderIdAttributeId`.propertyValue.value",
                    "additionalMessage":
                    "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
                }
            ],
            "emailConfigurations": [
                {
                    "from":
                    "$sitewise.assetModel.`assetModelId`.`fromAddressAttributeId`.propertyValue.value",
                    "recipients": {
                        "to": [
                            {
                                "ssoIdentity": {
                                    "identityStoreId":
                                    "$sitewise.assetModel.`assetModelId`.`identityStoreIdAttributeId`.propertyValue.value"
                                }
                            }
                        ]
                    }
                }
            ]
        }
    ]
}

```

```

        "userId":
        "$sitewise.assetModel.`assetModelId`.`userIdAttributeId`.propertyValue.value"
      }
    }
  ],
  },
  "content": {
    "subject":
    "$sitewise.assetModel.`assetModelId`.`emailSubjectAttributeId`.propertyValue.value",
    "additionalMessage":
    "$sitewise.assetModel.`assetModelId`.`additionalMessageAttributeId`.propertyValue.value"
  }
}
]
},
"alarmCapabilities": {
  "initializationConfiguration": {
    "disabledOnInitialization": false
  },
  "acknowledgeFlow": {
    "enabled": false
  }
},
"roleArn": "arn:aws:iam::123456789012:role/MyIoTEventsAlarmRole"
}

```

- i. Exécutez la commande suivante pour créer le modèle AWS IoT Events d'alarme à partir de la charge utile. `alarm-model-payload.json`

```
aws iotevents create-alarm-model --cli-input-json file://alarm-model-payload.json
```

- j. L'opération renvoie une réponse qui inclut l'ARN du modèle d'alarme, `alarmModelArn`. Copiez cet ARN pour définir la définition de l'alarme sur votre modèle d'actif à l'étape suivante.

Étape 3 : Activer le flux de données entre AWS IoT SiteWise et AWS IoT Events

Après avoir créé les ressources requises dans AWS IoT SiteWise et AWS IoT Events, vous pouvez activer le flux de données entre les ressources pour activer votre alarme. Dans cette section, vous

allez mettre à jour la définition d'alarme dans le modèle d'actif pour utiliser le modèle d'alarme que vous avez créé à l'étape précédente.

Pour activer le flux de données entre AWS IoT SiteWise et AWS IoT Events (CLI)

- Définissez le modèle d'alarme comme source de l'alarme dans le modèle d'actif. Procédez comme suit :
 - a. Exécutez la commande suivante pour récupérer la définition de modèle de ressource existante. Remplacez *asset-model-id* par l'ID du modèle d'actif.

```
aws iotsitewise describe-asset-model --asset-model-id asset-model-id
```

L'opération renvoie une réponse qui contient les détails du modèle de ressource.

- b. Créez un fichier appelé `update-asset-model-payload.json` et copiez la réponse de la commande précédente dans le fichier.
- c. Supprimez les paires clé-valeur suivantes du `update-asset-model-payload.json` fichier :
 - `assetModelId`
 - `assetModelArn`
 - `assetModelCreationDate`
 - `assetModelLastUpdateDate`
 - `assetModelStatus`
- d. Ajoutez la propriété de source d'alarme (`AWS/ALARM_SOURCE`) au modèle composite d'alarme que vous avez défini précédemment. *alarmModelArn* Remplacez-le par l'ARN du modèle d'alarme, qui définit la valeur de la propriété de la source d'alarme.

```
{
  ...
  "assetModelCompositeModels": [
    ...
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
```

```

    "name": "AWS/ALARM_TYPE",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "IOT_EVENTS"
      }
    }
  },
  {
    "id": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
    "name": "AWS/ALARM_STATE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/ALARM_STATE",
    "type": {
      "measurement": {}
    }
  },
  {
    "name": "AWS/ALARM_SOURCE",
    "dataType": "STRING",
    "type": {
      "attribute": {
        "defaultValue": "alarmModelArn"
      }
    }
  }
]
}

```

- e. Exécutez la commande suivante pour mettre à jour le modèle d'actif avec la définition stockée dans le `update-asset-model-payload.json` fichier. Remplacez *asset-model-id* par l'ID du modèle d'actif.

```

aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://update-asset-model-payload.json

```

Votre modèle d'équipement définit désormais une alarme qui détecte AWS IoT Events. L'alarme surveille la propriété cible dans tous les actifs sur la base de ce modèle d'actif. Vous pouvez

configurer l'alarme sur chaque actif afin de personnaliser les propriétés telles que le seuil ou le destinataire du IAM Identity Center pour chaque actif. Pour de plus amples informations, veuillez consulter [Configurer les alarmes sur les actifs dans AWS IoT SiteWise](#).

Définissez les alarmes externes dans AWS IoT SiteWise

Les alarmes externes contiennent l'état d'une alarme que vous détectez à l'extérieur AWS IoT SiteWise.

Définir une alarme externe (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour définir une alarme externe sur un modèle d'actif existant. Pour définir une alarme externe sur un nouveau modèle d'actif, créez le modèle d'actif, puis procédez comme suit. Pour de plus amples informations, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

Pour définir une alarme sur un modèle d'actif

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Models (Modèles).
3. Choisissez le modèle d'actif pour lequel vous souhaitez définir une alarme.
4. Choisissez l'onglet Définitions des alarmes.
5. Choisissez Ajouter une alarme.
6. Dans les options de type d'alarme, choisissez Alarme externe.
7. Saisissez un nom pour votre alarme.
8. (Facultatif) Entrez une description pour votre alarme.
9. Choisissez Ajouter une alarme.

Définir une alarme externe (CLI)

Vous pouvez utiliser le AWS CLI pour définir une alarme externe sur un modèle d'actif nouveau ou existant.

Pour ajouter une alarme externe à un modèle d'actif, vous devez ajouter un modèle composite d'alarme au modèle d'actif. Un modèle composite d'alarme externe spécifie le EXTERNAL type et ne spécifie aucune propriété de source d'alarme. L'exemple d'alarme composite suivant définit une alarme de température externe.

```

{
  ...
  "assetModelCompositeModels": [
    {
      "name": "BoilerTemperatureHighAlarm",
      "type": "AWS/ALARM",
      "properties": [
        {
          "name": "AWS/ALARM_TYPE",
          "dataType": "STRING",
          "type": {
            "attribute": {
              "defaultValue": "EXTERNAL"
            }
          }
        },
        {
          "name": "AWS/ALARM_STATE",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/ALARM_STATE",
          "type": {
            "measurement": {}
          }
        }
      ]
    }
  ]
}

```

Pour plus d'informations sur la façon d'ajouter un modèle composite à un modèle d'actif nouveau ou existant, consultez ce qui suit :

- [Création d'un modèle d'actifs \(AWS CLI\)](#)
- [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#)

Après avoir défini l'alarme externe, vous pouvez intégrer l'état de l'alarme aux actifs en fonction du modèle d'actif. Pour de plus amples informations, veuillez consulter [Ingérer un état d'alarme externe dans AWS IoT SiteWise](#).

Configurer les alarmes sur les actifs dans AWS IoT SiteWise

Après avoir défini une AWS IoT Events alarme sur un modèle d'actif, vous pouvez configurer l'alarme pour chaque actif en fonction du modèle d'actif. Vous pouvez modifier la valeur du seuil et les paramètres de notification de l'alarme. Chacune de ces valeurs est un attribut de la ressource. Vous pouvez donc mettre à jour la valeur par défaut de l'attribut pour configurer ces valeurs.

Note

Vous pouvez configurer ces valeurs pour les AWS IoT Events alarmes, mais pas pour les alarmes externes.

Rubriques

- [Configuration d'une valeur de seuil \(console\)](#)
- [Configurer une valeur de seuil \(AWS CLI\)](#)
- [Configurez les paramètres de notification dans AWS IoT SiteWise](#)

Configuration d'une valeur de seuil (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour la valeur de l'attribut qui indique le seuil d'une alarme.

Pour mettre à jour la valeur seuil d'une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez l'actif pour lequel vous souhaitez mettre à jour une valeur de seuil d'alarme.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Recherchez l'attribut que l'alarme utilise pour sa valeur de seuil, puis entrez sa nouvelle valeur.
6. Choisissez Enregistrer.

Configurer une valeur de seuil (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour mettre à jour la valeur de l'attribut qui indique le seuil d'une alarme.

Vous devez connaître l'`assetId` de votre ressource et le `propertyId` de la propriété pour effectuer cette procédure. Vous pouvez également utiliser l'identifiant externe. Si vous avez créé un actif et que vous ne le connaissez pas `assetId`, utilisez l'[ListAssets](#) API pour répertorier tous les actifs d'un modèle spécifique. Utilisez cette [DescribeAsset](#) opération pour afficher les propriétés de votre actif, y compris les propriétés IDs.

Utilisez l'opération [BatchPutAssetPropertyValue](#) pour affecter des valeurs d'attributs à votre ressource. Vous pouvez utiliser cette opération pour définir plusieurs attributs à la fois. La charge utile de cette opération contient une liste d'entrées, chacune contenant l'ID de ressource, l'ID de propriété et la valeur d'attribut.

Pour mettre à jour la valeur d'un attribut (AWS CLI)

1. Créez un fichier nommé `batch-put-payload.json` et copiez l'objet JSON suivant dans le fichier. Cet exemple de charge utile montre comment définir la latitude et la longitude d'une éolienne. Mettez à jour IDs les valeurs et les horodatages pour modifier la charge utile en fonction de votre cas d'utilisation.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ],
  {
```

```

    "entryId": "windfarm3-turbine7-longitude",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-5555EXAMPLE",
    "propertyValues": [
      {
        "value": {
          "doubleValue": 122.3491
        },
        "timestamp": {
          "timeInSeconds": 1575691200
        }
      }
    ]
  }
}

```

- Chaque entrée de la charge utile contient un `entryId` que vous pouvez définir sous la forme d'une chaîne unique. Si des entrées de demande échouent, chaque erreur contiendra l'`entryId` de la demande correspondante afin que vous sachiez quelles demandes réessayer.
- Pour définir une valeur d'attribut, vous pouvez inclure une structure `timestamp-quality-value` (TQV) dans la liste de chaque propriété `propertyValues` d'attribut. Cette structure doit contenir le nouveau `value` et le `timestamp` actuel.
- `value`— Structure contenant l'un des champs suivants, en fonction du type de propriété défini :
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `timestamp`— Une structure qui contient l'heure actuelle d'Unix en secondes, `timeInSeconds`. AWS IoT SiteWise rejette tous les points de données dont l'horodatage existait depuis plus de 7 jours ou moins de 5 minutes dans le futur.

Pour de plus amples informations sur la préparation d'une charge utile pour [BatchPutAssetPropertyValue](#), veuillez consulter [Ingérez des données avec AWS IoT SiteWise APIs](#).

2. Exécutez la commande suivante pour envoyer les valeurs d'attribut à AWS IoT SiteWise :

```
aws iotsitewise batch-put-asset-property-value -\-cli-input-json file://batch-put-payload.json
```

Configurez les paramètres de notification dans AWS IoT SiteWise

Vous pouvez configurer les paramètres de notification d'alarme à l'aide de la AWS IoT SiteWise console ou du AWS Command Line Interface (AWS CLI).

Configuration des paramètres de notification (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour mettre à jour la valeur des attributs qui spécifient les paramètres de notification pour une alarme.

Pour mettre à jour les paramètres de notification d'une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez l'actif pour lequel vous souhaitez mettre à jour les paramètres d'alarme.
4. Choisissez Modifier.
5. Recherchez l'attribut utilisé par l'alarme pour le paramètre de notification que vous souhaitez modifier, puis entrez sa nouvelle valeur.
6. Choisissez Enregistrer.

Configuration des paramètres de notification (CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour mettre à jour la valeur de l'attribut qui spécifie les paramètres de notification pour une alarme.

Vous devez connaître l'`assetId` de votre ressource et le `propertyId` de la propriété pour effectuer cette procédure. Vous pouvez également utiliser l'identifiant externe. Si vous avez créé un actif et que vous ne le connaissez pas `assetId`, utilisez l'[ListAssets](#) API pour répertorier tous les actifs d'un modèle spécifique. Utilisez cette [DescribeAsset](#) opération pour afficher les propriétés de votre actif, y compris les propriétés IDs.

Utilisez l'opération [BatchPutAssetPropertyValue](#) pour affecter des valeurs d'attributs à votre ressource. Vous pouvez utiliser cette opération pour définir plusieurs attributs à la fois. La charge utile de cette opération contient une liste d'entrées, chacune contenant l'ID de ressource, l'ID de propriété et la valeur d'attribut.

Pour mettre à jour la valeur d'un attribut (AWS CLI)

1. Créez un fichier nommé `batch-put-payload.json` et copiez l'objet JSON suivant dans le fichier. Cet exemple de charge utile montre comment définir la latitude et la longitude d'une éolienne. Mettez à jour IDs les valeurs et les horodatages pour modifier la charge utile en fonction de votre cas d'utilisation.

```
{
  "entries": [
    {
      "entryId": "windfarm3-turbine7-latitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 47.6204
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    },
    {
      "entryId": "windfarm3-turbine7-longitude",
      "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 122.3491
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

```

    ]
  }
]
}

```

- Chaque entrée de la charge utile contient un `entryId` que vous pouvez définir sous la forme d'une chaîne unique. Si des entrées de demande échouent, chaque erreur contiendra l'`entryId` de la demande correspondante afin que vous sachiez quelles demandes réessayer.
- Pour définir une valeur d'attribut, vous pouvez inclure une structure `timestamp-quality-value` (TQV) dans la liste de chaque propriété `propertyValues` d'attribut. Cette structure doit contenir le nouveau `value` et le `timestamp` actuel.
- `value`— Structure contenant l'un des champs suivants, en fonction du type de propriété défini :
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`
- `timestamp`— Une structure qui contient l'heure actuelle d'Unix en secondes, `timeInSeconds`. AWS IoT SiteWise rejette tous les points de données dont l'horodatage existait depuis plus de 7 jours ou moins de 5 minutes dans le futur.

Pour de plus amples informations sur la préparation d'une charge utile pour [BatchPutAssetPropertyValue](#), veuillez consulter [Ingérez des données avec AWS IoT SiteWise APIs](#).

2. Exécutez la commande suivante pour envoyer les valeurs d'attribut à AWS IoT SiteWise :

```
aws iotsitewise batch-put-asset-property-value -\cli-input-json file://batch-put-payload.json
```

Répondez aux alarmes dans AWS IoT SiteWise

Lorsqu'une AWS IoT Events alarme change d'état, vous pouvez effectuer les opérations suivantes pour y répondre :

- Accusez réception d'une alarme pour indiquer que vous êtes en train de résoudre le problème.
- Suspendez une alarme pour la désactiver temporairement.
- Désactivez une alarme pour la désactiver définitivement jusqu'à ce que vous la réactiviez.
- Activez une alarme désactivée pour détecter l'état de l'alarme.
- Réinitialisez une alarme pour effacer son état et sa dernière valeur.

Vous pouvez utiliser la AWS IoT SiteWise console ou l' AWS IoT Events API pour répondre à une alarme.

 Note

Vous pouvez répondre aux AWS IoT Events alarmes, mais pas aux alarmes externes.

Rubriques

- [Répondre à une alarme \(console\)](#)
- [Répondre à une alarme \(API\)](#)

Répondre à une alarme (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour accuser réception, suspendre, désactiver ou activer une alarme.

Rubriques

- [Confirmer une alarme \(console\)](#)
- [Suspendez une alarme \(console\)](#)
- [Désactiver une alarme \(console\)](#)
- [Activer une alarme \(console\)](#)
- [Réinitialiser une alarme \(console\)](#)

Confirmer une alarme (console)

Vous pouvez accuser réception d'une alarme pour indiquer que vous êtes en train de résoudre le problème.

 Note

Vous devez activer le flux d'accusé de réception de l'alarme afin de pouvoir accuser réception de l'alarme. Cette option est activée par défaut si vous définissez l'alarme depuis la AWS IoT SiteWise console.

Pour accuser réception d'une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour laquelle vous souhaitez accuser réception d'une alarme.

 Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez l'onglet Alarmes.
5. Sélectionnez l'alarme pour accuser réception, puis choisissez Actions pour ouvrir le menu des actions de réponse.
6. Sélectionnez I acknowledge (Je confirme). L'état de l'alarme passe à Accusé.

Suspendez une alarme (console)

Vous pouvez suspendre une alarme pour la désactiver temporairement. Spécifiez la durée pendant laquelle l'alarme doit être interrompue.

Pour suspendre une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour laquelle vous souhaitez suspendre une alarme.

 Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez l'onglet Alarmes.
5. Sélectionnez l'alarme à suspendre, puis choisissez Actions pour ouvrir le menu des actions de réponse.
6. Choisissez Snooze. Un modèle s'ouvre dans lequel vous spécifiez la durée de la pause.
7. Choisissez la durée du rappel ou entrez une durée de rappel personnalisée.
8. Choisissez Enregistrer. L'état de l'alarme passe à Snoozed.

Désactiver une alarme (console)

Vous pouvez désactiver une alarme pour qu'elle ne soit plus détectée. Après avoir désactivé l'alarme, vous devez la réactiver si vous souhaitez qu'elle soit détectée.

Pour désactiver une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez l'actif pour lequel vous souhaitez désactiver une alarme.

 Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez l'onglet Alarmes.
5. Sélectionnez l'alarme à désactiver, puis choisissez Actions pour ouvrir le menu des actions de réponse.
6. Choisissez Désactiver. L'état de l'alarme passe à Désactivé.

Activer une alarme (console)

Vous pouvez activer une alarme pour qu'elle soit à nouveau détectée après l'avoir désactivée ou mise en pause.

Pour activer une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour laquelle vous souhaitez activer une alarme.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez l'onglet Alarmes.
5. Sélectionnez l'alarme à activer, puis choisissez Actions pour ouvrir le menu des actions de réponse.
6. Sélectionnez Activer. L'état de l'alarme passe à Normal.

Réinitialiser une alarme (console)

Vous pouvez réinitialiser une alarme pour effacer son état et sa dernière valeur.

Pour réinitialiser une alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez l'actif pour lequel vous souhaitez réinitialiser une alarme.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez l'onglet Alarmes.

5. Sélectionnez l'alarme à activer, puis choisissez Actions pour ouvrir le menu des actions de réponse.
6. Choisissez Réinitialiser. L'état de l'alarme passe à Normal.

Répondre à une alarme (API)

Vous pouvez utiliser l' AWS IoT Events API pour accuser réception, suspendre, désactiver, activer ou réinitialiser une alarme. Pour plus d'informations, consultez les opérations suivantes dans la référence AWS IoT Events d'API :

- [BatchAcknowledgeAlarm](#)
- [BatchSnoozeAlarm](#)
- [BatchDisableAlarm](#)
- [BatchEnableAlarm](#)
- [BatchResetAlarm](#)

Pour plus d'informations, consultez la section [Répondre aux alarmes](#) dans le guide du AWS IoT Events développeur.

Ingérer un état d'alarme externe dans AWS IoT SiteWise

Les alarmes externes sont des alarmes que vous évaluez en dehors de celles-ci AWS IoT SiteWise. Vous pouvez utiliser des alarmes externes lorsqu'une source de données indique l'état des alarmes que vous souhaitez ingérer AWS IoT SiteWise.

Les propriétés d'état d'alarme nécessitent un format spécifique pour les valeurs des données d'état d'alarme. Chaque valeur de données doit être un objet JSON sérialisé en chaîne. Ensuite, vous ingérez la chaîne sérialisée sous forme de valeur de chaîne. Pour de plus amples informations, veuillez consulter [Propriétés de l'état de l'alarme](#).

Exemple Exemple de valeur de données d'état d'alarme (non sérialisée)

```
{
  "stateName": "Active"
}
```

Exemple Exemple de valeur de données d'état d'alarme (sérialisée)

```
{\"stateName\": \"Active\"}
```

Note

Si votre source de données ne peut pas fournir de données dans ce format, ou si vous ne pouvez pas convertir vos données dans ce format avant de les ingérer, vous pouvez choisir de ne pas utiliser de propriété d'alarme. Au lieu de cela, vous pouvez ingérer les données en tant que propriété de mesure avec le type de données chaîne, par exemple. Pour plus d'informations, consultez [Définir les flux de données provenant de l'équipement \(mesures\)](#) et [Ingérez des données dans AWS IoT SiteWise](#).

Cartographiez les flux d'état des alarmes externes dans AWS IoT SiteWise

Vous pouvez définir des alias de propriété pour mapper vos flux de données aux propriétés de l'état de votre alarme. Cela vous permet d'identifier facilement une propriété d'état d'alarme lorsque vous ingérez ou récupérez des données. Pour plus d'informations sur les alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Rubriques

- [Cartographier les flux d'état des alarmes externes \(console\)](#)
- [Cartographier les flux d'état des alarmes externes \(AWS CLI\)](#)

Cartographier les flux d'état des alarmes externes (console)

Vous pouvez définir des alias de propriété pour mapper vos flux de données aux propriétés de l'état de votre alarme. Cela vous permet d'identifier facilement une propriété d'état d'alarme lorsque vous ingérez ou récupérez des données. Pour plus d'informations sur les alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Vous pouvez utiliser la AWS IoT SiteWise console pour définir un alias pour une propriété d'état d'alarme.

Pour définir un alias de propriété pour une propriété d'état d'alarme (console)

1. Accédez à la [console AWS IoT SiteWise](#).

2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour laquelle vous souhaitez définir un alias de propriété.

 Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Faites défiler la page jusqu'à Alarmes et développez la section.
6. Sous Alarmes externes, entrez l'alias dans Alias de propriété — facultatif.
7. Choisissez Enregistrer.

Cartographier les flux d'état des alarmes externes (AWS CLI)

Vous pouvez définir des alias de propriété pour mapper vos flux de données aux propriétés de l'état de votre alarme. Cela vous permet d'identifier facilement une propriété d'état d'alarme lorsque vous ingérez ou récupérez des données. Pour plus d'informations sur les alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour définir un alias pour une propriété d'état d'alarme.

Vous devez connaître l'assetId de votre ressource et le propertyId de la propriété pour effectuer cette procédure. Vous pouvez également utiliser l'identifiant externe. Si vous avez créé un actif et que vous ne le connaissez pas assetId, utilisez l'[ListAssets](#) API pour répertorier tous les actifs d'un modèle spécifique. Utilisez cette [DescribeAsset](#) opération pour afficher les propriétés de votre actif, y compris les propriétés IDs.

 Note

La [DescribeAsset](#) réponse inclut la liste des modèles d'actifs composites pour l'actif. Chaque alarme est un modèle composite. Pour trouver le propertyId, recherchez le modèle composite de l'alarme, puis recherchez la AWS/ALARM_STATE propriété dans ce modèle composite.

Pour plus d'informations sur la définition de l'alias de propriété, consultez [Mettre à jour un alias de propriété d'actif](#).

Ingérer les données d'état des alarmes dans AWS IoT SiteWise

Les propriétés d'état d'alarme prévoient un état d'alarme sous forme de chaîne JSON sérialisée. Pour intégrer l'état d'alarme à une alarme externe AWS IoT SiteWise, vous devez ingérer cette chaîne sérialisée sous forme de valeur de chaîne horodatée. L'exemple suivant montre une valeur de données d'état pour une alarme active.

```
{"stateName\":\"Active\"}
```

Pour identifier une propriété d'état d'alarme, vous pouvez spécifier l'une des options suivantes :

- La `assetId` fin `propertyId` de la propriété d'alarme à laquelle vous envoyez des données.
- Le `propertyAlias`, qui est un alias de flux de données (par exemple, `/company/windfarm/3/turbine/7/temperature/high`). Pour utiliser cette option, vous devez d'abord définir l'alias de la propriété de votre alarme. Pour savoir comment définir des alias de propriété pour les propriétés d'état des alarmes, consultez [Cartographiez les flux d'état des alarmes externes dans AWS IoT SiteWise](#).

L'exemple de charge utile de [BatchPutAssetPropertyValue](#) l'API suivant montre comment formater l'état d'une alarme externe. Cette alarme externe signale lorsque le nombre de rotations par minute (RPM) d'une éolienne est trop élevé.

Exemple Exemple de BatchPutAssetPropertyValue charge utile pour les données d'état des alarmes

```
{
  "entries": [
    {
      "entryId": "unique entry ID",
      "propertyAlias": "/company/windfarm/3/turbine/7/temperature/high",
      "propertyValues": [
        {
          "value": {
            "stringValue": "{\"stateName\":\"Active\"}"
          },
          "timestamp": {
            "timeInSeconds": 1607550262
          }
        }
      ]
    }
  ]
}
```

```
}  
  }  
    ]  
      }  
        ]  
          }  
            ]  
              }
```

Pour plus d'informations sur l'utilisation de l'`BatchPutAssetPropertyValueAPI` pour ingérer des données, consultez [Ingérez des données avec AWS IoT SiteWise APIs](#).

Pour plus d'informations sur les autres méthodes d'ingestion de données, consultez [Ingérez des données dans AWS IoT SiteWise](#).

AWS IoT SiteWise Assistante

L' AWS IoT SiteWise assistant est un assistant génératif alimenté par l'IA. Il permet aux utilisateurs tels que les directeurs d'usine, les ingénieurs qualité et les techniciens de maintenance d'obtenir des informations, de résoudre des problèmes et de prendre des mesures directement à partir de leurs données opérationnelles et d'entreprise.

L' AWS IoT SiteWise assistant consolide les informations issues AWS IoT des données, des modèles d'actifs, des manuels et de la documentation dans des résumés compréhensibles des événements critiques. Il permet également des sessions interactives de questions-réponses approfondies pour faciliter le diagnostic, l'exploration des causes profondes et les recommandations guidées.

Rubriques

- [Configuration de l' AWS IoT SiteWise assistant](#)
- [Créer un jeu de données](#)
- [Modifier un jeu de données](#)
- [Supprimer un jeu de données](#)
- [AWS IoT SiteWise Questions de l'assistant](#)

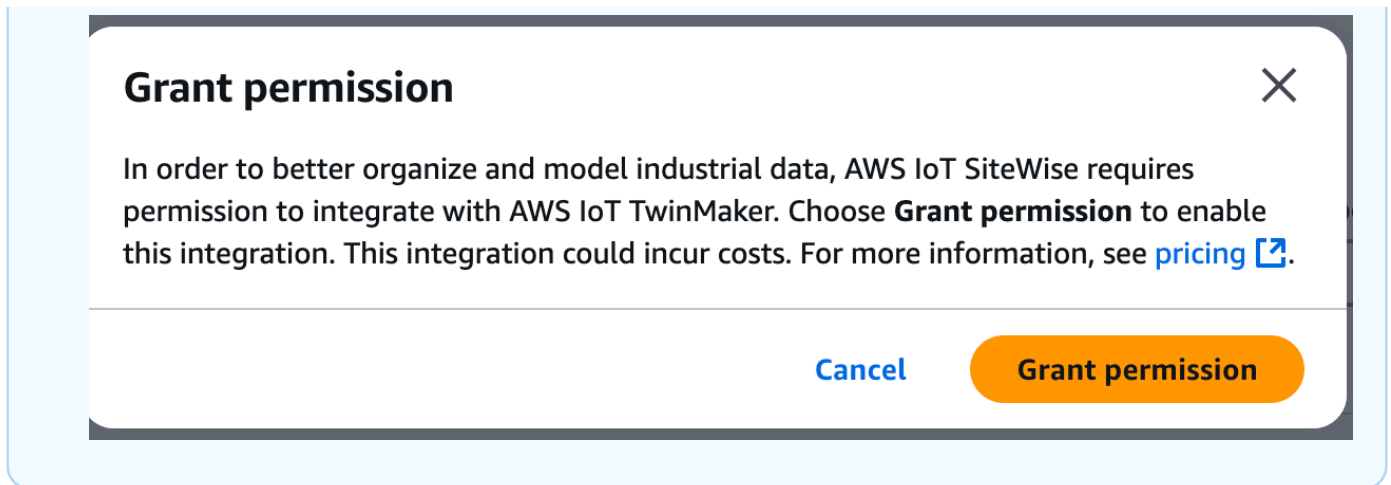
Configuration de l' AWS IoT SiteWise assistant

AWS IoT SiteWise Configuration de l'assistant

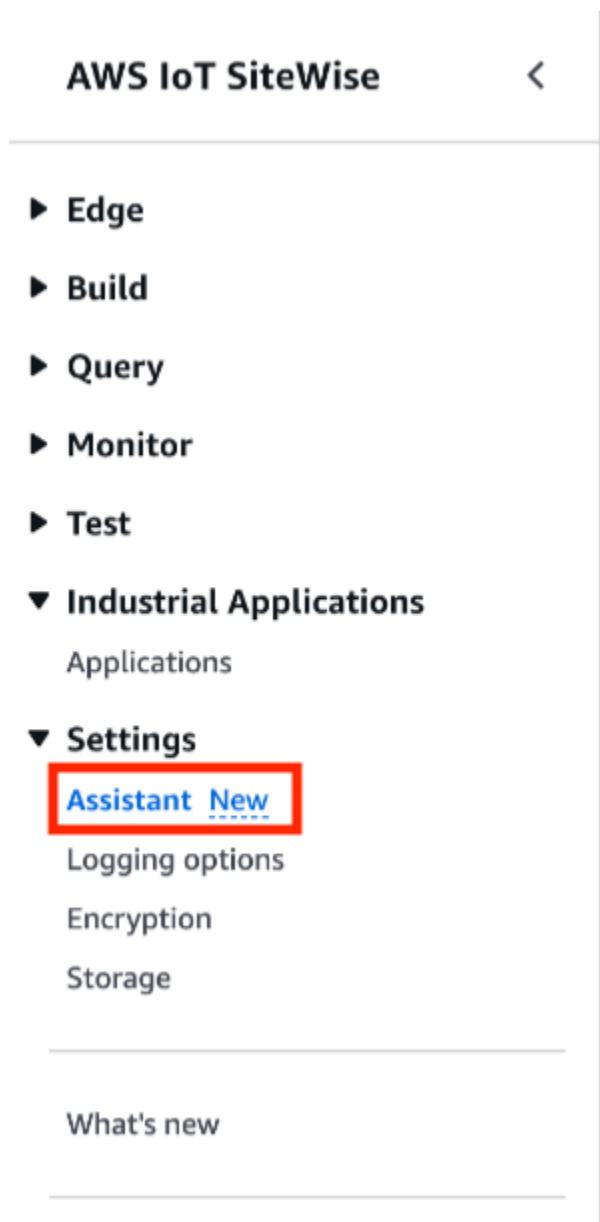
1. Connectez-vous à la [console AWS IoT SiteWise](#).

Note

Accordez des autorisations pour permettre l'intégration au AWS IoT TwinMaker service. Cela est nécessaire pour que l' AWS IoT SiteWise assistant et le tableau de bord puissent exécuter des requêtes SQL dans les AWS IoT SiteWise ressources. Voir [Intégrer AWS IoT SiteWise et AWS IoT TwinMaker](#).



2. Choisissez Assistant dans le panneau de navigation de gauche.



Créer un jeu de données

Note

L' AWS IoT SiteWise assistant doit utiliser un ensemble de données avec un index [Amazon Kendra](#) pour obtenir des connaissances et des conseils au niveau de l'entreprise. Si vous ne possédez pas d'index Amazon Kendra, consultez [Création d'un index pour en créer un](#). L'ajout d'un [jeu de données](#) améliore la qualité de réponse de l'assistant et minimise les hallucinations.

Console

Création d'un jeu de données dans la AWS IoT SiteWise console

1. Les ensembles de données sont affichés dans la section Ensembles de données de la page AWS IoT SiteWise Assistant.
2. S'il n'existe aucun jeu de données, choisissez Create dataset.
3. Sur la page des détails du jeu de données, choisissez un index Kendra dans le menu déroulant à associer à l'ensemble de données.
4. Le nom du jeu de données est renseigné par l'index Kendra sélectionné à l'étape 3. Modifiez le nom si nécessaire.
5. (Facultatif) La description du jeu de données est renseignée par l'indice Kendra sélectionné à l'étape 3. Modifiez la description si nécessaire.
6. Dans la section Autorisations, choisissez l'une des options ci-dessous :
 - a. Choisissez Créer et utiliser un nouveau rôle de service. Par défaut, crée AWS IoT SiteWise automatiquement un rôle de service. Ce rôle permet à l' AWS IoT SiteWise Assistant d'accéder à vos index Kendra.
 - b. Choisissez Utiliser un rôle de service existant, puis choisissez le rôle cible.
7. Sélectionnez Create (Créer).

[AWS IoT SiteWise](#) > [Assistant](#) > Create dataset

Create dataset

Create a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#).

Choose a method to authorize Assistant

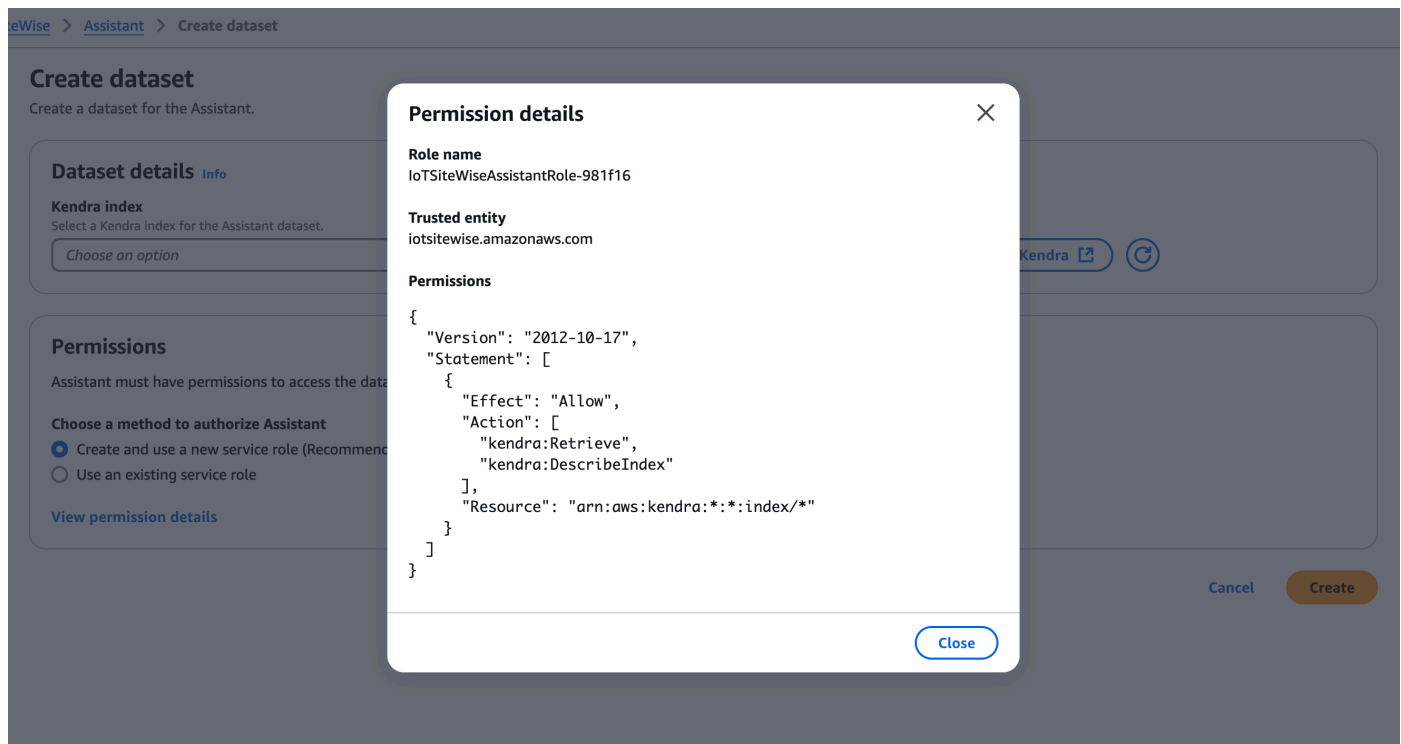
- ☒ Create and use a new service role - *recommended*
- ☐ Use an existing service role

[View permission details](#)

[Cancel](#)

Create

Le rôle de service créé par AWS IoT SiteWise pour l'utilisateur, si celui-ci a choisi de créer et d'utiliser un nouveau rôle de service.



AWS CLI

Créez un jeu de données dans AWS CLI

1. Créez un rôle IAM utilisé pour créer un ensemble de données. Utilisez la politique d'autorisation suivante :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kendra:Retrieve"
      ],
      "Resource": "arn:aws:kendra:*:*:index/*"
    }
  ]
}
```

Utilisez la relation de confiance suivante :

```
{
  "Version": "2008-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Créez un fichier `create-dataset.json` avec le modèle fourni dans l'exemple. Renseignez `datasetId` `kendra` `knowledgeBaseArn` et connectez-vous `roleArn` à cet ensemble de données.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

3. Créez l'ensemble de données à l'aide de la commande suivante :

```
aws iotsitewise create-dataset --cli-input-json file://create-dataset.json --
region us-east-1
```

Modifier un jeu de données

Console

Modifier un jeu de données

1. Les ensembles de données sont affichés dans la section Ensembles de données de la page Assistant. Choisissez un jeu de données à modifier. Choisissez Modifier pour commencer à modifier.
2. Sur la page des détails du jeu de données, choisissez un index Kendra dans le menu déroulant à associer à l'ensemble de données.
3. Le nom du jeu de données est renseigné par l'index Kendra sélectionné à l'étape 2. Modifiez le nom si nécessaire.
4. (Facultatif) La description du jeu de données est renseignée par l'index Kendra sélectionné à l'étape 2. Modifiez la description si nécessaire.
5. Dans la section Autorisations, choisissez l'une des options ci-dessous :
 - a. Choisissez Créer et utiliser un nouveau rôle de service. Par défaut, crée AWS IoT SiteWise automatiquement un rôle de service. Ce rôle permet à l' AWS IoT SiteWise Assistant d'accéder à vos index Kendra.
 - b. Choisissez Utiliser un rôle de service existant, puis choisissez le rôle cible.
6. Choisissez Enregistrer les modifications pour enregistrer votre sélection.

[AWS IoT SiteWise](#) > [Assistant](#) > Edit dataset

Edit dataset

Edit a dataset for the Assistant.

Dataset details [Info](#)

Kendra index

Select a Kendra index for the Assistant dataset.

test-index

[Amazon Kendra](#)



Dataset name

Dataset name is pre-populated from the Kendra index selected. You can modify the dataset name.

Dataset1

Dataset name must be 1-256 characters. Valid characters are a-z, A-Z, 0-9, _ (underscore) and - (hyphen).

Dataset description - optional

Dataset description is pre-populated from the Kendra index selected. You can modify the dataset description.

Dataset for AWS IoT SiteWise Assistant

Dataset description must be 1-2048 characters.

Permissions

Assistant must have permissions to access the data. To create a custom role, visit the [IAM console](#).

Choose a method to authorize Assistant

- ☐ Create and use a new service role - *recommended*
- ☒ Use an existing service role

Existing role

IoTSiteWiseAssistantRole-40f994

[View the IoTSiteWiseAssistantRole-40f994 role in the IAM console](#)

[Cancel](#)

[Save changes](#)

AWS CLI

Modifier un jeu de données dans AWS CLI

1. Créez un fichier update-dataset.json avec le modèle fourni dans l'exemple. Renseignez datasetId kendra knowledgeBaseArn et connectez-vous roleArn à cet ensemble de données.

```
{
  "datasetId": "<UUID>",
  "datasetName": "DatasetForAssistant",
  "datasetSource": {
    "sourceType": "KENDRA",
    "sourceFormat": "KNOWLEDGE_BASE",
    "sourceDetail": {
      "kendra": {
        "knowledgeBaseArn": "arn:aws:kendra::%s:index/index",
```



```
        "roleArn": "arn:aws:iam::%s:role/role"
      }
    }
  }
}
```

2. Mettez à jour l'ensemble de données à l'aide de la commande suivante :

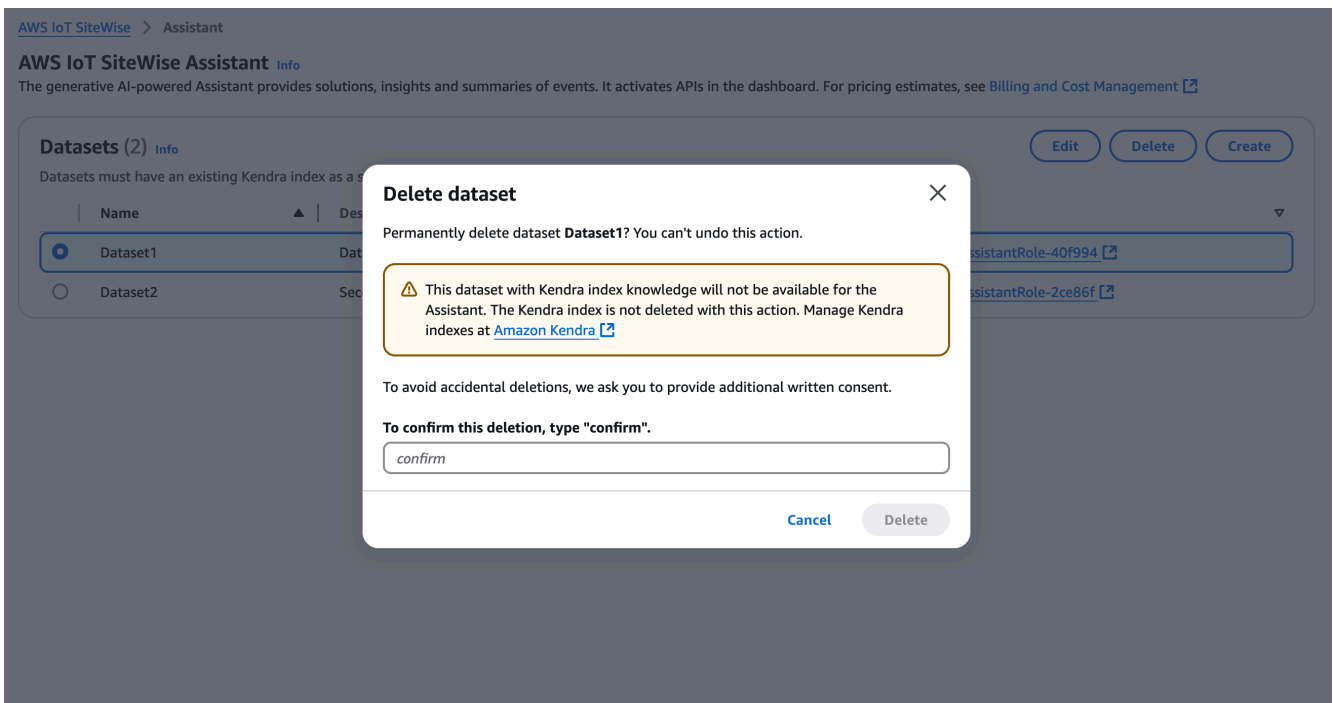
```
aws iotsitewise update-dataset --cli-input-json file://update-dataset.json --  
region us-east-1
```

Supprimer un jeu de données

Console

Supprimer un jeu de données

1. Les ensembles de données sont affichés dans la section Ensembles de données de la page Assistant. Choisissez un ensemble de données. Sélectionnez Delete (Supprimer).
2. Tapez confirm dans la fenêtre contextuelle pour confirmer la suppression.



3. Sélectionnez Delete (Supprimer).

AWS CLI

Supprimer un jeu de données

- Supprimez le jeu de données avec `datasetId`.

```
aws iotsitewise delete-dataset --region us-east-1 --dataset-id <UUID>
```

AWS IoT SiteWise Questions de l'assistant

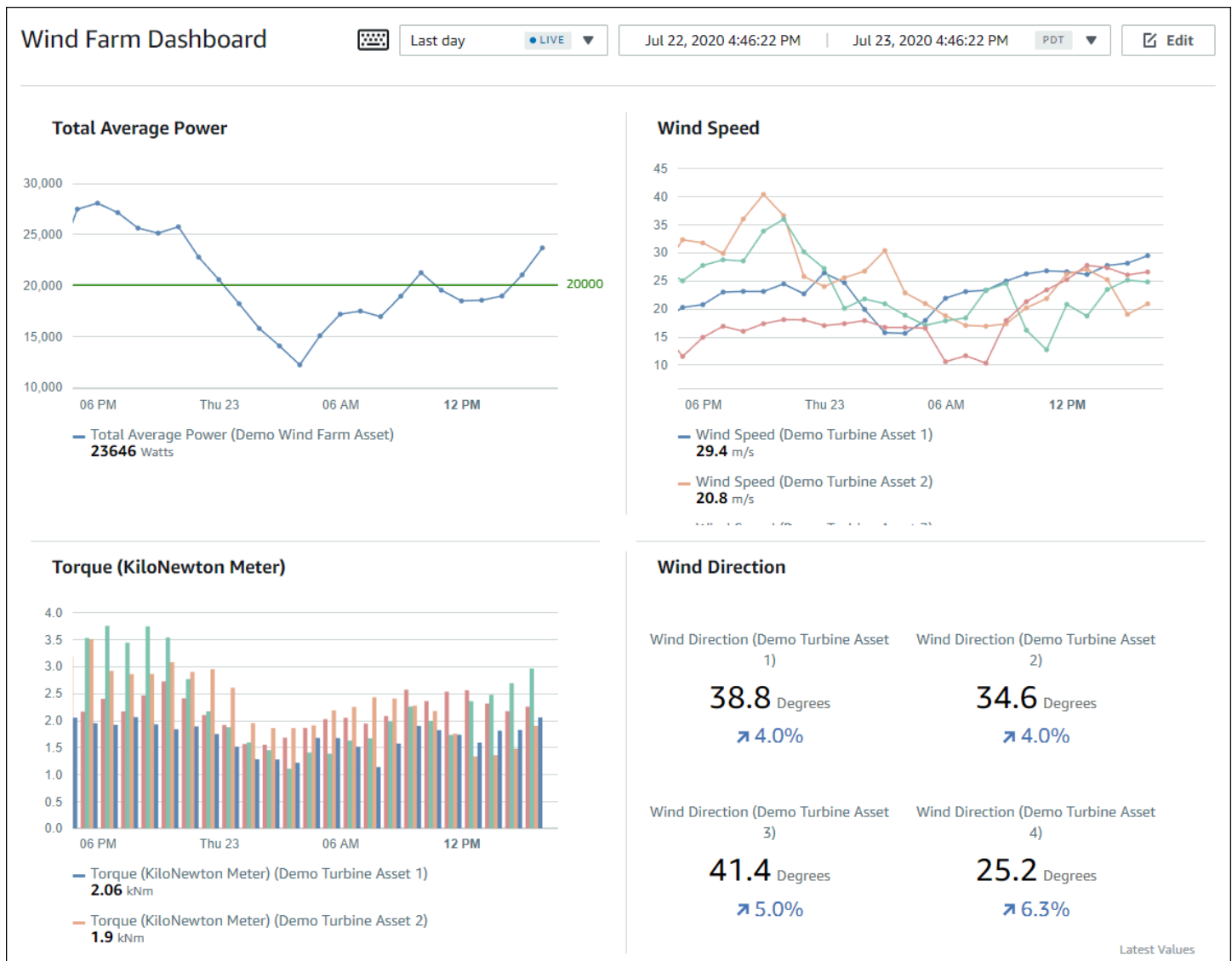
Consultez [Exemples de questions à poser à l' AWS IoT SiteWise assistant](#) pour en savoir plus sur l'interrogation de l' AWS IoT SiteWise Assistant.

Surveillez les données avec AWS IoT SiteWise Monitor

Vous pouvez l'utiliser AWS IoT SiteWise pour surveiller les données de vos processus, appareils et équipements en créant des portails Web SiteWise Monitor. SiteWise Monitor est une fonctionnalité AWS IoT SiteWise que vous pouvez utiliser pour créer des portails sous la forme d'une application Web gérée. Vous pouvez ensuite utiliser ces portails pour afficher et partager vos données opérationnelles. Vous pouvez créer des projets avec des tableaux de bord pour visualiser les données de vos processus, périphériques et équipements connectés à AWS IoT.

Les experts de domaine, tels que les ingénieurs de processus, peuvent utiliser ces portails pour obtenir rapidement des informations sur leurs données opérationnelles afin de comprendre le comportement des appareils et des équipements.

Voici un exemple de tableau de bord qui affiche les données d'un parc éolien.



Comme il AWS IoT SiteWise capture les données au fil du temps, vous pouvez utiliser SiteWise Monitor pour visualiser les données opérationnelles au fil du temps ou les dernières valeurs signalées à des moments spécifiques. Cela vous permet de découvrir des informations qui sont généralement difficiles à trouver.

SiteWise Contrôler les rôles

Quatre rôles interagissent avec SiteWise Monitor :

AWS administrateur

L' AWS administrateur utilise la AWS IoT SiteWise console pour créer des portails.

L'administrateur AWS peut également affecter des administrateurs de portail et ajouter des

utilisateurs de portail. Les administrateurs du portail attribueront ultérieurement des utilisateurs du portail à des projets en tant que propriétaires ou utilisateurs standard. L' AWS administrateur travaille exclusivement dans la AWS console.

Administrateur du portail

Chaque portail SiteWise Monitor possède un ou plusieurs administrateurs de portail. Les administrateurs du portail utilisent ce dernier pour créer des projets contenant des collections de ressources et de tableaux de bord. Ils attribuent ensuite des ressources et des propriétaires à chaque projet. En contrôlant l'accès au projet, les administrateurs de portail spécifient les ressources que les propriétaires et les utilisateurs de projet peuvent voir.

Propriétaire du projet

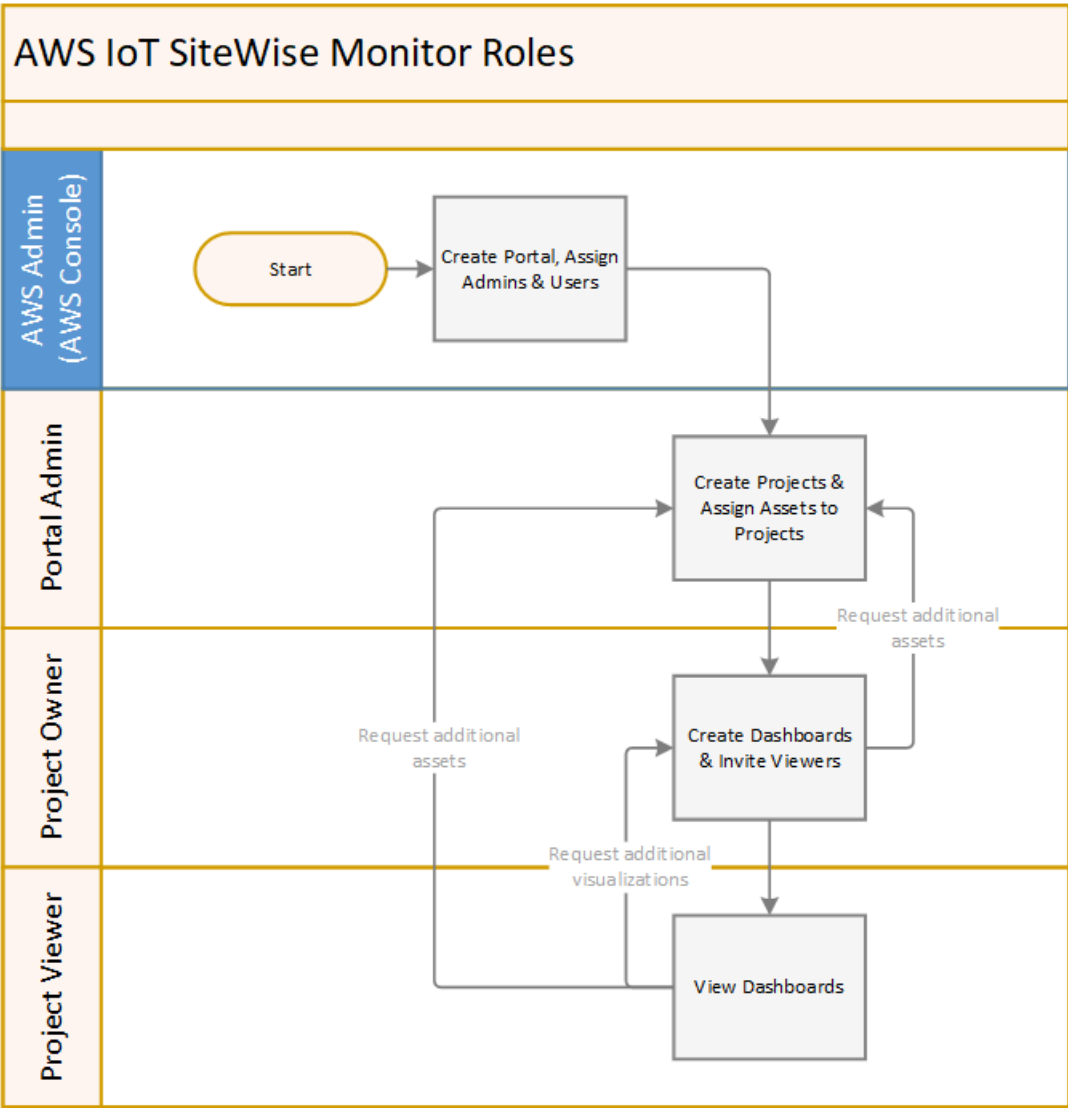
Chaque projet SiteWise Monitor a des propriétaires. Ces propriétaires créent des visualisations sous la forme de tableaux de bord afin de représenter les données opérationnelles de manière cohérente. Lorsque les tableaux de bord sont prêts à être partagés, tout propriétaire du projet peut inviter des utilisateurs. Les propriétaires de projet peuvent également affecter d'autres propriétaires au projet. Les propriétaires de projets peuvent configurer des seuils et des paramètres de notification pour les alarmes.

Utilisateur de projet

Chaque projet SiteWise Monitor possède des spectateurs. Les utilisateurs de projet peuvent se connecter au portail pour consulter les tableaux de bord créés par les propriétaires de projet. Dans chaque tableau de bord, les utilisateurs du projet peuvent ajuster la plage de temps pour mieux comprendre les données opérationnelles. Les utilisateurs de projet ne peuvent afficher que les tableaux de bord des projets auxquels ils ont accès. Les spectateurs du projet peuvent accuser réception des alarmes et les suspendre.

Selon votre organisation, la même personne peut jouer plusieurs rôles.

L'image suivante montre comment ces quatre rôles interagissent dans le portail SiteWise Monitor.



Vous pouvez gérer qui a accès à vos données en utilisant AWS IAM Identity Center ou IAM. Les utilisateurs de vos données peuvent se connecter à SiteWise Monitor depuis un navigateur de bureau ou mobile à l'aide de leurs informations d'identification IAM Identity Center ou IAM.

Fédération SAML

IAM Identity Center et IAM prennent en charge la fédération des identités avec le langage [SAML \(Security Assertion Markup Language\)](#) 2.0. SAML 2.0 est un standard ouvert que de nombreux fournisseurs d'identité externes (IdPs) utilisent pour authentifier les utilisateurs et transmettre leur identité et leurs informations de sécurité aux fournisseurs de services (SPs). SPs sont généralement des applications ou des services. La fédération SAML permet aux administrateurs et aux utilisateurs de votre portail SiteWise Monitor de se connecter aux portails qui leur sont assignés à l'aide

d'informations d'identification externes, telles que le nom d'utilisateur et le mot de passe de leur entreprise.

Vous pouvez configurer IAM Identity Center et IAM pour utiliser la fédération basée sur SAML pour accéder à vos portails Monitor. SiteWise

IAM Identity Center

Les administrateurs et les utilisateurs de votre portail peuvent se connecter au portail d' AWS accès à l'aide de leur nom d'utilisateur et de leur mot de passe d'entreprise. Ils peuvent ensuite accéder aux portails de SiteWise surveillance qui leur sont assignés. IAM Identity Center utilise des certificats pour établir une relation de confiance SAML entre votre fournisseur d'identité et. AWS Pour plus d'informations, le [profil SCIM et l'implémentation de SAML 2.0](#) sont disponibles dans le guide de l'AWS IAM Identity Center utilisateur.

IAM

Les administrateurs et utilisateurs de votre portail peuvent demander des informations d'identification de sécurité temporaires pour accéder aux portails SiteWise Monitor qui leur sont assignés. Vous créez une identité de fournisseur d'identité SAML dans IAM pour établir une relation de confiance entre votre fournisseur d'identité et. AWS Pour plus d'informations, consultez la section [Utilisation de la fédération basée sur SAML pour l'accès à l'API](#) dans le AWS guide de l'utilisateur IAM.

Les administrateurs et utilisateurs de votre portail peuvent se connecter au portail de votre entreprise et sélectionner l'option permettant d'accéder à la console AWS de gestion. Ils peuvent ensuite accéder aux portails de SiteWise surveillance qui leur sont assignés. Le portail de votre entreprise gère l'échange de confiance entre votre fournisseur d'identité et AWS. Pour plus d'informations, consultez la section [Permettre aux utilisateurs fédérés SAML 2.0 d'accéder à la console de AWS gestion dans le](#) guide de l'utilisateur IAM.

Note

Lorsque vous ajoutez des utilisateurs ou des administrateurs au portail, évitez de créer des politiques IAM qui limitent les autorisations des utilisateurs, telles qu'une adresse IP limitée. Les politiques associées avec des autorisations restreintes ne pourront pas se connecter au AWS IoT SiteWise portail.

SiteWise Concepts de surveillance

Pour utiliser SiteWise Monitor, vous devez connaître les concepts suivants :

Portal

Un AWS IoT SiteWise Monitor portail est une application Web que vous pouvez utiliser pour visualiser et partager vos AWS IoT SiteWise données. Un portail compte un ou plusieurs administrateurs et contient zéro ou plusieurs projets.

Projet

Chaque portail SiteWise Monitor contient un ensemble de projets. Chaque projet est associé à un sous-ensemble de vos ressources AWS IoT SiteWise . Les propriétaires de projet créent un ou plusieurs tableaux de bord pour fournir un moyen cohérent de visualiser les données liées à ces ressources. Ils peuvent inviter des utilisateurs standard dans le projet pour leur permettre d'en consulter les ressources et les tableaux de bord. Le projet est l'unité de base du partage au sein de SiteWise Monitor. Les propriétaires de projets peuvent inviter les utilisateurs auxquels l' AWS administrateur a donné accès au portail. Tout utilisateur doit avoir accès à un portail avant qu'un projet de ce portail puisse être partagé avec lui.

Ressource

Lorsque des données sont ingérées AWS IoT SiteWise depuis votre équipement industriel, vos appareils, équipements et processus sont chacun représentés comme des actifs. Chaque actif possède des propriétés et des alarmes qui lui sont associées. L'administrateur du portail affecte des ensembles de ressources à chaque projet.

Propriété

Les propriétés sont des séries chronologiques associées à des actifs. Par exemple, une pièce d'équipement peut avoir un numéro de série, un emplacement, une marque et un modèle, ainsi qu'une date d'installation. Elle peut également comporter des valeurs de séries chronologiques pour la disponibilité, les performances, la qualité, la température, la pression, etc.

alerte

Les alarmes surveillent les propriétés pour identifier lorsque l'équipement se trouve en dehors de sa plage de fonctionnement. Chaque alarme définit un seuil et une propriété à surveiller. Lorsque la propriété dépasse le seuil, l'alarme s'active et indique que vous ou un membre de votre équipe devez régler le problème. Les propriétaires de projets peuvent personnaliser les seuils

et les paramètres de notification pour les alarmes. Les spectateurs du projet peuvent accuser réception des alarmes et les suspendre, et ils peuvent laisser un message contenant des détails sur l'alarme ou sur les mesures prises pour y remédier.

Tableau de bord

Chaque projet contient un ensemble de tableaux de bord. Les tableaux de bord fournissent un ensemble de visualisations pour les valeurs d'un ensemble de ressources. Les propriétaires de projet créent les tableaux de bord et les visualisations qu'il contient. Lorsqu'un propriétaire de projet est prêt à partager l'ensemble de tableaux de bord, il peut inviter des utilisateurs dans le projet, ce qui leur donne accès à tous les tableaux de bord correspondants. Si vous souhaitez affecter différents groupes d'utilisateurs à différents tableaux de bord, vous devez diviser les tableaux de bord entre plusieurs projets. Lorsque les utilisateurs consultent les tableaux de bord, ils peuvent personnaliser la plage horaire pour examiner des données spécifiques.

Visualisation

Dans chaque tableau de bord, les propriétaires de projet décident comment afficher les propriétés et les alarmes des actifs associés au projet. La disponibilité peut être représentée sous forme de graphique linéaire, tandis que d'autres valeurs peuvent être affichées sous forme de diagrammes à barres ou d'indicateurs de performance clés (KPIs). Il est préférable d'afficher les alarmes sous forme de grilles d'état et de chronologies d'état. Les propriétaires de projet personnalisent chaque visualisation pour fournir une compréhension optimale des données de la ressource concernée.

Commencez avec AWS IoT SiteWise Monitor (Classic)

Si vous êtes l' AWS administrateur de votre organisation, vous créez des portails à partir de la AWS IoT SiteWise console. Procédez comme suit pour créer un portail afin que les membres de votre organisation puissent consulter vos AWS IoT SiteWise données :

1. Configurer et créer un portail
2. Ajouter des administrateurs de portail et envoyer des e-mails d'invitation
3. Ajouter des utilisateurs de portail

Après avoir créé un portail, l'administrateur du portail peut consulter vos AWS IoT SiteWise actifs et les affecter à des projets sur le portail. Les propriétaires de projet peuvent ensuite créer des tableaux de bord pour visualiser les propriétés des ressources qui aident les utilisateurs de projet à comprendre les performances de vos appareils, processus et équipement.

 Note

Lorsque vous ajoutez des utilisateurs ou des administrateurs au portail, évitez de créer des politiques AWS Identity and Access Management (IAM) qui limitent les autorisations des utilisateurs, telles qu'une adresse IP limitée. Les politiques associées avec des autorisations restreintes ne pourront pas se connecter au AWS IoT SiteWise portail.

Vous pouvez suivre un didacticiel qui décrit les étapes requises pour configurer un portail avec un projet, des tableaux de bord et plusieurs utilisateurs pour un scénario spécifique à l'aide de données de parc éolien. Pour de plus amples informations, veuillez consulter [Visualisez et partagez les données des parcs éoliens dans SiteWise Monitor](#).

Rubriques

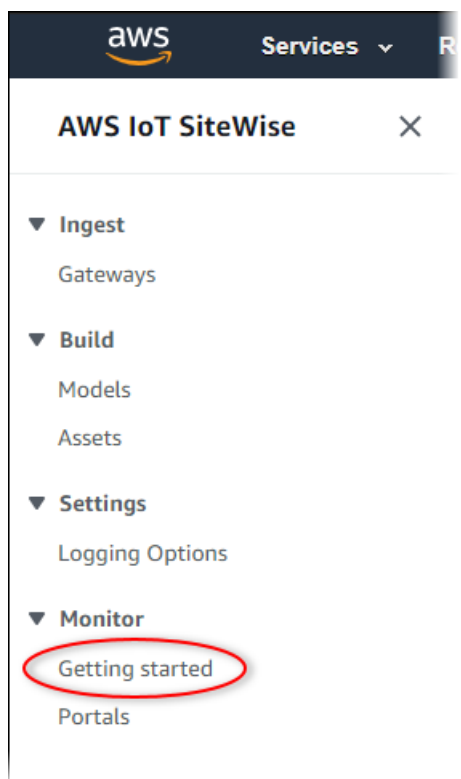
- [Création d'un portail dans SiteWise Monitor](#)
- [Configurez votre portail dans SiteWise Monitor](#)
- [Inviter des administrateurs dans SiteWise Monitor](#)
- [Ajouter des utilisateurs du portail dans SiteWise Monitor](#)
- [Création de AWS IoT SiteWise tableaux de bord \(\)AWS CLI](#)
- [Activez les alarmes pour vos portails dans AWS IoT SiteWise](#)
- [Activation de votre AWS IoT SiteWise portail à la périphérie](#)
- [Administrez vos portails SiteWise Monitor](#)

Création d'un portail dans SiteWise Monitor

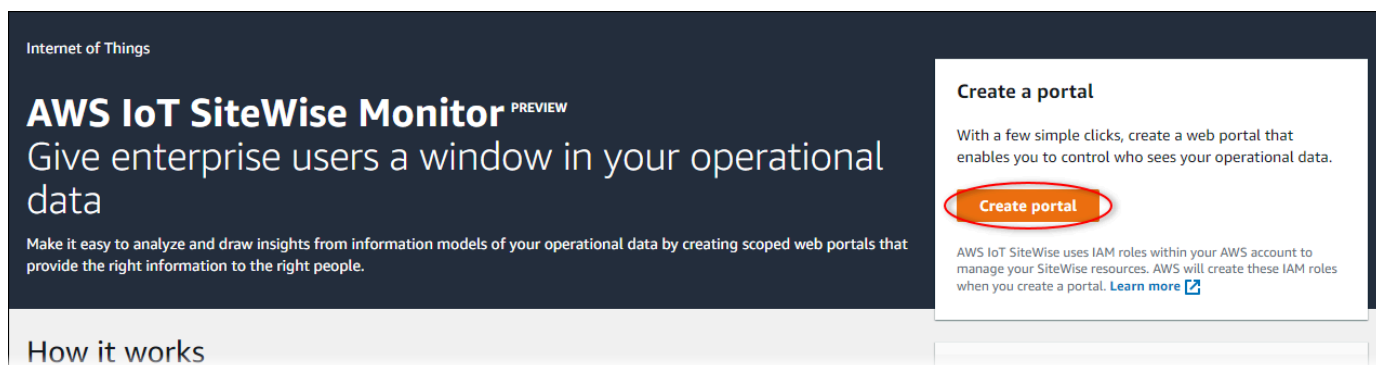
Vous créez un portail SiteWise Monitor dans la AWS IoT SiteWise console.

Pour créer un portail

1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Moniteur, Mise en route.



3. Choisissez Créer un portail.



Ensuite, vous devez fournir quelques informations de base pour configurer votre portail.

Configurez votre portail dans SiteWise Monitor

Les utilisateurs utilisent des portails pour afficher vos données. Vous pouvez personnaliser le nom, la description, l'image de marque, l'authentification des utilisateurs, l'e-mail de contact du service d'assistance et les autorisations d'un portail.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configurationStep 2- optional
Additional featuresStep 3
Invite administratorsStep 4
Assign users

Portal configuration

Each web portal provides enterprise users with access to your IoT SiteWise assets. [Learn more](#)

Portal details

Portal name

Choose a portal name to identify the web portal to your users. Company name is recommended.

Name should be 1-128 characters and only contain A-Z a-z 0-9 _ and -.

Description - optional

Create a description of your portal

Description should contain a maximum of 2048 characters.

Portal branding

You can provide your logo image to display your brand in this web portal.

Logo image

Upload a square, high-resolution .png file. The image is displayed on a dark background.

The file size must be less than 1 MB.

User authentication

Your users can sign in to this portal with their AWS Single Sign-On (AWS SSO) or AWS Identity and Access Management (IAM) credentials. If you choose AWS SSO, you must enable the service for your AWS account.

 You haven't enabled AWS SSO in your account yet. When you create your first portal user, this automatically enables AWS SSO in your AWS account.

☒ AWS SSO

Your users can sign in to the portal with their corporate usernames and passwords.

☐ IAM

Your users can sign in to the portal with their IAM credentials.

Support contact email

You can provide an email address for cases where there's a problem or issue with this portal and your users need to contact support to resolve.

Email

Tags

This resource doesn't have any tags.

You can add up to 50 more tags.

Permissions

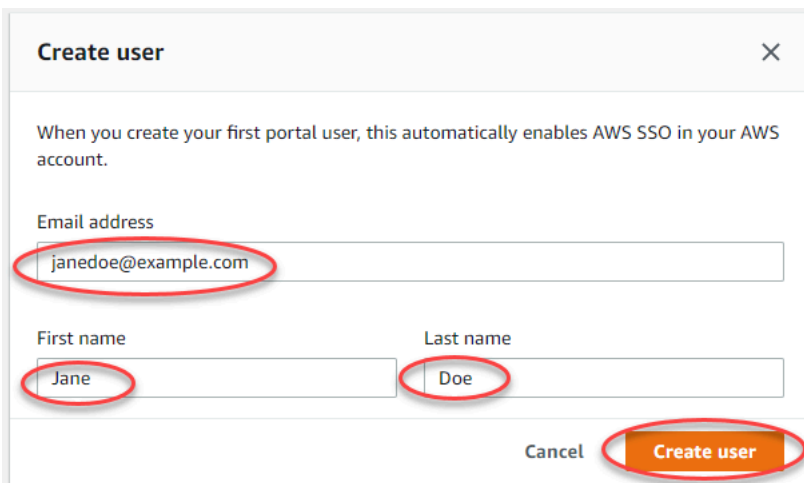
SiteWise Monitor assumes this role to give permissions to your federated users to access AWS IoT SiteWise resources. [Learn](#)

Pour configurer un portail

1. Entrez un nom pour votre portail.
2. (Facultatif) Saisissez une description pour votre portail. Si vous avez plusieurs portails, utilisez des descriptions significatives pour vous aider à suivre ce que contient chaque portail.
3. (Facultatif) Chargez une image pour afficher votre marque dans le portail. Choisissez une image PNG carrée. Si vous chargez une image qui n'est pas carrée, le portail réduit l'image à un carré.
4. Choisissez l'une des options suivantes :
 - Choisissez IAM Identity Center si les utilisateurs de votre portail se connectent à ce portail avec leur nom d'utilisateur et leur mot de passe d'entreprise.

Si vous n'avez pas activé IAM Identity Center dans votre compte, procédez comme suit :

- a. Choisissez Create user (Créer un utilisateur).
- b. Sur la page Créer un utilisateur, pour créer le premier portail, entrez l'adresse e-mail, le prénom et le nom de famille de l'utilisateur, puis choisissez Créer un utilisateur.



Note

- AWS active automatiquement IAM Identity Center dans votre compte lorsque vous créez le premier utilisateur du portail.
- Vous ne pouvez configurer IAM Identity Center que dans une seule région à la fois. SiteWise Monitor se connecte à la région que vous avez configurée pour IAM Identity Center. Cela signifie que vous utilisez une région pour

accéder à l'IAM Identity Center, mais que vous pouvez créer des portails dans n'importe quelle région.

- Choisissez IAM si les utilisateurs de votre portail se connectent à ce portail avec leurs informations d'identification IAM.

 Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

5. Entrez une adresse e-mail que les utilisateurs du portail peuvent contacter lorsqu'ils ont un problème avec le portail et ont besoin d'aide pour le résoudre.
6. (Facultatif) Ajoutez des balises pour votre portail. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
7. Choisissez l'une des options suivantes :
 - Choisissez Créer et utiliser un nouveau rôle de service. Par défaut, SiteWise Monitor crée automatiquement un rôle de service pour chaque portail. Ce rôle permet aux utilisateurs de votre portail d'accéder à vos AWS IoT SiteWise ressources. Pour de plus amples informations, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).
 - Choisissez Utiliser un rôle de service existant, puis choisissez le rôle cible.
8. Choisissez Next (Suivant)
9. (Facultatif) Activez les alarmes pour votre portail. Pour de plus amples informations, veuillez consulter [Activez les alarmes pour vos portails dans AWS IoT SiteWise](#).
10. Choisissez Créer. AWS IoT SiteWise créera votre portail.

 Note

Si vous fermez la console, vous pouvez terminer le processus d'installation en ajoutant des administrateurs et des utilisateurs. Pour de plus amples informations, veuillez consulter [Ajouter ou supprimer des administrateurs de portail dans AWS IoT SiteWise](#). Si vous ne souhaitez pas conserver ce portail, supprimez-le afin qu'il n'utilise pas de ressources. Pour de plus amples informations, veuillez consulter [Supprimer un portail dans AWS IoT SiteWise](#).

La colonne Status peut prendre l'une des valeurs suivantes.

- **CRÉATION** : AWS IoT SiteWise traite votre demande de création du portail. Ce processus peut prendre plusieurs minutes.
- **MISE À JOUR** : AWS IoT SiteWise traite votre demande de mise à jour du portail. Ce processus peut prendre plusieurs minutes.
- **EN ATTENTE** : AWS IoT SiteWise attend la fin de la propagation de l'enregistrement DNS. Ce processus peut prendre plusieurs minutes. Vous pouvez supprimer le portail lorsque le statut est EN ATTENTE.
- **SUPPRESSION** : AWS IoT SiteWise traite votre demande de suppression du portail. Ce processus peut prendre plusieurs minutes.
- **ACTIF** : lorsque le portail devient actif, les utilisateurs de votre portail peuvent y accéder.
- **ÉCHEC** : AWS IoT SiteWise impossible de traiter votre demande de création, de mise à jour ou de suppression du portail. Si vous avez activé AWS IoT SiteWise l'envoi de journaux vers Amazon CloudWatch Logs, vous pouvez utiliser ces journaux pour résoudre les problèmes. Pour plus d'informations, consultez la section [Surveillance AWS IoT SiteWise à l'aide de CloudWatch journaux](#).

Un message s'affiche lors de la création de votre portail.

A green banner with a white checkmark icon on the left and a white 'X' icon on the right. The text in the center reads: "Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws".

Successfully created portal URL at https://a1b2c3d4-5678-90ab-cdef-1111EXAMPLE.app.iotsitewise.aws

Ensuite, vous devez inviter un ou plusieurs administrateurs de portail dans le portail. Jusqu'à présent, vous avez créé un portail mais personne ne peut y accéder.

Inviter des administrateurs dans SiteWise Monitor

Pour commencer, vous devez affecter un administrateur à votre nouveau portail. L'administrateur du portail crée des projets, choisit les propriétaires de projets et affecte des ressources aux projets. Les administrateurs du portail peuvent consulter tous vos AWS IoT SiteWise actifs.

En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes :

IAM Identity Center

Si vous utilisez SiteWise Monitor pour la première fois, vous pouvez choisir l'utilisateur que vous avez créé précédemment comme administrateur du portail. Si vous souhaitez ajouter un autre

utilisateur en tant qu'administrateur du portail, vous pouvez créer un utilisateur IAM Identity Center à partir de cette page. Vous pouvez également connecter un fournisseur d'identité externe à IAM Identity Center. Pour plus d'informations, consultez le [AWS IAM Identity Center Guide de l'utilisateur](#).

Pour inviter les administrateurs

1. Activez les cases à cocher pour les utilisateurs qui seront administrateurs de votre portail. Cela ajoute les utilisateurs à la liste des administrateurs du portail.

 Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'administrateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

2. (Facultatif) Choisissez Envoyer l'invitation aux utilisateurs sélectionnés. Votre client de messagerie s'ouvre et une invitation est renseignée dans le corps du message.

Vous pouvez personnaliser l'e-mail avant de l'envoyer aux administrateurs de votre portail. Vous pouvez également envoyer l'e-mail aux administrateurs de votre portail ultérieurement. Si vous essayez SiteWise Monitor pour la première fois et que vous ajoutez votre nouvel utilisateur ou rôle IAM Identity Center ou IAM en tant qu'administrateur du portail, vous n'avez pas besoin de vous envoyer un e-mail.

3. Si vous ajoutez un utilisateur que vous ne souhaitez pas en tant qu'administrateur, désactivez la case à cocher correspondant à cet utilisateur.
4. Lorsque vous avez terminé d'inviter les administrateurs du portail, choisissez Suivant.

IAM

Vous pouvez choisir un utilisateur ou un rôle pour être l'administrateur du portail. Si vous souhaitez ajouter un autre utilisateur ou un autre rôle en tant qu'administrateur du portail, vous pouvez créer un utilisateur ou un rôle dans la console IAM. Pour plus d'informations, consultez les [sections Création d'un utilisateur IAM dans votre AWS compte](#) et [Création de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Pour inviter les administrateurs

1. Procédez comme suit :
 - Choisissez les utilisateurs IAM pour ajouter un utilisateur IAM en tant qu'administrateur de votre portail.
 - Choisissez les rôles IAM pour ajouter un rôle IAM en tant qu'administrateur du portail.
2. Cochez les cases correspondant aux utilisateurs ou aux rôles que vous souhaitez utiliser en tant qu'administrateurs de votre portail. Cela ajoute les utilisateurs ou les rôles à la liste des administrateurs du portail.
3. Si vous ajoutez un utilisateur ou un rôle que vous ne souhaitez pas utiliser en tant qu'administrateur, décochez la case correspondant à cet utilisateur ou à ce rôle.
4. Lorsque vous avez terminé d'inviter les administrateurs du portail, choisissez Suivant.

Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'administrateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

Vous pouvez modifier la liste des administrateurs de portail ultérieurement. Pour de plus amples informations, veuillez consulter [Ajouter ou supprimer des administrateurs de portail dans AWS IoT SiteWise](#).

 Note

Étant donné que seul un administrateur de portail peut créer des projets et leur attribuer des actifs, vous devez spécifier au moins un administrateur de portail.

La dernière étape consiste à ajouter des utilisateurs qui peuvent accéder à votre nouveau portail.

Ajouter des utilisateurs du portail dans SiteWise Monitor

Vous contrôlez les utilisateurs qui ont accès à vos portails. Dans chaque portail, les administrateurs du portail créent un ou plusieurs projets et attribuent des utilisateurs du portail en tant que propriétaires ou utilisateurs standard pour chaque projet. Chaque propriétaire de projet peut inviter d'autres utilisateurs de portail à administrer ou à afficher le projet.

En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes :

IAM Identity Center

Si vous souhaitez ajouter un utilisateur à la liste des utilisateurs, procédez comme suit.

Pour ajouter des utilisateurs du portail

1. Choisissez les utilisateurs dans la liste Utilisateurs à ajouter au portail. Cela ajoute les utilisateurs à la liste des utilisateurs du portail. Si vous utilisez SiteWise Monitor pour la première fois, il n'est pas nécessaire d'ajouter l'administrateur de votre portail en tant qu'utilisateur du portail.

 Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'utilisateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

2. Si vous ajoutez un utilisateur qui ne doit pas avoir accès au portail, désactivez la case à cocher correspondant à cet utilisateur.

3. Lorsque vous avez terminé de sélectionner des utilisateurs, choisissez Attribuer des utilisateurs.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
Invite administrators

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users (2) Create user

Find resources

	Display name	Email
☐	Jane Doe	janedoe@example.com
☒	John Doe	johndoe@example.com

Selected users (1)

Cancel Previous **Assign users**

IAM

Si vous voyez l'utilisateur ou le rôle que vous souhaitez ajouter dans la liste des utilisateurs ou des rôles IAM, procédez comme suit.

Pour ajouter des utilisateurs du portail

1. Utilisez les options suivantes :
 - Choisissez les utilisateurs IAM pour ajouter un utilisateur IAM en tant qu'utilisateur du portail.
 - Choisissez les rôles IAM pour ajouter un rôle IAM en tant qu'utilisateur du portail.

Si vous utilisez SiteWise Monitor pour la première fois, il n'est pas nécessaire d'ajouter l'administrateur de votre portail en tant qu'utilisateur du portail.

2. Cochez les cases correspondant aux utilisateurs ou aux rôles que vous souhaitez utiliser en tant qu'utilisateurs du portail. Cela ajoute les utilisateurs ou les rôles à la liste des utilisateurs du portail.

3. Si vous ajoutez un utilisateur qui ne doit pas avoir accès au portail, désactivez la case à cocher correspondant à cet utilisateur.
4. Lorsque vous avez terminé de sélectionner des utilisateurs, choisissez Attribuer des utilisateurs.

⚠ Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2
[Invite administrators](#)

Step 3
Assign users

Assign users

Select the users you want to be able to access and view this portal. Portal administrators will send invitations to these users at a later date. [Learn more](#)

Users **Roles**

IAM roles (66) [Manage roles in IAM console](#)

Find role name

	Name	Date created
☐	[REDACTED]	
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNCS-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► Portal users (2) [Remove](#)

Cancel Previous **Assign users**

Félicitations ! Vous avez créé avec succès un portail, affecté des administrateurs de portail et affecté des utilisateurs qui peuvent utiliser ce portail lorsqu'ils sont invités à le faire. Les administrateurs de votre portail peuvent désormais créer des projets et y ajouter des ressources. Ensuite, les propriétaires de votre projet pourront créer des tableaux de bord pour visualiser les données correspondant aux ressources de chaque projet.

Vous pourrez modifier la liste des utilisateurs du portail ultérieurement. Pour de plus amples informations, veuillez consulter [Ajouter ou supprimer des utilisateurs du portail dans AWS IoT SiteWise](#).

Si vous devez apporter des modifications au portail, reportez-vous à la section [Administrez vos portails SiteWise Monitor](#).

Pour commencer à utiliser le portail, reportez-vous à la section [Getting started](#) du Guide de l'application SiteWise Monitor.

Création de AWS IoT SiteWise tableaux de bord (AWS CLI)

Lorsque vous définissez des visualisations (ou des widgets) dans des tableaux de bord à l'aide du AWS CLI, vous devez spécifier les informations suivantes dans le document `dashboardDefinition` JSON. Cette définition est un paramètre des opérations [UpdateDashboard](#) et [CreateDashboard](#).

`widgets`

Liste des structures de définition de widget contenant chacune les informations suivantes :

`type`

Type de widget. AWS IoT SiteWise fournit les types de widgets suivants :

- `sc-line-chart`— Un graphique linéaire. Pour plus d'informations, voir les [diagrammes linéaires](#) dans le guide AWS IoT SiteWise Monitor d'application.
- `sc-scatter-chart`— Un diagramme de points. Pour plus d'informations, reportez-vous à la section [Graphiques](#) de points dans le guide AWS IoT SiteWise Monitor d'application.
- `sc-bar-chart`— Un graphique à barres. Pour plus d'informations, voir les [diagrammes à barres](#) dans le guide AWS IoT SiteWise Monitor d'application.
- `sc-status-grid`— Un widget d'état qui affiche la dernière valeur des propriétés des actifs sous forme de grille. Pour plus d'informations, consultez la section [Widgets d'état](#) dans le guide de AWS IoT SiteWise Monitor candidature.
- `sc-status-timeline`— Un widget d'état qui affiche les valeurs historiques des propriétés des actifs sous forme de chronologie. Pour plus d'informations, consultez la section [Widgets d'état](#) dans le guide de AWS IoT SiteWise Monitor candidature.
- `sc-kpi`— Une visualisation des indicateurs de performance clés (KPI). Pour plus d'informations, consultez la section [Widgets KPI](#) dans le guide de AWS IoT SiteWise Monitor l'application.

- `sc-table`— Un widget de tableau. Pour plus d'informations, consultez la section [Widgets de tableau](#) dans le guide de AWS IoT SiteWise Monitor l'application.

`title`

Le titre du widget.

`x`

Position horizontale du widget, à partir de la gauche de la grille. Cette valeur fait référence à la position du widget dans la grille du tableau de bord.

`y`

Position verticale du widget, à partir du haut de la grille. Cette valeur fait référence à la position du widget dans la grille du tableau de bord.

`width`

Largeur du widget, exprimée en nombre d'espaces sur la grille du tableau de bord.

`height`

Hauteur du widget, exprimée en nombre d'espaces sur la grille du tableau de bord.

`metrics`

Liste des structures de métrique qui définissent chacune un flux de données pour ce widget. Chaque structure de la liste doit contenir les informations suivantes :

`label`

Étiquette à afficher pour cette métrique.

`type`

Type de source de données pour cette métrique. AWS IoT SiteWise fournit les types de métriques suivants :

- `iotsitewise`— Le tableau de bord récupère les données d'une propriété d'actif dans AWS IoT SiteWise. Si vous choisissez cette option, vous devez définir `assetId` et `propertyId` pour cette métrique.

`assetId`

(Facultatif) ID d'une ressource dans AWS IoT SiteWise.

Ce champ est obligatoire si vous choisissez `iotsitewise` pour type cette métrique.

`propertyId`

(Facultatif) ID d'une propriété d'actif dans AWS IoT SiteWise.

Ce champ est obligatoire si vous choisissez `iotsitewise` pour type cette métrique.

`analysis`

(Facultatif) Structure qui définit l'analyse, telle que les courbes de tendance, à afficher pour le widget. Pour plus d'informations, consultez [la section Configuration des courbes de tendance](#) dans le Guide de AWS IoT SiteWise Monitor l'application. Vous pouvez ajouter une courbe de tendance de chaque type par propriété dans le widget. La structure d'analyse contient les informations suivantes :

`trends`

(Facultatif) Une liste de structures de tendance qui définissent chacune une analyse des tendances pour ce widget. Chaque structure de la liste contient les informations suivantes :

`type`

Type de ligne de tendance. Choisissez l'option suivante :

- `linear-regression`— Affiche une droite de régression linéaire. SiteWise Monitor utilise la méthode [des moindres carrés](#) pour calculer la régression linéaire.

`annotations`

(Facultatif) Une structure d'annotations qui définit les seuils pour le widget. Pour plus d'informations, consultez [la section Configuration des seuils](#) dans le Guide de AWS IoT SiteWise Monitor l'application. Vous pouvez ajouter jusqu'à six annotations par widget. La structure des annotations contient les informations suivantes :

`y`

(Facultatif) Liste des structures d'annotation qui définissent chacune un seuil horizontal pour ce widget. Chaque structure de la liste contient les informations suivantes :

`comparisonOperator`

L'opérateur de comparaison pour le seuil. Sélectionnez l'une des méthodes suivantes :

- LT— Mettez en évidence les propriétés dont au moins un point de données est inférieur à `value`.
- GT— Mettez en évidence les propriétés dont au moins un point de données est supérieur à `value`.
- LTE— Mettez en évidence les propriétés dont au moins un point de données est inférieur ou égal à `value`.
- GTE— Mettez en évidence les propriétés dont au moins un point de données est supérieur ou égal à `value`.
- EQ— Mettez en évidence les propriétés dont au moins un point de données est égal à `value`.

`value`

La valeur de seuil pour comparer les points de données avec `comparisonOperator`.

`color`

(Facultatif) Le code hexadécimal à 6 chiffres de la couleur du seuil. La visualisation affiche les légendes des propriétés dans cette couleur pour les propriétés dont au moins un point de données répond à la règle du seuil. La valeur par défaut est `black` (`#000000`).

`showValue`

(Facultatif) Indique s'il faut afficher ou non la valeur du seuil dans les marges du widget. La valeur par défaut est `true`.

`properties`

(Facultatif) Un dictionnaire plat des propriétés du widget. Les membres de cette structure dépendent du contexte. AWS IoT SiteWise fournit les widgets suivants qui utilisent `properties` :

- [Les graphiques linéaires](#), les [graphiques en nuage](#) de points et les [graphiques à barres](#) présentent les propriétés suivantes :

`colorDataAcrossThresholds`

(Facultatif) S'il faut ou non modifier la couleur des données qui dépassent les seuils dans ce widget. Lorsque vous activez cette option, les données qui franchissent un seuil apparaissent dans la couleur que vous choisissez. La valeur par défaut est `true`.

- [Les grilles de statut](#) ont les propriétés suivantes :

labels

(Facultatif) Structure qui définit les étiquettes à afficher sur la grille d'état. La structure des étiquettes contient les informations suivantes :

showValue

(Facultatif) Indique s'il faut afficher ou non l'unité et la valeur de chaque propriété d'actif dans ce widget. La valeur par défaut est `true`.

Exemple Exemple de définition de tableau de bord

L'exemple suivant définit un tableau de bord à partir d'une charge utile stockée dans un fichier JSON.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeeeEXAMPLE \  
  --dashboard-name "Wind Farm Dashboard" \  
  --dashboard-definition file://dashboard-definition.json
```

L'exemple JSON suivant pour `dashboard-definition.json` définit un tableau de bord avec les widgets de visualisation suivants :

- Graphique linéaire qui visualise la puissance totale du parc éolien en haut à gauche du tableau de bord. Ce graphique linéaire inclut un seuil qui indique le moment où le parc éolien produit moins d'énergie que sa production minimale attendue. Ce graphique linéaire inclut également une courbe de tendance de régression linéaire.
- Diagramme à barres qui visualise la vitesse du vent pour quatre turbines en haut à droite du tableau de bord.

Note

Cet exemple représente des visualisations de diagrammes linéaires et à barres sur un tableau de bord. Ce tableau de bord est similaire à l'[exemple de tableau de bord de parc éolien](#).

```
{  
  "widgets": [  
    {
```

```

    "type": "sc-line-chart",
    "title": "Total Average Power",
    "x": 0,
    "y": 0,
    "height": 3,
    "width": 3,
    "metrics": [
      {
        "label": "Power",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
        "propertyId": "a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
        "analysis": {
          "trends": [
            {
              "type": "linear-regression"
            }
          ]
        }
      }
    ],
    "annotations": {
      "y": [
        {
          "comparisonOperator": "LT",
          "value": 20000,
          "color": "#D13212",
          "showValue": true
        }
      ]
    }
  },
  {
    "type": "sc-bar-chart",
    "title": "Wind Speed",
    "x": 3,
    "y": 3,
    "height": 3,
    "width": 3,
    "metrics": [
      {
        "label": "Turbine 1",
        "type": "iotsitewise",
        "assetId": "a1b2c3d4-5678-90ab-cdef-2a2a2EXAMPLE",

```

```
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 2",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2b2b2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 3",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2c2c2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  },
  {
    "label": "Turbine 4",
    "type": "iotsitewise",
    "assetId": "a1b2c3d4-5678-90ab-cdef-2d2d2EXAMPLE",
    "propertyId": "a1b2c3d4-5678-90ab-cdef-55555EXAMPLE"
  }
]
}
```

Activez les alarmes pour vos portails dans AWS IoT SiteWise

Vous pouvez activer la fonction d'alarmes prise en charge par AWS IoT Events for your portal afin que les administrateurs du portail puissent créer, modifier et supprimer des modèles AWS IoT Events d'alarme dans vos portails SiteWise Monitor. Les propriétaires de projets peuvent configurer des alarmes. Les spectateurs du projet peuvent consulter les détails des alarmes. Cette section explique comment utiliser la AWS IoT SiteWise console pour activer la fonction d'alarmes pour vos portails.

Important

- Vous ne pouvez pas créer d'alarmes externes dans vos portails.
- Si vous souhaitez envoyer des notifications d'alarme, vous devez choisir IAM Identity Center pour le service d'authentification des utilisateurs.
- La fonction de notification d'alarme n'est pas disponible en Chine (Pékin) Région AWS.

Lorsque vous configurez et créez un portail, vous pouvez activer les alarmes et les notifications d'alarme à l'étape 2 Fonctionnalités supplémentaires. En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes :

IAM Identity Center

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2 - optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - optional

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

☒ **Enable alarms**
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template

☐ Use an existing role

☒ **Enable alarm notifications**
If enabled, alarms can send email or SMS notifications.

Sender
Specify the email address that sends alarm notifications. To edit or add a sender, go to the [Amazon SES console](#).

AWS Lambda role
Choose an IAM role that allows AWS Lambda to send data to Amazon SES and Amazon SNS. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template

☐ Use an existing role

AWS Lambda function
Choose an AWS Lambda function to manage alarm notifications. To edit the function, go to the [AWS Lambda console](#).

☒ Create a lambda from an AWS managed template

☐ Use an existing lambda

Previous **Create**

Pour activer les alarmes pour un portail

1. (Facultatif) Choisissez Activer les alarmes.
 - Pour le rôle d'AWS IoT SiteWise accès, utilisez un rôle existant ou créez un rôle avec les autorisations requises. Ce rôle nécessite `iot:events:BatchPutMessage` et une relation de confiance qui permettent `iot.amazonaws.com` et `iotevents.amazonaws.com` permettent d'assumer le rôle.

2. (Facultatif) Choisissez Activer les notifications d'alarme.

- a. Pour Expéditeur, choisissez l'expéditeur.

Important

Vous devez vérifier l'adresse e-mail de l'expéditeur dans Amazon SES. Pour plus d'informations, consultez la section [Vérification des adresses e-mail dans Amazon SES](#), dans le manuel Amazon Simple Email Service Developer Guide.

- b. Pour AWS Lambda le rôle, utilisez un rôle existant ou créez-en un avec les autorisations requises. Ce rôle nécessite les `sso-directory:DescribeUser` autorisations `lambda:InvokeFunction` et une relation de confiance qui permet `iotevents.amazonaws.com` et permet `lambda.amazonaws.com` d'assumer le rôle.
- c. Pour les AWS Lambda fonctions, choisissez une fonction Lambda existante ou créez une fonction qui gère les notifications d'alarme. Pour plus d'informations, consultez [la section Gestion des notifications d'alarme](#) dans le Guide du AWS IoT Events développeur.

IAM

AWS IoT SiteWise > Monitor > Portals > Create portal

Step 1
Portal configuration

Step 2- optional
Additional features

Step 3
Invite administrators

Step 4
Assign users

Additional features - optional

Alarms

Your portal users can create alarms in the portal to monitor equipment or processes. They can also get notified when the equipment or processes perform outside specified range.

☒ **Enable alarms**
If enabled, your portal users can define AWS IoT Events alarms in SiteWise Monitor.

AWS IoT SiteWise access role
Choose an IAM role that allows AWS IoT Events to send data to AWS IoT SiteWise. To edit the role, go to the [IAM console](#).

☒ Create a role from an AWS managed template

☐ Use an existing role

 Alarms created in the portal can't send notifications. If you want to send alarm notifications, choose **Previous**. Then, on the **Portal configuration** page, choose **AWS SSO for User authentication**.

Previous **Create**

Pour activer les alarmes pour un portail

- (Facultatif) Choisissez Activer les alarmes.
 - Pour le rôle d'AWS IoT SiteWise accès, utilisez un rôle existant ou créez un rôle avec les autorisations requises. Ce rôle nécessite `iot:events:BatchPutMessage` et une relation de confiance qui permettent `iot.amazonaws.com` et `iotevents.amazonaws.com` permettent d'assumer le rôle.

Pour plus d'informations sur les alarmes dans SiteWise Monitor, voir [Surveillance à l'aide d'alarmes](#) dans le guide AWS IoT SiteWise d'application.

Activation de votre AWS IoT SiteWise portail à la périphérie

Une fois que vous avez activé votre portail en périphérie, celui-ci est disponible sur toutes les passerelles SiteWise Edge avec le pack de traitement des données activé dans votre compte.

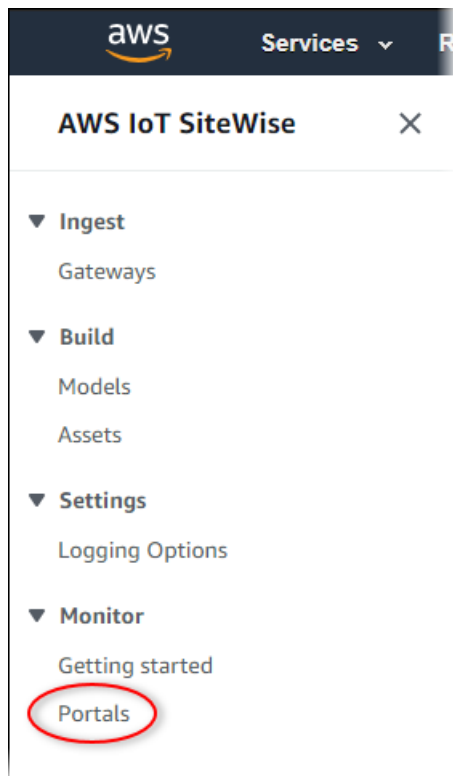
Pour activer le portail en périphérie

1. Dans la section Configuration Edge, activez Activer ce portail en périphérie.
2. Sélectionnez Create (Créer).

Administrez vos portails SiteWise Monitor

Vous avez la possibilité de gérer et de configurer différents aspects du portail. Cela inclut l'ajout et la suppression d'utilisateurs ou d'administrateurs, la définition des autorisations et des rôles des utilisateurs, la personnalisation de l'URL et du nom du portail, la définition des coordonnées d'assistance et l'envoi d'invitations par e-mail aux administrateurs du portail.

1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Moniteur, Portails.



3. Choisissez le portail, puis choisissez View details (Afficher les détails) (ou choisissez le nom du portail).
4. Vous pouvez effectuer l'une des tâches administratives suivantes :
 - [Modifier les détails du portail dans AWS IoT SiteWise](#)
 - [Ajouter ou supprimer des administrateurs de portail dans AWS IoT SiteWise](#)
 - [Envoyer des invitations par e-mail aux administrateurs du portail](#)
 - [Ajouter ou supprimer des utilisateurs du portail dans AWS IoT SiteWise](#)
 - [Supprimer un portail dans AWS IoT SiteWise](#)

Pour obtenir des informations sur la façon de créer un portail, veuillez consulter [Commencez avec AWS IoT SiteWise Monitor \(Classic\)](#).

Rubriques

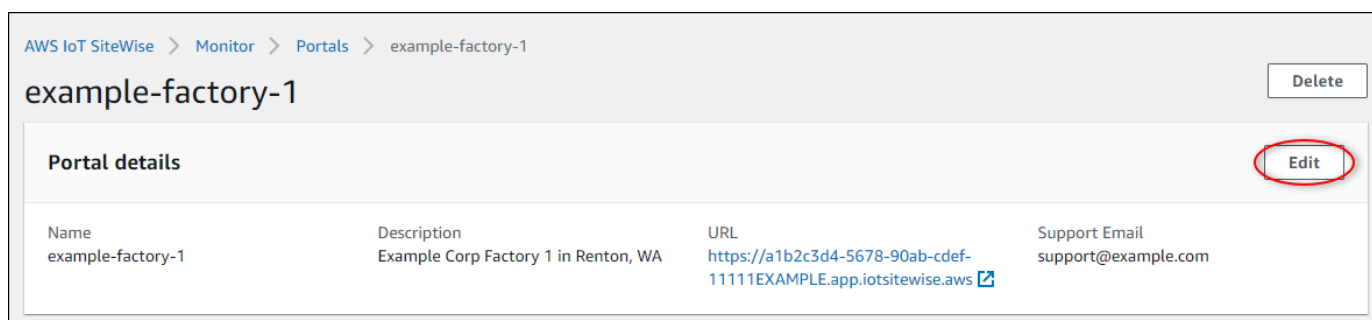
- [Modifier les détails du portail dans AWS IoT SiteWise](#)
- [Ajouter ou supprimer des administrateurs de portail dans AWS IoT SiteWise](#)
- [Envoyer des invitations par e-mail aux administrateurs du portail](#)
- [Ajouter ou supprimer des utilisateurs du portail dans AWS IoT SiteWise](#)

- [Supprimer un portail dans AWS IoT SiteWise](#)

Modifier les détails du portail dans AWS IoT SiteWise

Vous pouvez modifier le nom, la description, l'e-mail de support la personnalisation et les autorisations d'un portail.

1. Sur la page de détails du portail, dans la section Détails du portail choisissez Modifier.

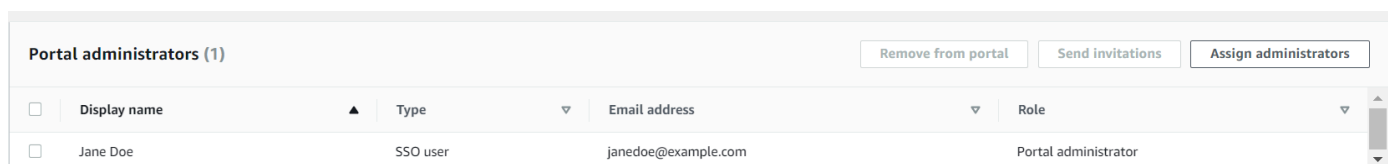


2. Mettez à jour les champs Nom, Description, Personnalisation du portail, E-mail de contact du support, ou Autorisations.
3. Lorsque vous avez terminé, sélectionnez Enregistrer.

Ajouter ou supprimer des administrateurs de portail dans AWS IoT SiteWise

En quelques étapes, vous pouvez ajouter ou supprimer des utilisateurs en tant qu'administrateurs d'un portail. En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes.

IAM Identity Center



Pour ajouter des administrateurs de portail

1. Sur la page des détails du portail, dans la section Administrateurs du portail, sélectionnez Affecter des administrateurs.
2. Sur la page Affecter des administrateurs, cochez les cases correspondant aux utilisateurs à ajouter au portail en tant qu'administrateurs.

Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'administrateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

3. Choisissez Attribuer des administrateurs.

AWS IoT SiteWise > Monitor > Portals > example-factory-1 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

Users (2)

Find resources

Display name	Email
☐ Jane Doe	janedoe@example.com
☒ John Doe	john Doe

Selected users (1)

Cancel Assign administrators

Pour supprimer des administrateurs de portail

- Dans la page de détails du portail, dans la section Portal administrators (Administrateurs du portail), cochez la case de chaque utilisateur à supprimer, puis choisissez Remove from portal (Supprimer du portail).

Note

Nous vous recommandons de sélectionner au moins un administrateur du portail.

IAM

Portal administrators (1)

Remove from portal Send invitations Assign administrators

Display name	Type	Email address	Role
☒ [redacted]	IAM user	-	Portal administrator

Pour ajouter des administrateurs de portail

1. Sur la page des détails du portail, dans la section Administrateurs du portail, sélectionnez Affecter des administrateurs.
2. Sur la page Affecter des administrateurs, procédez comme suit :
 - Choisissez Utilisateurs IAM si vous souhaitez ajouter un utilisateur IAM en tant qu'administrateur de votre portail.
 - Choisissez les rôles IAM si vous souhaitez ajouter un rôle IAM en tant qu'administrateur de portail.
3. Cochez les cases correspondant aux utilisateurs ou aux rôles que vous souhaitez utiliser en tant qu'administrateurs de votre portail. Cela ajoute les utilisateurs ou les rôles à la liste des administrateurs du portail.
4. Choisissez Attribuer des administrateurs.

Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

AWS IoT SiteWise > Monitor > Portals > example-factory-2 > Assign administrators

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

 IAM users or roles must have the `iotsitewise:DescribePortal` permission to sign in to the portal.

Users Roles

IAM users (1) [Manage users in IAM console](#)

Find user name

☒	Name	Date created
☒	raspberrypi-testing	11-08-2019

Portal administrators (1) [Remove](#)

[Cancel](#) [Assign administrators](#)

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > [example-factory-2](#) > [Assign administrators](#)

Assign administrators

Choose the users that you want to be portal administrators. Portal administrators can grant users access to specific industrial equipment data. [Learn more](#)

ⓘ IAM users or roles must have the `lotsitewise:DescribePortal` permission to sign in to the portal.

Users **Roles**

IAM roles (66)

[Manage roles in IAM console](#)

Find role name

< 1 2 3 4 5 6 7 >

☐	Name	Date created
☐	[REDACTED]	
☒	AWSIoTSiteWiseMonitorServiceRole_4wZigNpA1	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_ECKT-2Oar	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_GTnd0O4Wr	03-16-2021
☐	AWSIoTSiteWiseMonitorServiceRole_rHINLNC5-	03-11-2021
☐	AWSIoTSiteWiseMonitorServiceRole_XB330QUIO	03-10-2021
☐	[REDACTED]	
☐	[REDACTED]	
☐	[REDACTED]	

► Portal administrators (2)

Remove

Cancel **Assign administrators**

Pour supprimer des administrateurs de portail

- Dans la page de détails du portail, dans la section Portal administrators (Administrateurs du portail), cochez la case de chaque utilisateur à supprimer, puis choisissez Remove from portal (Supprimer du portail).

Note

Il n'est pas recommandé de laisser un portail sans administrateur de portail.

Envoyer des invitations par e-mail aux administrateurs du portail

Vous pouvez envoyer des invitations par e-mail aux administrateurs du portail.

- Sur la page de détails du portail, dans la section Portal administrators (Administrateurs du portail), activez les cases à cocher correspondant aux administrateurs du portail.

Portal administrators (1)				Remove from portal	Send invitations	Assign users
☒	Display name	Email address	Role			
☒	John Doe	john.doe@example.com	Portal administrator			

2. Choisissez Envoyer des invitations. Votre client de messagerie s'ouvre et une invitation est renseignée dans le corps du message.

Vous pouvez personnaliser l'e-mail avant de l'envoyer aux administrateurs de votre portail.

Ajouter ou supprimer des utilisateurs du portail dans AWS IoT SiteWise

Vous contrôlez les utilisateurs qui ont accès à votre portail. Les utilisateurs du portail apparaissent dans la liste des utilisateurs d'un portail SiteWise Monitor. Dans cette liste, les administrateurs du portail peuvent ajouter des propriétaires de projet, et les propriétaires de projet peuvent ajouter des utilisateurs de projet.

Note

Les administrateurs de votre portail et les utilisateurs du portail peuvent vous contacter via l'adresse e-mail de support d'un portail s'ils ont besoin que vous ajoutiez ou supprimiez un utilisateur.

En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes.

IAM Identity Center

Portal users (1)					Remove from portal	Assign users
☐	Display name	Type	Email address	Role		
☐	John Doe	SSO user	johndoe@example.com	Portal viewer		

Pour ajouter des utilisateurs du portail

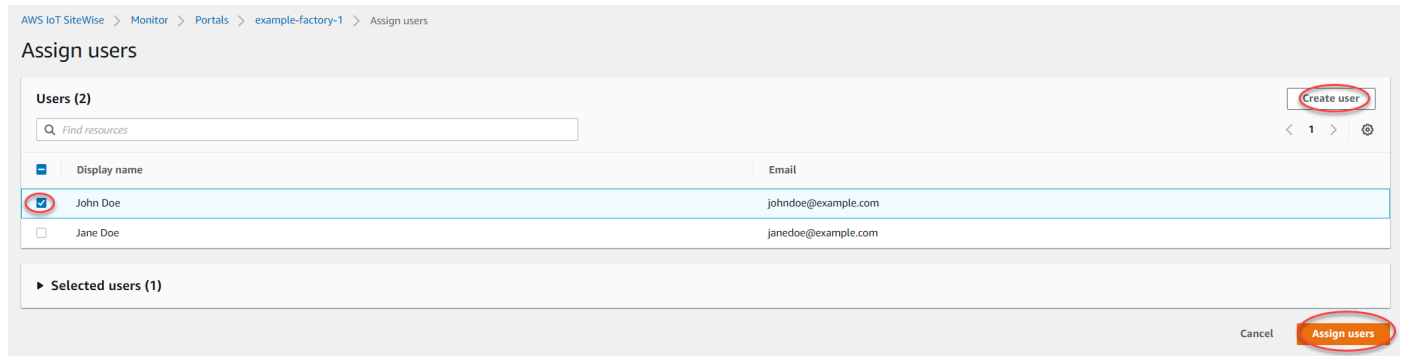
1. Sur la page de détails du portail, dans la section Portal users (Utilisateurs du portail), choisissez Assign users (Affecter des utilisateurs).
2. Sur la page Attribuer des utilisateurs, cochez la case correspondant aux utilisateurs à ajouter au portail.

Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-

mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'utilisateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

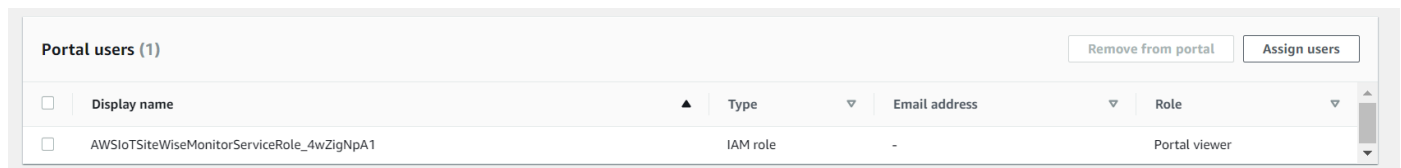
3. Choisissez Assign users (Affecter des utilisateurs).



Pour supprimer des utilisateurs du portail

- Sur la page des détails du portail, dans la section Utilisateurs du portail, cochez la case correspondant aux utilisateurs à supprimer du portail, puis choisissez Supprimer du portail.

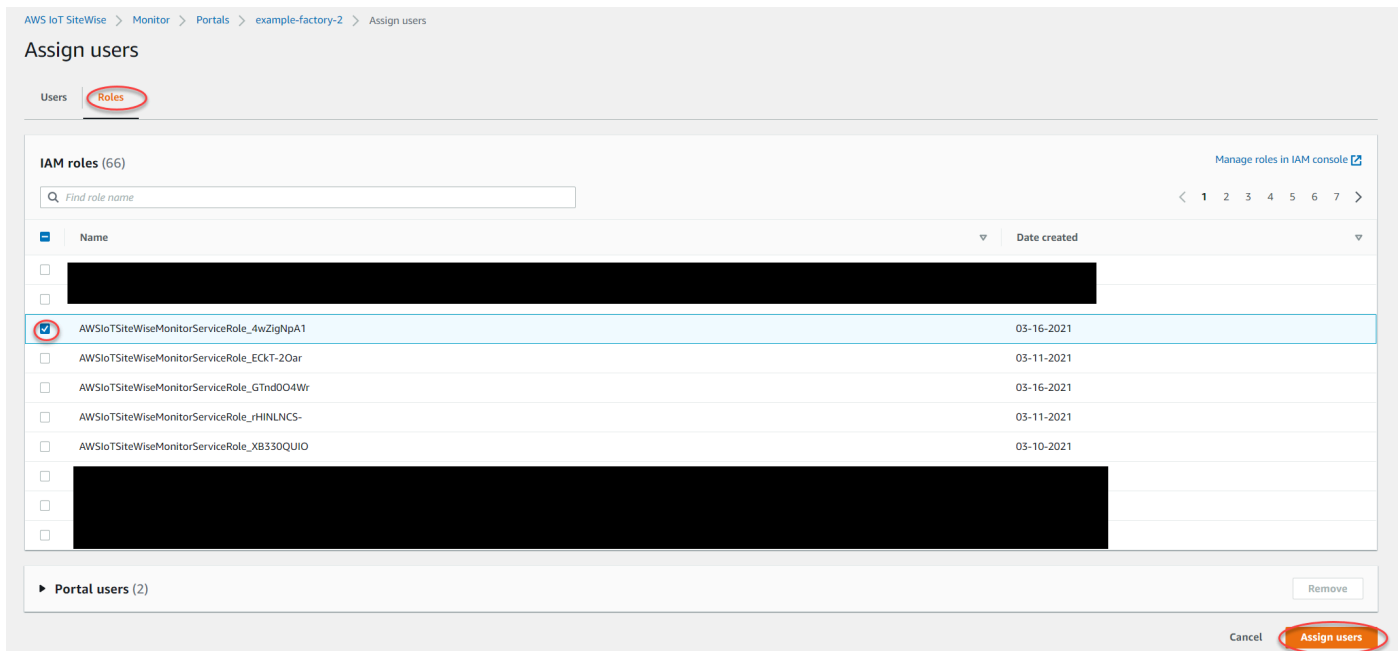
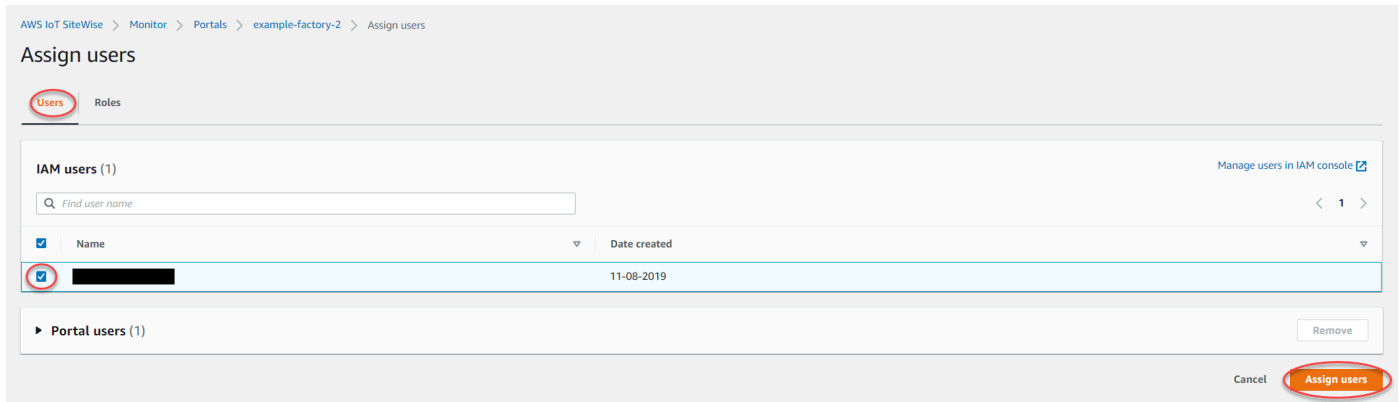
IAM



Pour ajouter des utilisateurs du portail

- Sur la page de détails du portail, dans la section Portal users (Utilisateurs du portail), choisissez Assign users (Affecter des utilisateurs).
- Sur la page Attribuer des utilisateurs, procédez comme suit :
 - Choisissez les utilisateurs IAM pour ajouter un utilisateur IAM en tant qu'utilisateur de votre portail.
 - Choisissez les rôles IAM pour ajouter un rôle IAM en tant qu'utilisateur de votre portail.

3. Cochez les cases correspondant aux utilisateurs ou aux rôles que vous souhaitez ajouter en tant qu'utilisateurs de votre portail. Cela ajoute les utilisateurs ou les rôles à la liste des utilisateurs du portail.
4. Choisissez Assign users (Affecter des utilisateurs).



Pour supprimer des utilisateurs du portail

- Sur la page des détails du portail, dans la section Utilisateurs du portail, cochez la case correspondant aux utilisateurs à supprimer du portail, puis choisissez Supprimer du portail.

⚠ Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

Supprimer un portail dans AWS IoT SiteWise

Vous pouvez supprimer un portail si vous l'avez créé à des fins de test ou en cas de doublon.

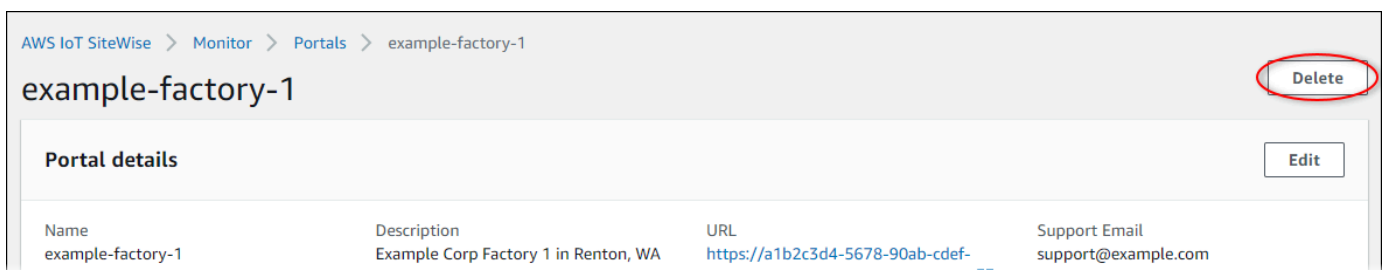
ℹ Note

Vous devez supprimer manuellement tous les tableaux de bord et projets du portail avant de pouvoir supprimer un portail. Pour plus d'informations, consultez les sections [Suppression de projets](#) et [Suppression de tableaux](#) de bord dans le Guide de l'application SiteWise Monitor.

1. Sur la page de détails du portail, choisissez Supprimer.

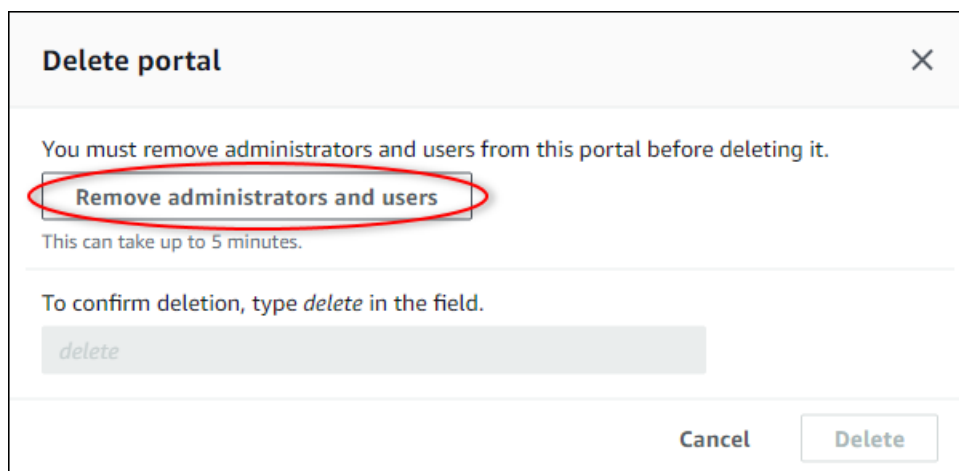
⚠ Important

Lorsque vous supprimez un portail, vous perdez tous les projets qu'il contient et tous les tableaux de bord de chaque projet. Cette action ne peut pas être annulée. Les données de vos ressources ne sont pas affectées.

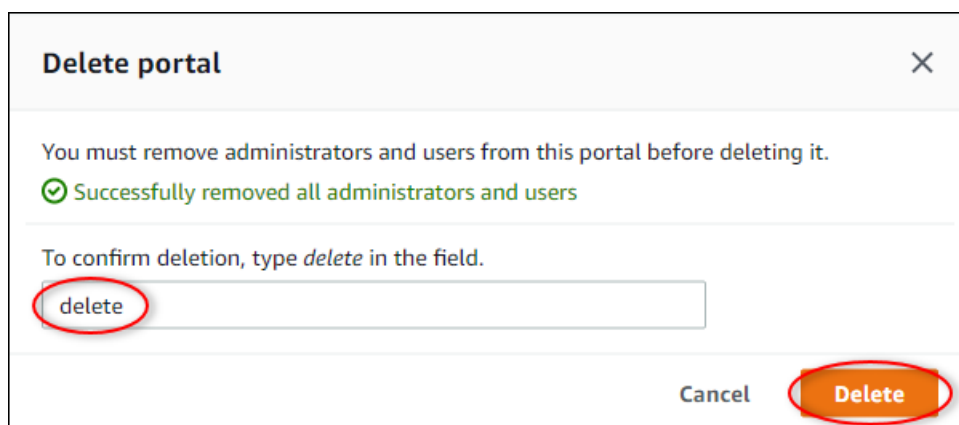


2. Dans la boîte de dialogue Supprimer les portails, choisissez Supprimer les administrateurs et les utilisateurs.

Vous devez supprimer les administrateurs et les utilisateurs du portail avant de pouvoir supprimer ce dernier. Si votre portail ne dispose pas d'administrateurs ou d'utilisateurs, le bouton n'apparaît pas et vous pouvez passer à l'étape suivante.



3. Si vous êtes sûr de vouloir supprimer l'intégralité du portail, saisissez **delete** dans le champ pour confirmer la suppression.



4. Sélectionnez Supprimer.

Commencer avec AWS IoT SiteWise Monitor (compatible avec l'IA) - aperçu

En tant qu' AWS administrateur de votre organisation, vous pouvez créer des portails à partir de la AWS IoT SiteWise console, permettant aux membres de votre organisation de consulter vos AWS IoT SiteWise données. Pour commencer, suivez les étapes ci-dessous.

1. Configurez et créez un portail.
2. Ajoutez des administrateurs de portail et envoyez des e-mails d'invitation.
3. Ajoutez des utilisateurs du portail.

Après avoir créé un portail, l'administrateur du portail peut créer des projets et ajouter un utilisateur au projet. Les membres du projet créent ensuite des tableaux de bord AWS IoT SiteWise pour visualiser les données connectées et leur permettre de surveiller les performances de leurs appareils, processus et équipements connectés.

Note

Lorsque vous ajoutez des utilisateurs ou des administrateurs au portail, évitez de créer des politiques AWS Identity and Access Management (IAM) qui limitent les autorisations des utilisateurs, telles qu'une adresse IP limitée. Les politiques associées avec des autorisations restreintes ne pourront pas se connecter au AWS IoT SiteWise portail.

Créez des projets à partager avec vos équipes. Les responsables de projet peuvent ensuite créer des tableaux de bord pour visualiser les propriétés des actifs afin d'aider les personnes chargées du projet à comprendre les performances des appareils, des processus et des équipements. Il fournit également une vue cohérente des opérations à vos équipes.

Les tableaux de bord permettent de visualiser et de comprendre les données de votre projet. Il aide les entreprises et les utilisateurs d'applications à suivre leurs AWS IoT appareils et leurs données. Choisissez le type de visualisation qui affiche le mieux vos données en fonction de vos besoins. Réorganisez et redimensionnez les visualisations pour créer une mise en page adaptée à votre équipe. Explorez les actifs et les données de vos appareils, processus et équipements, identifiez rapidement les problèmes et améliorez l'efficacité opérationnelle.

Rubriques

- [Création d'un portail](#)
- [Configurez votre portail](#)
- [Administrez vos portails](#)
- [Supprimer un portail](#)
- [Créez des tableaux de bord avec AWS CLI](#)
- [Connexion au portail](#)
- [Création d'un projet](#)
- [Met à jour un projet.](#)
- [Supprime un projet.](#)

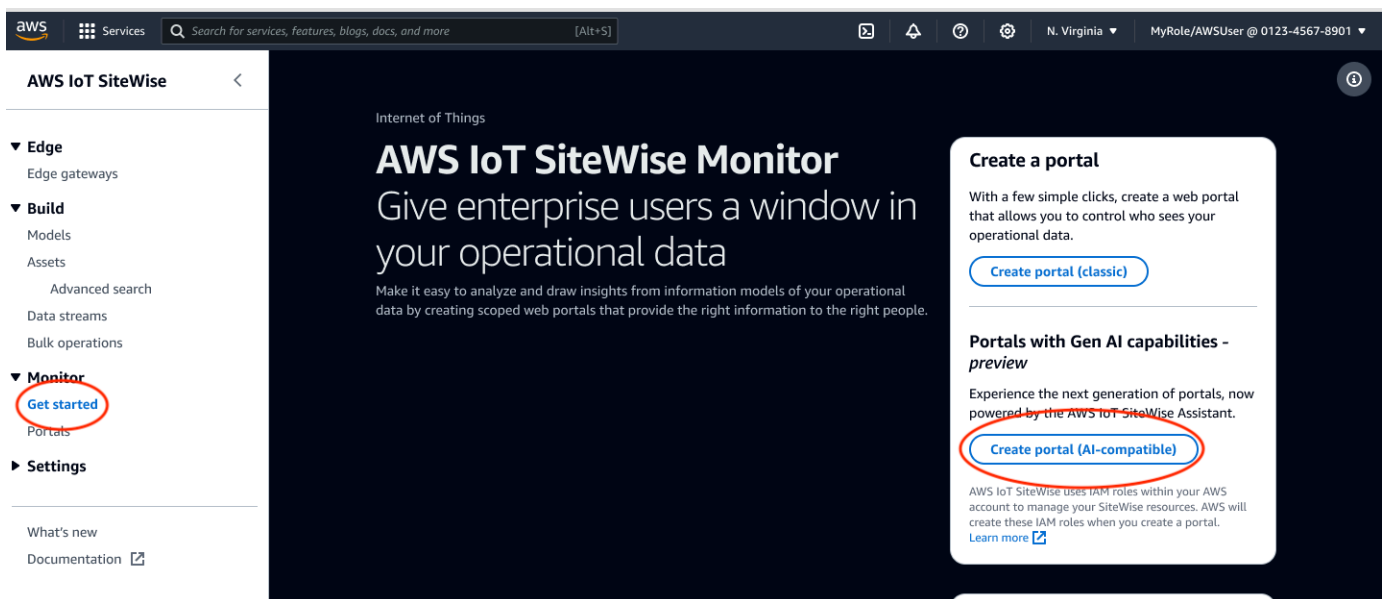
- [Création d'un tableau de bord](#)
- [Mise à jour d'un tableau de bord](#)
- [Suppression d'un tableau de bord](#)
- [Configuration du tableau de bord](#)

Création d'un portail

Vous créez un portail SiteWise Monitor dans la AWS IoT SiteWise console.

Pour créer un portail

1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Monitor, Get started.
3. Choisissez Créer un portail (compatible avec l'IA).



Ensuite, vous devez fournir quelques informations de base pour configurer votre portail.

Configurez votre portail

Les utilisateurs utilisent des portails pour afficher vos données. Vous pouvez personnaliser le nom, la description, l'image de marque, l'authentification des utilisateurs, l'e-mail de contact du service d'assistance et les autorisations d'un portail.

Étapes pour configurer un portail :

1. Entrez un nom pour votre portail.
2. (Facultatif) Saisissez une description pour votre portail. Si vous avez plusieurs portails, utilisez des descriptions significatives pour vous aider à suivre ce que contient chaque portail.
3. (Facultatif) Chargez une image pour afficher votre marque dans le portail. Choisissez une image PNG carrée. Si vous chargez une image qui n'est pas carrée, le portail réduit l'image à un carré.
4. Entrez une adresse e-mail dans la boîte e-mail de contact du support pour les problèmes d'assistance.
5. Dans le champ Authentification utilisateur, choisissez l'option suivante :
 - Choisissez IAM Identity Center si les utilisateurs de votre portail se connectent à ce portail avec leur nom d'utilisateur et leur mot de passe d'entreprise.

Si vous n'avez pas activé IAM Identity Center dans votre compte, procédez comme suit :

- a. Choisissez Create user (Créer un utilisateur).
- b. Sur la page Créer un utilisateur, pour créer le premier portail, entrez l'adresse e-mail, le prénom et le nom de famille de l'utilisateur, puis choisissez Créer un utilisateur.

Note

Support pour les informations d'identification IAM bientôt disponible.

6. Choisissez l'une des options suivantes dans la section Accès au service :
 - Choisissez Créer et utiliser un nouveau rôle de service. Par défaut, SiteWise Monitor crée automatiquement un rôle de service pour chaque portail. Ce rôle permet aux utilisateurs de votre portail d'accéder à vos AWS IoT SiteWise ressources. Pour de plus amples informations, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).
 - Choisissez Utiliser un rôle de service existant, puis choisissez le rôle cible.
7. Choisissez d'activer l' AWS IoT SiteWise assistant pour ce portail. L' AWS IoT SiteWise assistant fournit une analyse rapide des données, des informations en temps réel et des recommandations guidées.

 Note

L'activation de l' AWS IoT SiteWise Assistant entraînera des frais. Pour utiliser les solutions de connaissances et les conseils au niveau de l'entreprise, vous devez disposer d'un ensemble de données associé à l'index Amazon Kendra.

8. (Facultatif) Ajoutez des balises pour votre portail. Pour de plus amples informations, veuillez consulter [Marquez vos AWS IoT SiteWise ressources](#).
9. Choisissez Create portal. AWS IoT SiteWise créera votre portail.

 Note

Si vous fermez la console, vous pouvez terminer le processus d'installation en ajoutant des administrateurs et des utilisateurs. Pour de plus amples informations, veuillez consulter [Ajouter ou supprimer des administrateurs de portail](#). Si vous ne souhaitez pas conserver ce portail, supprimez-le afin qu'il n'utilise pas de ressources. Pour de plus amples informations, veuillez consulter [Supprimer un portail](#).

Un message s'affiche lors de la création de votre portail.

Successfully created portal "example portal".

[AWS IoT SiteWise](#) > [Monitor](#) > [Portals](#) > example portal

example portal

Edit

Delete

Open portal

Portal details

Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

Une fois qu'un portail est créé, il est répertorié dans la section Portails. La section Détails du portail répertorie le nom, la description, l'identifiant, l'URL, le statut, les dates de dernière mise à jour et de création, l'image de marque du portail et l'e-mail d'assistance de chaque portail.

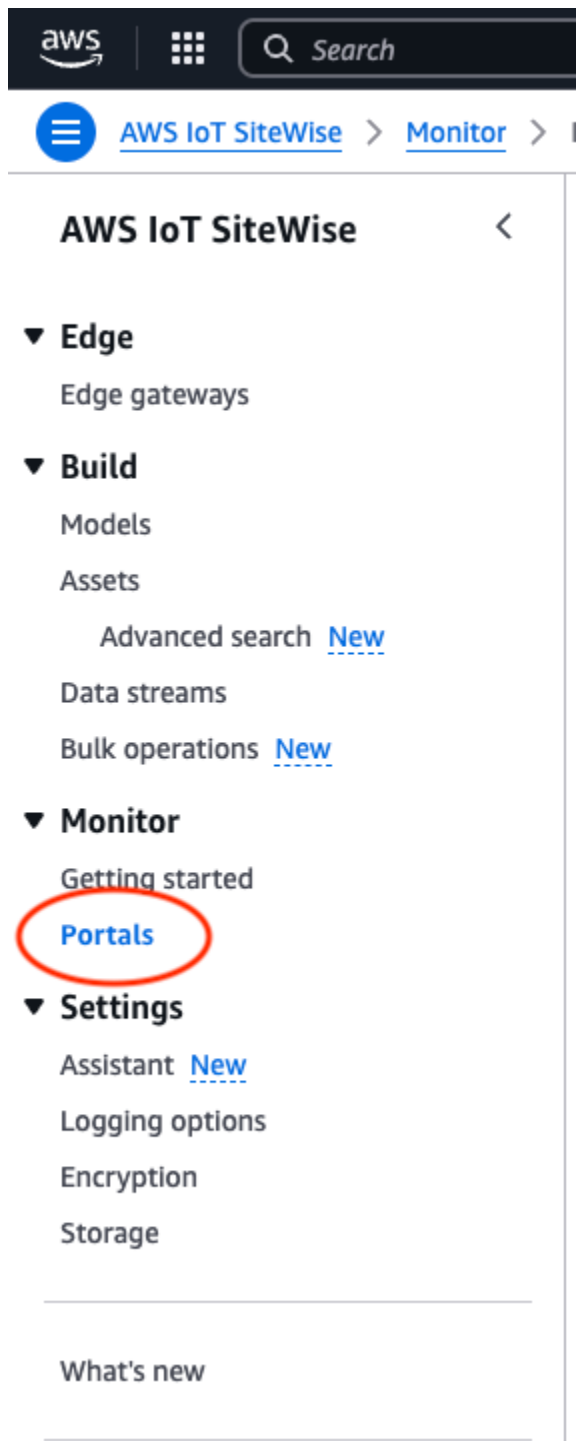
La colonne Status peut prendre l'une des valeurs suivantes.

- **AWS IoT SiteWise CREATING** - traite votre demande de création du portail. Ce processus peut prendre plusieurs minutes.
- **MISE À JOUR** - AWS IoT SiteWise traite votre demande de mise à jour du portail. Ce processus peut prendre plusieurs minutes.
- **PENDING** - AWS IoT SiteWise attend la fin de la propagation de l'enregistrement DNS. Ce processus peut prendre plusieurs minutes. Vous pouvez supprimer le portail lorsque le statut est **EN ATTENTE**.
- **AWS IoT SiteWise DELETING** - traite votre demande de suppression du portail. Ce processus peut prendre plusieurs minutes.
- **ACTIF** - Lorsque le portail devient actif, les utilisateurs de votre portail peuvent y accéder.
- **ÉCHEC** - AWS IoT SiteWise Impossible de traiter votre demande de création, de mise à jour ou de suppression du portail. Si vous avez activé AWS IoT SiteWise l'envoi de journaux vers Amazon CloudWatch Logs, vous pouvez utiliser ces journaux pour résoudre les problèmes. Pour plus d'informations, consultez la section [Surveillance AWS IoT SiteWise à l'aide de CloudWatch journaux](#).

Administrez vos portails

Vous avez la possibilité de gérer et de configurer différents aspects du portail. Cela inclut l'ajout et la suppression d'administrateurs, la définition des autorisations et des rôles, la personnalisation du nom, de la description, la configuration du courrier électronique d'assistance et l'invitation aux administrateurs du portail.

1. Connectez-vous à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez **Moniteur, Portails**.



3. Choisissez un portail, puis choisissez Ouvrir le portail (ou choisissez le nom du portail).
4. Vous pouvez effectuer l'une des tâches administratives suivantes :
 - [Modifier les attributs du portail](#)
 - [Ajouter ou supprimer des administrateurs de portail](#)

- [Envoyer des invitations par e-mail aux administrateurs du portail](#)
- [Supprimer un portail dans AWS IoT SiteWise](#)

Modifier les attributs du portail

Vous pouvez modifier le nom, la description, l'image de marque, l'e-mail d'assistance et l'accès aux services d'un portail.

1. Sur la page de détails du portail, dans la section Détails du portail choisissez Modifier.

Successfully created portal "example portal".

AWS IoT SiteWise > Monitor > Portals > example portal

example portal

Edit Delete Open portal

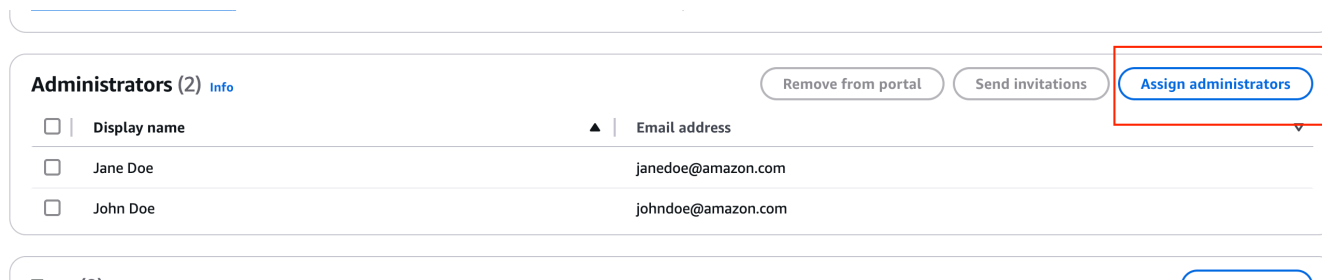
Portal details		
Name example portal	Status Active	URL https://p-jfnlf2d8.gamma.iotsitewise.aws
Type AI-compatible	AWS IoT SiteWise Assistant Disabled	Portal branding -
Description -	Last updated November 1, 2024, 15:37 (UTC-07:00)	Support contact email myemail@mycompany.com
ID f5fc93a1-011c-4c5a-81a6-e001b50d2547	Date created November 1, 2024, 15:37 (UTC-07:00)	

2. Mettez à jour le nom, la description, l'image de marque du portail, l'e-mail de contact du Support, l'accès à l'AWS IoT SiteWise assistant ou au service.
3. Lorsque vous avez terminé, sélectionnez Enregistrer les modifications.

Ajouter ou supprimer des administrateurs de portail

En quelques étapes, vous pouvez ajouter ou supprimer des utilisateurs en tant qu'administrateurs d'un portail. En fonction du service d'authentification utilisateur, choisissez l'une des options suivantes.

IAM Identity Center



Pour ajouter des administrateurs de portail

1. Sur la page des détails du portail, dans la section Administrateurs, choisissez Affecter des administrateurs.
2. Sur la page Affecter des administrateurs, sélectionnez les utilisateurs à ajouter au portail en tant qu'administrateurs.

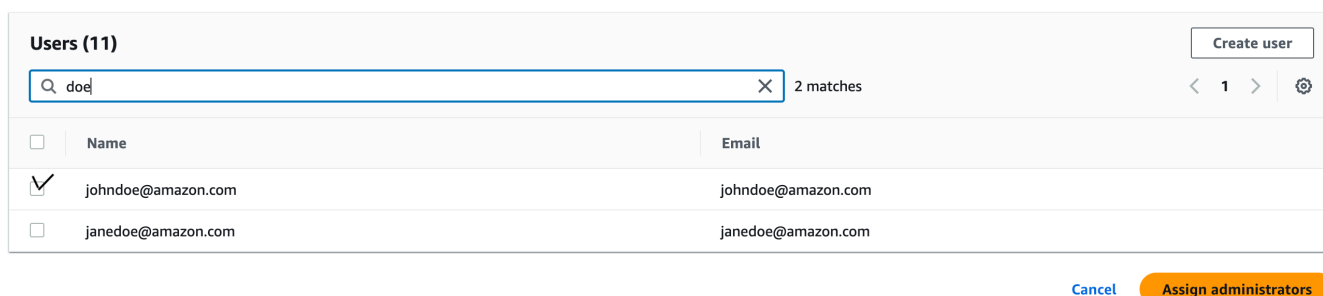
Note

Si vous utilisez IAM Identity Center comme banque d'identités et que vous êtes connecté à votre compte de AWS Organizations gestion, vous pouvez choisir Create user pour créer un utilisateur IAM Identity Center. IAM Identity Center envoie un e-mail au nouvel utilisateur pour qu'il définisse son mot de passe. Vous pouvez ensuite affecter l'utilisateur au portail en tant qu'administrateur. Pour plus d'informations, consultez la rubrique [Manage identities in IAM Identity Center](#).

3. Choisissez Attribuer des administrateurs.

Assign administrators [Info](#)

Choose portal administrators from the users list. Portal administrators grant access to specific industrial equipment data. [Learn more](#)



Pour supprimer des administrateurs de portail

- Dans la page de détails du portail, dans la section Portal administrators (Administrateurs du portail), cochez la case de chaque utilisateur à supprimer, puis choisissez Remove from portal (Supprimer du portail).

Note

Le champ Administrateurs (#) indique le nombre d'administrateurs du portail. Vous pouvez ajouter plusieurs administrateurs de portail pour gérer et travailler sur des projets.

Envoyer des invitations par e-mail aux administrateurs du portail

Vous pouvez envoyer des invitations par e-mail aux administrateurs du portail.

1. Sur la page des détails du portail, dans la section Administrateurs, cochez les cases correspondant aux administrateurs du portail.
2. Choisissez Envoyer des invitations. Votre client de messagerie s'ouvre et une invitation est renseignée dans le corps du message.

Vous pouvez personnaliser l'e-mail avant de l'envoyer aux administrateurs de votre portail.

Supprimer un portail

Vous pouvez supprimer un portail si vous l'avez créé à des fins de test ou en cas de doublon.

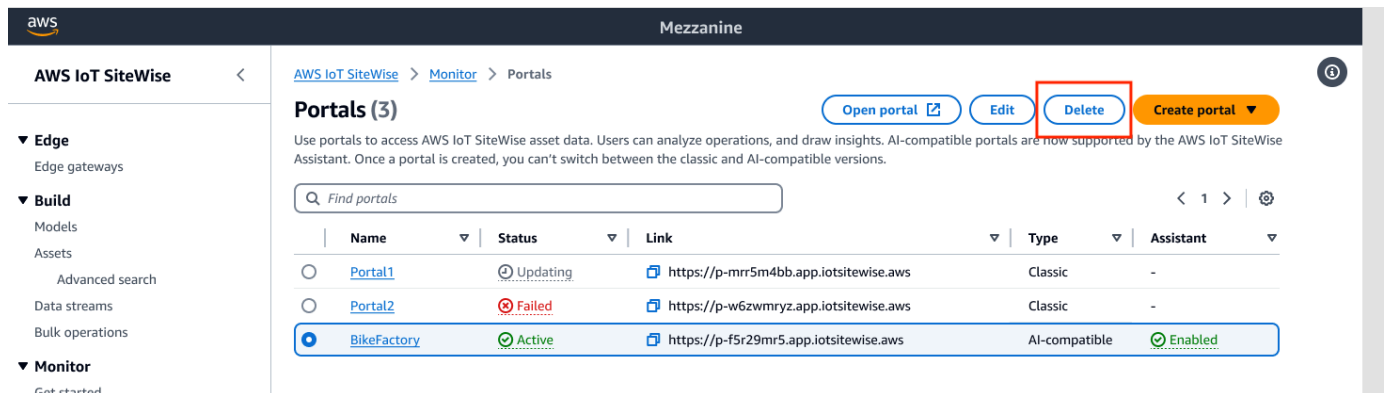
Note

Vous devez supprimer manuellement tous les tableaux de bord et projets du portail avant de pouvoir supprimer un portail.

1. Sur la page de détails du portail, choisissez Supprimer.

⚠ Important

Lorsque vous supprimez un portail, vous perdez tous les projets qu'il contient et tous les tableaux de bord de chaque projet. Cette action ne peut pas être annulée. Les données de vos ressources ne sont pas affectées.



2. Dans la boîte de dialogue Supprimer le portail, choisissez Supprimer les administrateurs et les utilisateurs.

Vous devez supprimer les administrateurs et les utilisateurs du portail avant de pouvoir supprimer ce dernier. Si votre portail ne dispose pas d'administrateurs ou d'utilisateurs, le bouton n'apparaît pas et vous pouvez passer à l'étape suivante.

3. Si vous êtes sûr de vouloir supprimer l'intégralité du portail, saisissez **confirm** dans le champ pour confirmer la suppression.
4. Sélectionnez Delete (Supprimer).

Créez des tableaux de bord avec AWS CLI

Lorsque vous définissez des visualisations (ou des widgets) dans des tableaux de bord à l'aide du AWS CLI, vous devez spécifier les informations suivantes dans le document `dashboardDefinition` JSON. Cette définition est un paramètre des [UpdateDashboard](#) opérations [CreateDashboard](#).

`displaySettings`

Les paramètres d'affichage avec les paramètres suivants :

- `numRows`— Nombre de lignes dans la mise en page du tableau de bord. Chaque ligne a la largeur de `CellSize`.
- `numColumns`— Nombre de colonnes dans la mise en page du tableau de bord. La largeur de chaque colonne est de `CellSize`.
- `cellSize`— (Facultatif) Taille d'une cellule dans le modèle en pixels. Il doit s'agir d'un nombre positif. La valeur par défaut est 10.
- `significantDigits`— (Facultatif) Nombre de chiffres significatifs à afficher dans le tableau de bord. La valeur par défaut est 4.

`querySettings`

Les informations de requête avec le paramètre suivant :

- `refreshRate`— (Facultatif) Vitesse d'actualisation des données en millisecondes. Accepte les valeurs suivantes : 1000, 5000, 10000, 60000, 300000.

`defaultViewport`

Si ce n'est pas le cas, la valeur par défaut est les cinq dernières minutes. Contient les paramètres suivants :

- `duration`— (Facultatif) Détermine jusqu'où il faut interroger les données en commençant par le présent.
- `start`— (Facultatif) Il est de type `Date`. La plage horaire de début pour interroger les données. Nécessite une `end date` spécifiée.
- `end`— (Facultatif) Il est de type `Date`. L'intervalle de temps de fin pour interroger les données. Nécessite une `start date` spécifiée.

`widgets`

Liste des structures de définition de widgets contenant les informations suivantes :

`type`

Type de widget. AWS IoT SiteWise fournit les types de widgets suivants :

- `xy-plot`— Un graphique linéaire ou un nuage de points selon la configuration.
- `bar-chart`— Un graphique à barres.
- `kpi-chart`— Un tableau des indicateurs de performance clés.
-

`status-timeline`— Un widget d'état qui permet de visualiser et de parcourir les données de séries chronologiques provenant d'une ou de plusieurs sources de données.

- `text`— Un widget de texte.
- `table`— Un widget de tableau.

`id`

Identifiant unique pour le widget.

`x`

Position horizontale du widget, en partant de la gauche du tableau de bord. Cette valeur fait référence à la position du widget dans la grille du tableau de bord.

`y`

Position verticale du widget, en commençant par le haut du tableau de bord. Cette valeur fait référence à la position du widget dans la grille du tableau de bord.

`z`

L'ordre relatif des widgets. Un widget de valeur Z plus grande est affiché devant le widget de valeur Z inférieure, s'ils se chevauchent.

`width`

Largeur du widget, exprimée en nombre de cellules sur le tableau de bord.

`height`

Hauteur du widget, exprimée en nombre de cellules sur le tableau de bord.

`properties`

Liste des propriétés du widget. Cela varie en fonction du type de widget. Consultez [IoT App Kit](#) pour plus de détails.

Exemple Exemple de définition de tableau de bord

L'exemple suivant définit un tableau de bord à partir d'une charge utile stockée dans un fichier JSON.

```
aws iotsitewise create-dashboard \  
  --project-id a1b2c3d4-5678-90ab-cdef-eeeeEXAMPLE \
```

```
--dashboard-name "Example Dashboard" \  
--dashboard-definition file://dashboard-definition.json
```

L'exemple JSON suivant pour `dashboard-definition.json` définit un tableau de bord avec les widgets de visualisation suivants :

```
{  
  "displaySettings": {  
    "numColumns": 200,  
    "numRows": 1000,  
    "cellSize": 20,  
    "significantDigits": 4  
  },  
  "widgets": [{  
    "id": "0t73JcxUoc6oEXAMPLE",  
    "type": "xy-plot",  
    "width": 33,  
    "height": 20,  
    "x": 0,  
    "y": 0,  
    "z": 0,  
    "properties": {  
      "aggregationType": "AVERAGE",  
      "queryConfig": {  
        "source": "iotsitewise",  
        "query": {  
          "assets": [{  
            "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",  
            "properties": [{  
              "propertyId": "97cc61f4-57a4-4c5f-a82c-EXAMPLE",  
              "refId": "692ce941-f3d9-4074-a297-EXAMPLE",  
              "aggregationType": "AVERAGE",  
              "color": "#7d2105",  
              "resolution": "1m"  
            }]  
          }]  
        },  
        "properties": [],  
        "assetModels": [],  
        "alarms": [],  
        "alarmModels": []  
      }  
    }  
  },  
}
```

```

        "line": {
            "connectionStyle": "linear",
            "style": "solid"
        },
        "symbol": {
            "style": "filled-circle"
        },
        "axis": {
            "yVisible": true,
            "xVisible": true
        },
        "legend": {
            "visible": true,
            "position": "right",
            "width": "30%",
            "height": "30%",
            "visibleContent": {
                "unit": true,
                "asset": true,
                "latestValue": true,
                "latestAlarmStateValue": true,
                "maxValue": false,
                "minValue": false
            }
        }
    }, {
        "id": "fto7rF40Ny1EXAMPLE-G",
        "type": "bar-chart",
        "width": 33,
        "height": 20,
        "x": 0,
        "y": 20,
        "z": 0,
        "properties": {
            "aggregationType": "AVERAGE",
            "queryConfig": {
                "source": "iotsitewise",
                "query": {
                    "assets": [{
                        "assetId": "97c97abf-e883-47bb-a3f4-EXAMPLE",
                        "properties": [{
                            "propertyId": "c84ca8f3-3dea-478a-afec-EXAMPLE",
                            "aggregationType": "AVERAGE",

```

```

        "refId": "2960b958-2034-4d6e-bcc2-EXAMPLE"
      }
    ]
  },
  "properties": [],
  "assetModels": [],
  "alarms": [],
  "alarmModels": [],
  "requestSettings": {
    "aggregation": "AVERAGE"
  }
},
"axis": {
  "showX": true,
  "showY": true
},
"styleSettings": {
  "2960b958-2034-4d6e-bcc2-360f1f02e505": {
    "color": "#7d2105"
  }
}
}],
"querySettings": {
  "refreshRate": 5000
}
}

```

Connexion au portail

Connexion utilisateur

1. Dans votre navigateur, saisissez l'URL de l'application.
2. Entrez votre nom d'utilisateur et votre mot de passe, puis cliquez sur le bouton Se connecter.
3. Vous êtes maintenant connecté à l'application.



Création d'un projet

Création d'un projet

1. Un projet est créé de deux manières :
 - a. Sur la page d'accueil, dans la section Bienvenue, sous Démarrage rapide, choisissez Créer un projet.
 - b. Dans le volet de navigation de gauche, sélectionnez Projets. Choisissez Créer dans le coin supérieur droit pour créer un projet.
2. Dans la section Créer des projets, entrez un nom de projet et donnez une description facultative.
3. Sélectionnez Create (Créer).

Met à jour un projet.

Modifier le projet

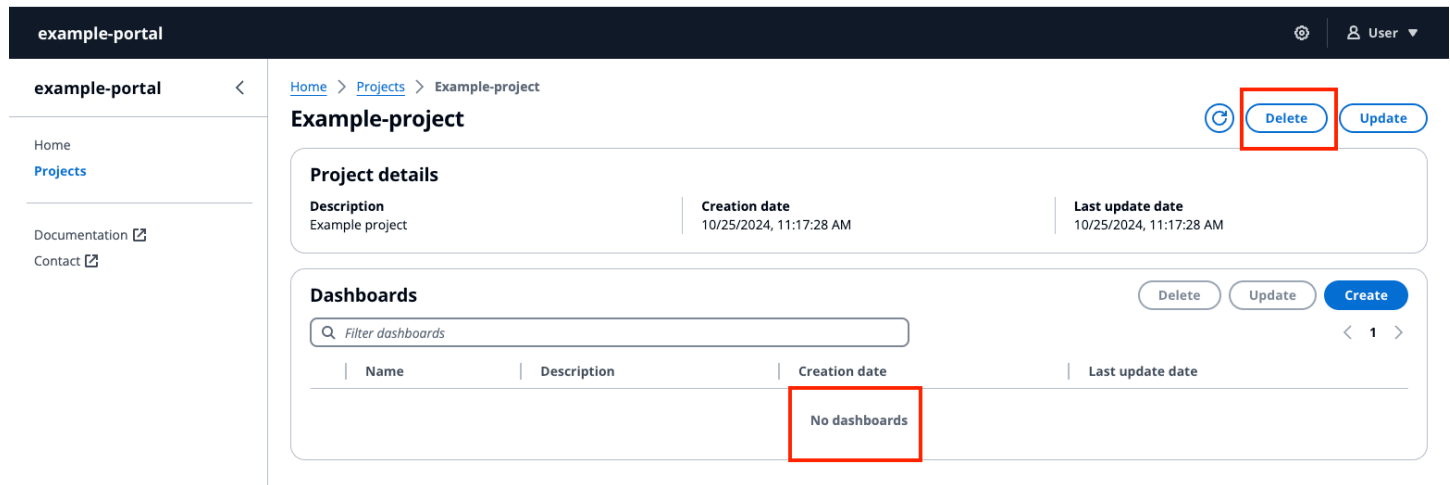
1. Cliquez sur le bouton Mettre à jour dans le coin supérieur droit de la page du projet pour modifier les détails du projet.
2. Modifiez le nom du projet en modifiant le nom du projet.
3. Modifiez la description du projet en modifiant les détails de la description.
4. Sélectionnez Mettre à jour pour enregistrer vos modifications.

The screenshot shows the 'Example-project' page in the AWS IoT SiteWise portal. The page has a dark header with 'example-portal' and a user profile. A sidebar on the left contains links for 'Home', 'Projects', 'Documentation', and 'Contact'. The main content area is titled 'Example-project' and includes a breadcrumb trail 'Home > Projects > Example-project'. In the top right corner of the main content area, there are three buttons: a refresh icon, a 'Delete' button, and an 'Update' button, which is highlighted with a red rectangular box. Below the buttons, the 'Project details' section shows the 'Description' as 'Example project', the 'Creation date' as '10/25/2024, 11:17:28 AM', and the 'Last update date' as '10/25/2024, 11:17:28 AM'. The 'Dashboards' section features a search bar with the placeholder 'Filter dashboards' and a table with one dashboard entry. The table has columns for 'Name', 'Description', 'Creation date', and 'Last update date'. The entry is 'example dashboard' with description 'dashboard' and creation/update dates of '10/25/2024, 11:17:52 AM'. Above the table, there are 'Delete', 'Update', and 'Create' buttons. A pagination indicator shows '< 1 >'.

Supprime un projet.

Supprimer le projet

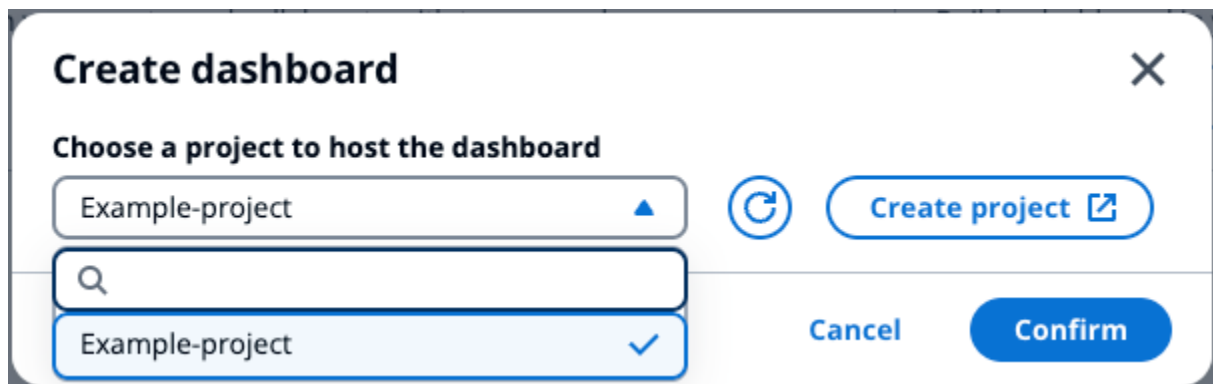
1. Vous ne pouvez supprimer le projet qu'une fois que tous les tableaux de bord du projet ont été supprimés.
2. Cliquez sur le bouton Supprimer en haut à droite de la page du projet.
3. Confirmez à nouveau que vous souhaitez supprimer le projet.
4. Sélectionnez Supprimer pour supprimer le projet.



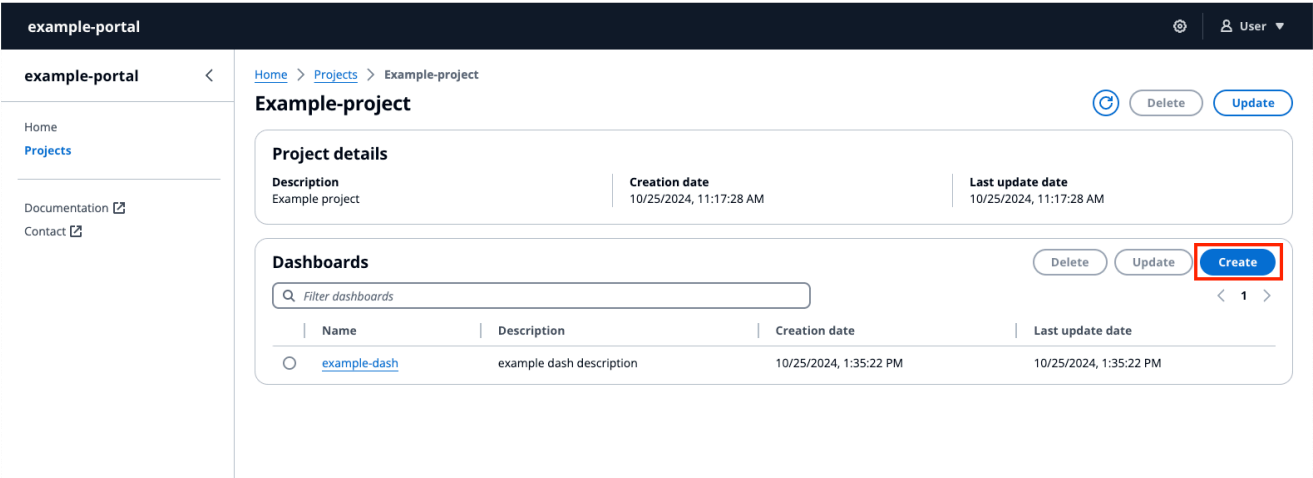
Création d'un tableau de bord

Création d'un tableau de bord

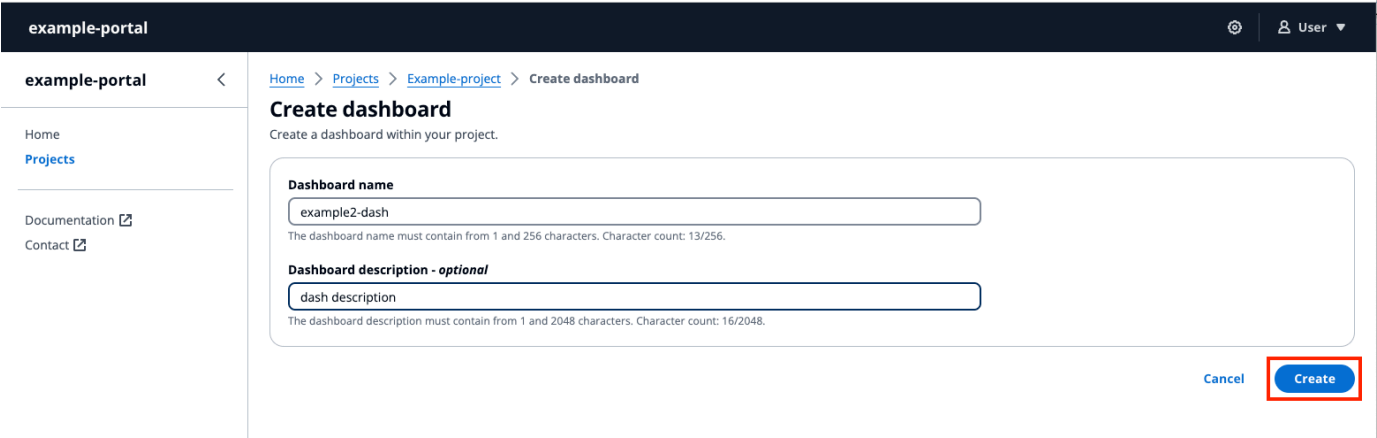
1. Créez un tableau de bord de deux manières :
 - a. Créez un tableau de bord depuis Créer un tableau de bord sur la page d'accueil.
 - i. Pour créer le tableau de bord dans un projet existant, choisissez un nom de projet dans le menu déroulant de la section Choisissez un projet pour héberger le tableau de bord.
 - ii. Si vous n'avez pas de projet, choisissez Créer un projet, puis Confirmer.



- b. Créez un tableau de bord à partir d'un projet dans la section Projets, sous Tableaux de bord.



2. Sélectionnez Créer dans le coin supérieur droit.
3. Entrez un nom de tableau de bord et donnez une description facultative du tableau de bord.
4. Sélectionnez Créer.



5. Configurez le tableau de bord que vous venez de créer.

Mise à jour d'un tableau de bord

La section Tableaux de bord répertorie les tableaux de bord du projet. Sélectionnez un tableau de bord dans la liste.

Mise à jour d'un tableau de bord

1. Sélectionnez un tableau de bord à mettre à jour.

The screenshot shows the 'Example-project' page in the AWS IoT SiteWise portal. At the top, there's a navigation bar with 'example-portal' and a user profile. Below the navigation bar, the breadcrumb is 'Home > Projects > Example-project'. The main heading is 'Example-project'. To the right of the heading are buttons for 'Delete' and 'Update'. Below the heading is a 'Project details' section with three columns: 'Description' (Example project), 'Creation date' (10/25/2024, 11:17:28 AM), and 'Last update date' (10/25/2024, 11:17:28 AM). Below this is a 'Dashboards' section. It has a search bar 'Filter dashboards' and buttons for 'Delete', 'Update' (highlighted with a red box), and 'Create'. Below the buttons is a table with columns: 'Name', 'Description', 'Creation date', and 'Last update date'. The table has one row with the dashboard 'example-dash'.

Name	Description	Creation date	Last update date
example-dash	example dash description	10/25/2024, 1:35:22 PM	10/25/2024, 1:35:22 PM

2. Mettez à jour le nom du tableau de bord et éventuellement la description du tableau de bord. Sélectionnez Mettre à jour pour enregistrer les modifications.

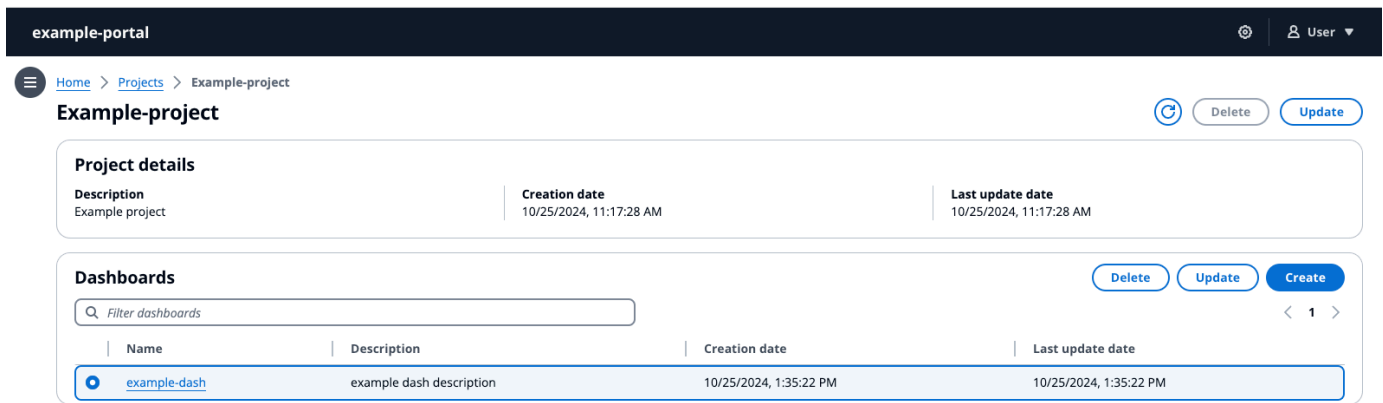
The screenshot shows the 'Update dashboard: example-dash' form. The breadcrumb is 'Home > Projects > example-dash > Update'. The main heading is 'Update dashboard: example-dash'. Below the heading is a form with two input fields: 'Dashboard name' (example-update-dash-name) and 'Dashboard description - optional' (update dash description). Below the 'Dashboard name' field is a note: 'The dashboard name must contain from 1 and 256 characters. Character count: 24/256.' Below the 'Dashboard description - optional' field is a note: 'The dashboard description must contain from 1 and 2048 characters. Character count: 23/2048.' At the bottom right of the form are buttons for 'Cancel' and 'Update' (highlighted with a red box).

Suppression d'un tableau de bord

La section Tableaux de bord répertorie les tableaux de bord du projet. Sélectionnez un tableau de bord dans la liste.

Suppression d'un tableau de bord

1. Sélectionnez le tableau de bord à supprimer.



2. Sélectionnez Supprimer pour supprimer le tableau de bord. Cela ne peut pas être annulé.

Configuration du tableau de bord

La section Tableaux de bord répertorie les tableaux de bord du projet. Sélectionnez un tableau de bord dans la liste. Le mode Édition vous permet de configurer votre tableau de bord en ajoutant des widgets et en les configurant. Le bouton Aperçu vous permet de visualiser vos modifications.



Étapes pour configurer votre tableau de bord :

- Faites glisser et déposez différents types de widgets de données sur le canevas du tableau de bord pour visualiser les données.
- Ajoutez des données aux widgets souhaités, à partir de l'explorateur de ressources sur la gauche. L'explorateur de ressources comprend les sections Ressources modélisées, Non modélisées et Dynamiques. Effectuez une recherche par nom d'actif ou par nom de propriété. Sélectionnez la propriété à ajouter, puis cliquez sur Ajouter.
- Ajustez la mise en page et le style en modifiant les configurations des widgets. Configurez les composants, y compris le titre, les seuils et les autres caractéristiques de configuration.
- Configurez la plage de temps sur laquelle les données sont affichées.

- Choisissez la plage de temps sur laquelle les données sont affichées. Choisissez une plage de temps et un taux de rafraîchissement dans le coin supérieur droit, puis personnalisez la plage. Choisissez la fréquence à laquelle les données doivent être actualisées dans le menu.
- Sélectionnez la plage de temps sur un widget en utilisant la molette de défilement de votre souris trackball ou en cliquant avec le bouton droit de la souris. Cela déplace la plage de temps d'affichage.
- Choisissez Save (Enregistrer).

Rubriques

- [Explorateur de ressources](#)
- [Widgets](#)
- [Configuration des widgets](#)
- [Utiliser des widgets](#)
- [Alarmes dans les widgets](#)
- [AWS IoT SiteWise Utilisation de l'assistant dans les widgets](#)
- [Exemples de questions à poser à l' AWS IoT SiteWise assistant](#)

Explorateur de ressources

Cette section décrit les actifs modélisés, non modélisés et dynamiques. Choisissez des actifs parmi les trois, ajoutez-les à vos widgets et visualisez-les.

Rubriques

- [Modélisé](#)
- [Non modélisé](#)
- [Des actifs dynamiques](#)

Modélisé

Cette section décrit le processus de sélection et de visualisation des actifs modélisés.

Sélection des actifs

Les actifs peuvent être interrogés comme suit :

- Recherchez le nom d'un actif. Utilisez un joker. * Par exemple, Wind* renvoie les noms des actifs qui commencent par le texteWind. Vous devez procéder AWS IoT TwinMakerà [l'intégration](#) pour bénéficier de cette fonctionnalité.
- Tous les actifs sont répertoriés par défaut.

À partir des actifs répertoriés, filtrez par nom, description, ID ou ID de modèle d'actif. Sélectionnez un actif pour répertorier ses propriétés (flux de données) et ses alarmes.

Sélection du flux de données

Les flux de données sont répertoriés sous le menu Flux de données. Filtrez les flux de données répertoriés par Métadonnées de [propriété](#) dans la <https://docs.aws.amazon.com/iot-sitewise/latest/APIReference>. Sélectionnez un ou plusieurs flux de données en fonction du widget sélectionné.

- KPI et Gauge ne prennent en charge qu'un seul flux de données.
- Les autres widgets prennent en charge plusieurs flux de données avec sélection multiple.

Sélection de l'alarme

AWS IoT SiteWise les alarmes sont répertoriées sous le menu Alarm Data Streams. Filtrez les flux de données d'alarme répertoriés par métadonnées d'alarme. le nom, la propriété d'entrée et l'ID du modèle composite sont des métadonnées utilisées pour le filtrage. Sélectionnez un ou plusieurs flux de données en fonction du widget sélectionné.

- KPI et Gauge ne prennent en charge qu'une seule alarme.
- Les autres widgets prennent en charge plusieurs alarmes avec sélection multiple.

Visualisation des actifs modélisés

1. Faites glisser le widget vers le canevas. Sélectionnez les propriétés de chaque panneau de widgets pour créer un tableau de bord.
2. L'option Filtrer filtre les actifs pour choisir l'actif à visualiser. Le filtrage s'effectue par texte, propriété ou valeur. Le filtrage concerne les ressources chargées dans le navigateur, et non le filtrage principal.
3. Effectuez une recherche pour répertorier une ressource à ajouter à votre widget.

4. Ajoutez l'actif au widget dans le canevas.
5. Choisissez Réinitialiser pour sélectionner un autre actif ou apporter des modifications à l'actif choisi.
6. Enregistrez le tableau de bord. En mode aperçu, choisissez différentes ressources dans le menu déroulant pour surveiller les propriétés de chaque ressource sans reconstruire les panneaux de données.

 Note

La roue des paramètres de configuration sur le côté droit affiche les préférences que l'utilisateur peut choisir, telles que le format de page, les premières colonnes autocollantes, les dernières colonnes autocollantes et les préférences de colonne. Personnalisez vos préférences et choisissez Confirmer pour appliquer les modifications.

Modeled

Unmodeled

Dynamic assets

Assets (1)

Browse through your asset hierarchy and select an asset to view its associated data streams.

Root

Search

Q

Search for resources

Search

Filter

Q

Filter assets by text, property, or value

< 1 > ⚙

Name

Description

Demo Wind Farm Asset

Asset properties (8)

Select a modeled datastream to add to a selected widget

Filter

Q

Filter asset properties by text, property, or value

< 1 > ⚙

☐	Name	Unit	Data ty...	Latest ...	Latest value time
☐	Total Aver...		DOUBLE	37478.2303	2024-10-01 09:50:00 p.m.
☐	Total Aver...		DOUBLE	6.0000	2022-10-26 03:42:43 p.m.
☐	Total Aver...		DOUBLE	555.0000	2022-10-13 11:59:49 p.m.
☐	Code		INTEGER	300.0000	2022-10-13 10:59:28 p.m.
☐	Reliability ...		STRING	Mary Major	2022-10-13 10:59:28 p.m.
☐	Location		STRING	Renton	2022-10-13 10:59:28 p.m.
☐	Total Over...		DOUBLE	900.0000	2024-10-01 09:50:00 p.m.
☐	recipient a...		STRING	54a88418-...	2022-10-26 03:42:43 p.m.

Non modélisé

Cette section décrit la recherche de flux de données non modélisés et leur ajout aux widgets pour les visualiser.

Visualisation de flux de données non modélisés

1. Faites glisser le widget vers le canevas. Sélectionnez les propriétés de chaque panneau de widgets pour créer un tableau de bord.
2. Les flux de données non modélisés sont répertoriés dans la section Séries chronologiques. Leurs propriétés sont personnalisables.
3. L'option Filtrer filtre les flux de données à visualiser. Le filtrage concerne les flux de données chargés dans le navigateur, et non le filtrage principal.
4. Ajoutez le flux de données au widget dans le canevas.
5. Choisissez Réinitialiser pour désélectionner le flux de données.
6. Enregistrez le tableau de bord. En mode aperçu, choisissez différentes ressources dans le menu déroulant pour surveiller les propriétés de chaque ressource sans reconstruire les panneaux de données.

Note

La roue des paramètres de configuration sur le côté droit affiche les préférences que l'utilisateur peut choisir, telles que le format de page, les premières colonnes autocollantes, les dernières colonnes autocollantes et les préférences de colonne. Personnalisez vos préférences et choisissez Confirmer pour appliquer les modifications.

Modeled | **Unmodeled** | Dynamic assets

Time series (1)
Select a unmodeled datastream to add to a selected widget

Filter < 1 > 

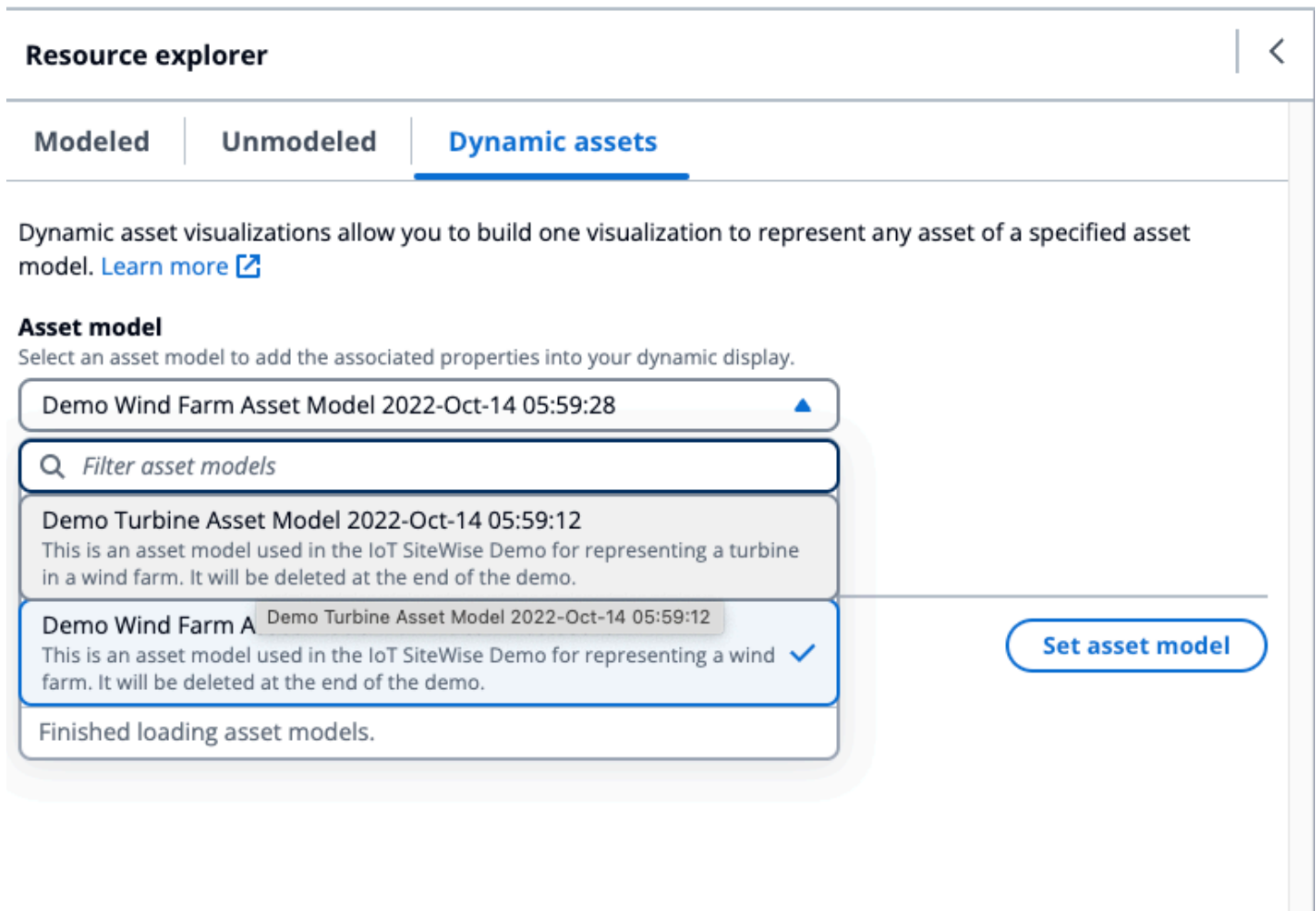
☒	Alias ▾	ID ▾	Data type ▾	Latest val... ▾	Latest valu... ▾
☒	DemoDisass...	33e78bb9-39...	DOUBLE	3.0283	2024-10-01 0...

Des actifs dynamiques

Le nouveau SiteWise moniteur permet aux clients de changer d'actifs de manière dynamique pour un modèle d'actif sélectionné. Vous pouvez visualiser les propriétés de différents actifs en les sélectionnant dans un menu déroulant.

Visualisation dynamique des actifs

1. Choisissez l'onglet Ressources dynamiques dans l'explorateur de ressources.
2. Sélectionnez un modèle d'actif pour lequel répertorier les actifs dans le menu déroulant.
3. Sélectionnez l'actif par défaut dans le menu déroulant.
4. Choisissez Définir le modèle d'actif pour sélectionner le modèle d'actif.
5. Enregistrez le tableau de bord. En mode aperçu, choisissez différentes ressources dans le menu déroulant pour surveiller les propriétés de chaque ressource, sans reconstruire les panneaux de données.



Widgets

Les widgets prennent en charge un large éventail de fonctionnalités, notamment les alarmes, la diffusion en direct haute performance et la synchronisation fluide avec les autres composants de l'IoT App Kit. Le tableau de bord prend en charge les widgets suivants :

- **Ligne** — Le widget Ligne est un widget de visualisation qui affiche les tendances et les changements au fil du temps. Il consiste en une série de points de données, chacun représenté par un point ou un marqueur, reliés par des segments de ligne droite pour créer un graphique linéaire. Il prend en charge un large éventail de fonctionnalités, notamment les alarmes, les seuils, la diffusion en direct haute performance et une synchronisation fluide avec les autres composants de l'IoT App Kit. Ce widget est personnalisable pour communiquer des données complexes de manière claire et concise.
- **Graphique à barres** — Le graphique à barres est un puissant outil de visualisation qui affiche des données chronologiques. Il prend en charge un large éventail de fonctionnalités, notamment les

alarmes, la diffusion en direct haute performance et une synchronisation fluide avec les autres composants de l'IoT App Kit.

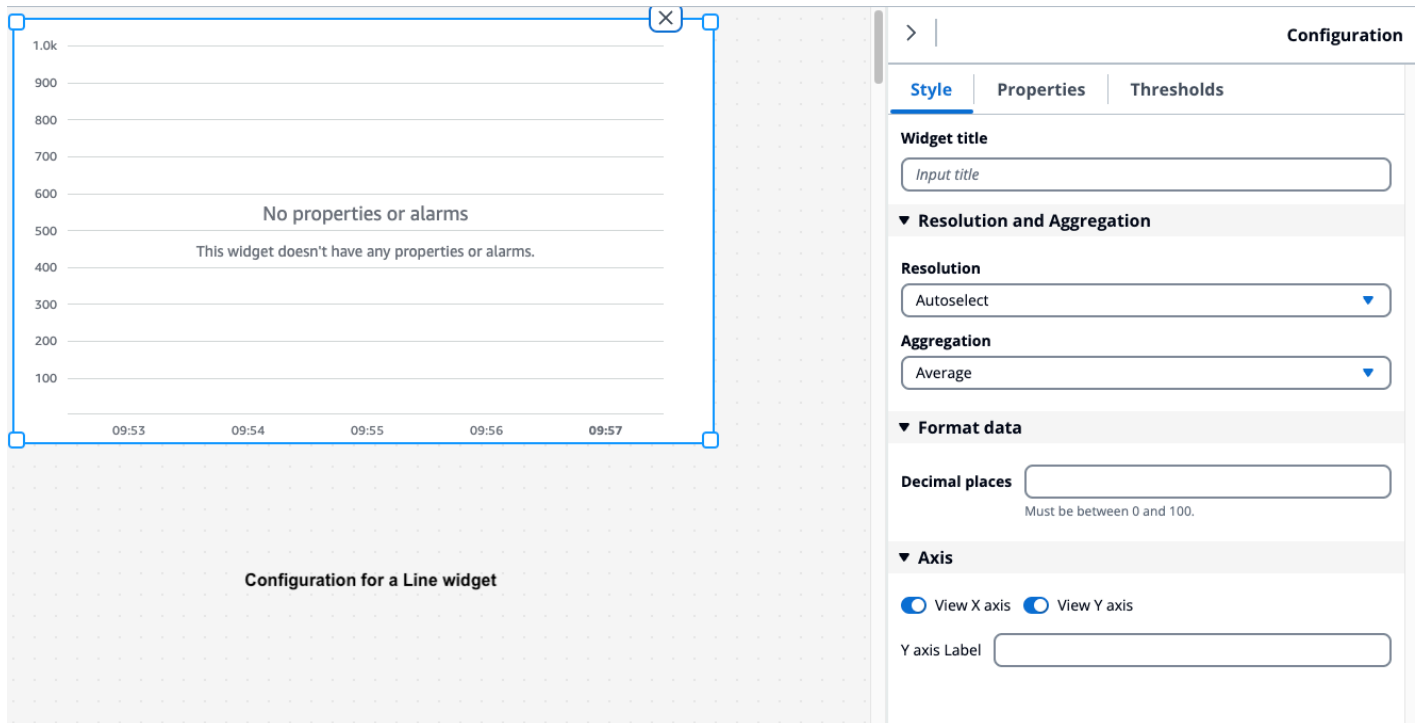
- **Chronologie** — Le widget Chronologie permet de visualiser et de parcourir les données de séries chronologiques à partir de sources de données. Il est unique car les valeurs des flux de données sont affichées sous forme de couleurs distinctes sur la chronologie. Il prend en charge un ensemble complet de fonctionnalités, notamment des alarmes, une diffusion en direct haute performance et une synchronisation fluide entre les autres composants de l'IoT App Kit. Il est préférable de l'utiliser pour afficher des types de données non numériques/
- **KPI** — Le composant Indicateur de performance clé (KPI) fournit une représentation compacte d'une vue d'ensemble des propriétés de vos actifs. Il prend en charge les alarmes et les seuils. Cette présentation fournit des informations essentielles sur les performances globales de vos appareils, équipements et processus. Le KPI ne prend en charge qu'un seul flux de données ou une seule alarme, et non plusieurs flux de données.
- **Jauge** — Le composant Gauge fournit une représentation compacte d'une vue d'ensemble des propriétés de vos actifs. Il est utilisé pour visualiser des informations critiques sur les performances globales de vos appareils, équipements ou processus. Il est fonctionnellement identique au KPI, mais visuellement différent. La jauge affiche la valeur du flux de données, le seuil et la plage de valeurs. Vous pouvez interagir avec les AWS IoT données d'une ou de plusieurs sources de données avec Gauge.
- **Tableau** — Le composant Table fournit un formulaire compact permettant de visualiser un ou plusieurs flux de données provenant d'une ou de plusieurs sources de données de séries chronologiques. Il affiche les actifs avec la propriété, la dernière valeur et l'unité sous forme de tableau. Supporte les AWS IoT SiteWise alarmes.
- **Texte** — Le widget Texte permet d'écrire du texte avec différentes couleurs et polices. Vous pouvez créer un lien en associant un texte à une URL. Les champs Propriétés et Seuils ne sont pas activés pour ce widget.



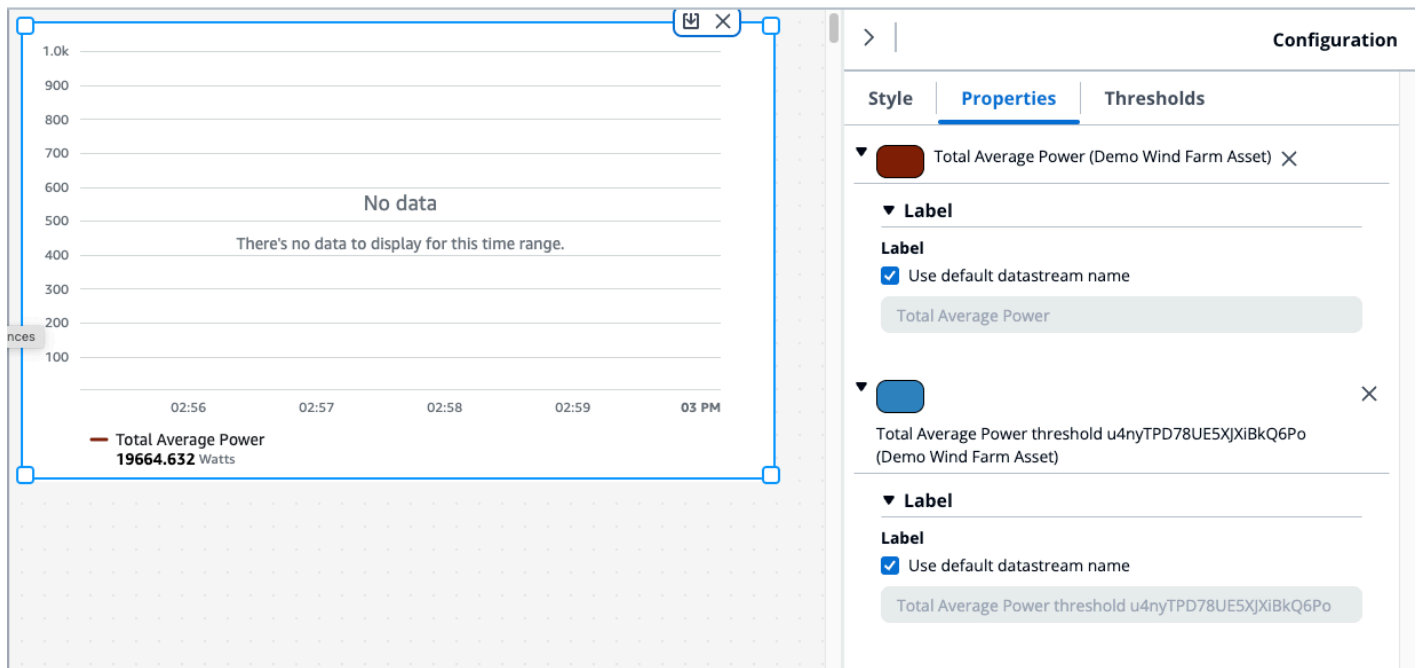
Configuration des widgets

Une fois le widget ajouté au tableau de bord, vous pouvez le configurer en choisissant l'icône de configuration dans le panneau de droite.

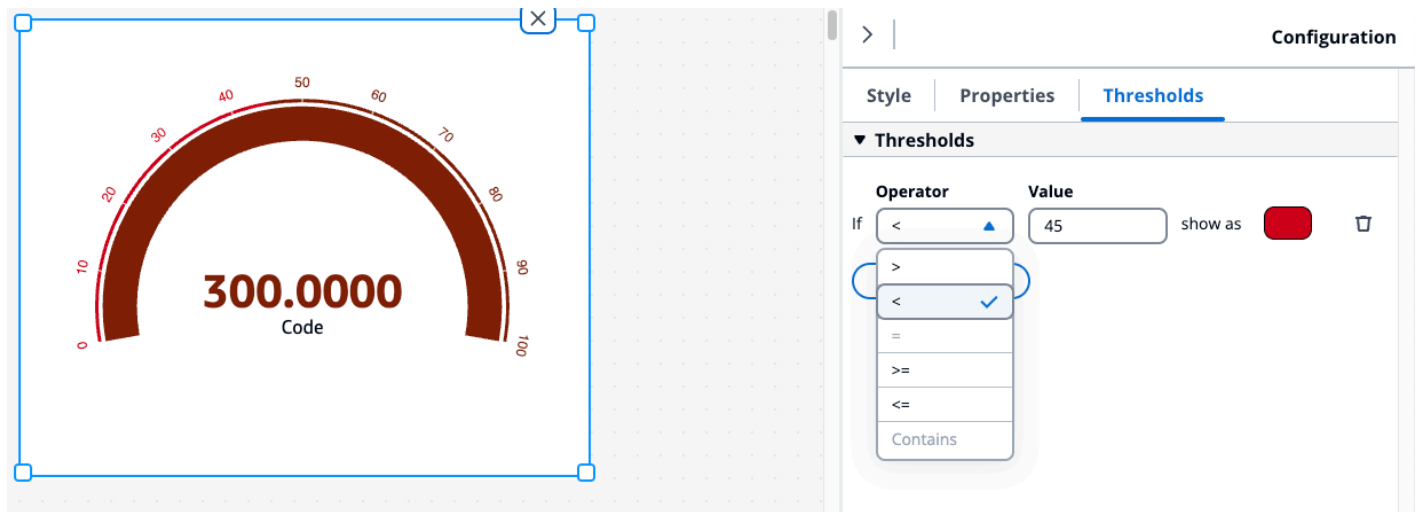
- **Style** — Ajoutez un titre dans le titre du widget. Les différents widgets ont des configurations différentes. Quelques exemples sont listés ci-dessous.
 - **Widget de barre** :
 - **Résolution et agrégation** : définissez les valeurs de résolution et d'agrégation ici.
 - **Formater les données** — Définissez le nombre de décimales à afficher au nombre de décimales.
 - **Style d'affichage** : sélectionnez les valeurs à afficher.
 - **Axe** — Choisissez d'afficher l'axe.
 - **Widget linéaire** :
 - **Résolution et agrégation** : définissez les valeurs de résolution et d'agrégation ici.
 - **Formater les données** — Définissez le nombre de décimales à afficher au nombre de décimales.
 - **Axe Y** — Ajoutez une étiquette et des valeurs min et max.
 - **Style du widget** : sélectionnez le type de ligne, le style de ligne, l'épaisseur de ligne et les valeurs de forme des points de données.
 - **Légende** — Choisissez l'alignement et l'affichage.
 - **Widget de jauge** :
 - **Résolution et agrégation** : définissez les valeurs de résolution et d'agrégation ici.
 - **Formater les données** — Définissez le nombre de décimales à afficher au nombre de décimales.
 - **Style d'affichage** : sélectionnez les valeurs à afficher.
 - **Axe Y** — Ajoutez une étiquette et des valeurs min et max.
 - **Polices** : sélectionnez les valeurs de taille de police, de taille de police d'unité et de taille de police d'étiquette.



- **Propriétés** — Toutes les propriétés des widgets sont répertoriées dans cette section. Les différents widgets ont des propriétés différentes. Quelques exemples sont listés ci-dessous.
- **Widget linéaire :**
 - **Label** — Choisissez d'utiliser le nom du flux de données par défaut ou d'en attribuer un nouveau.
 - **Style** — Définissez le type de ligne, le style de ligne selon le nombre de décimales à afficher.
 - **Axe Y** — Sélectionnez les valeurs selon le style par défaut, affichez les commandes de l'axe Y et définissez les valeurs min et max.
- **Widget de tableau :**
 - **Label** — Choisissez d'utiliser le nom du flux de données par défaut ou d'en attribuer un nouveau.
- **Widget de tableau :**
 - **Label** — Choisissez d'utiliser le nom du flux de données par défaut ou d'en attribuer un nouveau.



- Seuils — Ajoutez un seuil pour un widget. Les différents widgets ont des configurations différentes. Quelques exemples sont listés ci-dessous.
- Widget graphique à barres :
 - Choisissez Ajouter un seuil à ajouter au widget.
 - Choisissez Opérateur et attribuez une valeur au seuil. Personnalisez le seuil avec une couleur de la palette de couleurs.
 - Vous pouvez choisir d'appliquer le seuil à toutes les données.
- Widget linéaire :
 - Choisissez Ajouter un seuil à ajouter au widget.
 - Choisissez Opérateur et attribuez une valeur au seuil. Personnalisez le seuil avec une couleur de la palette de couleurs.
 - Choisissez comment afficher les seuils dans le menu déroulant.
- Widget de jauge :
 - Choisissez Ajouter un seuil à ajouter au widget.
 - Choisissez Opérateur et attribuez une valeur au seuil. Personnalisez le seuil avec une couleur de la palette de couleurs.



Utiliser des widgets

Vous pouvez utiliser les widgets du tableau de bord individuellement ou en les sélectionnant plusieurs fois.

Modifier les widgets dans le tableau de bord

Choisissez un seul widget et modifiez-le. Pour modifier plusieurs widgets dans le tableau de bord, cliquez avec le bouton gauche de la touche Shift et sélectionnez tous les widgets du tableau de bord. Une fois sélectionné, les utilisateurs peuvent ajouter de nouveaux flux de données et modifier le titre du widget dans les paramètres de configuration du style. Le titre de tous les widgets du tableau de bord est modifié.

Cliquez avec le bouton droit sur le canevas, puis procédez comme suit :

- Copier — Ajoutez une copie du widget au canevas.
- Supprimer — Supprimez le widget.
- Mettre au premier plan : place le widget sélectionné au premier plan du canevas.
- Envoyer à l'arrière : envoie le widget sélectionné au dos du canevas.

Redimensionner les widgets

Redimensionnez les widgets individuellement ou en groupe en les sélectionnant plusieurs fois dans le tableau de bord.

Pour modifier la taille des widgets :

- Pour modifier la taille d'un seul widget, sélectionnez-le et faites-le glisser dans un coin pour modifier sa taille.
- Pour modifier la taille de plusieurs widgets, sélectionnez-les en appuyant sur Shift + clic gauche, puis faites-les glisser dans un coin pour modifier leur taille.

Supprimer des widgets dans le tableau de bord

Supprimez les widgets individuellement ou en groupe en les sélectionnant plusieurs fois dans le tableau de bord.

Pour supprimer des widgets :

- Pour supprimer un seul widget, sélectionnez-le, cliquez avec le bouton droit de la souris et choisissez Supprimer. Vous pouvez également sélectionner et cliquer sur X dans le coin supérieur droit pour supprimer le widget.
- Pour supprimer plusieurs widgets, sélectionnez-les en appuyant sur Shift+ en cliquant avec le bouton gauche de la souris, puis en cliquant avec le bouton droit de la souris et en choisissant Supprimer.

Alarmes dans les widgets

Les alarmes vous alertent, vous et votre équipe, lorsque l'équipement ou les processus ne fonctionnent pas de manière optimale. Les performances optimales d'une machine ou d'un procédé signifient que les valeurs de certaines métriques doivent se situer dans une plage de limites élevées et basses. Lorsque ces paramètres se situent en dehors de leur plage de fonctionnement, les opérateurs de l'équipement doivent être avertis afin qu'ils puissent résoudre le problème. Les alarmes vous aident à identifier rapidement les problèmes et à avertir les opérateurs afin d'optimiser les performances de votre équipement et de vos processus.

Vous pouvez trouver une alarme associée à un actif dans l'onglet Modélisé de l'explorateur de ressources.

- Recherchez et sélectionnez un actif.
- Faites défiler le tableau des flux de données vers le bas jusqu'à la section Alarm Data Streams et développez-la.
- Sélectionnez une alarme dans le tableau des alarmes, puis cliquez sur Ajouter.

Rubriques

- [Alarmes dans différents widgets](#)

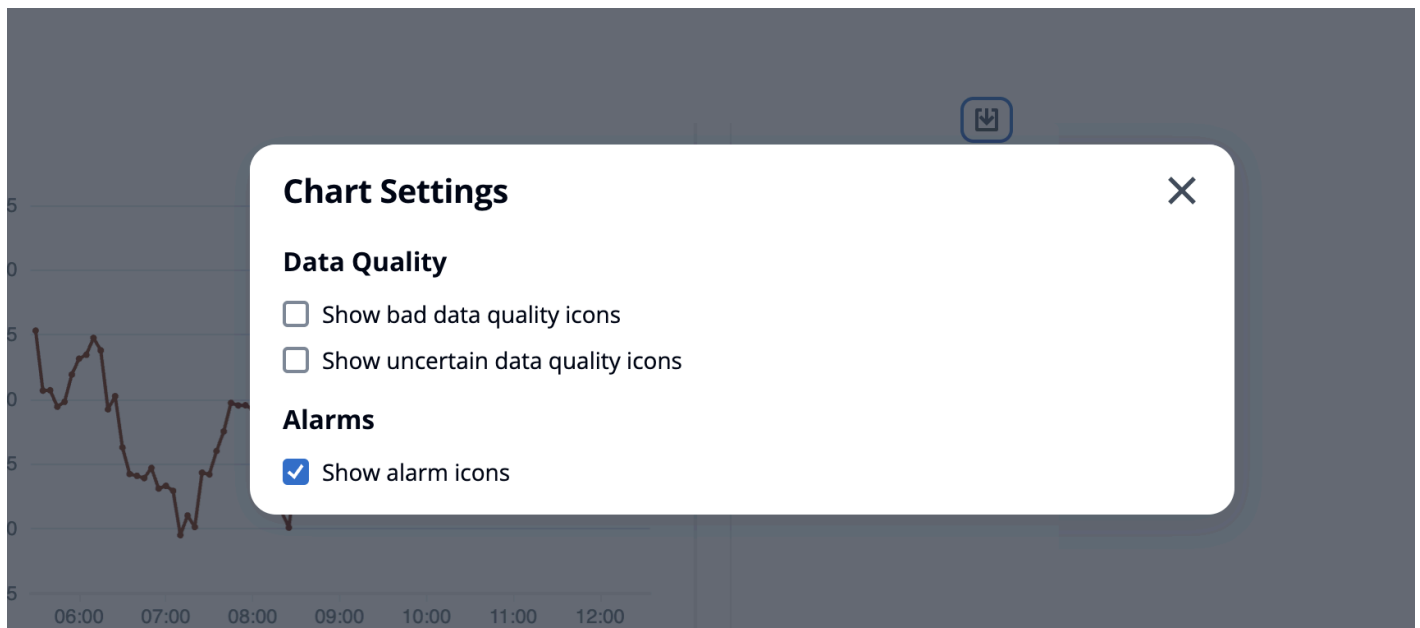
Alarmes dans différents widgets

Pour tous les widgets :

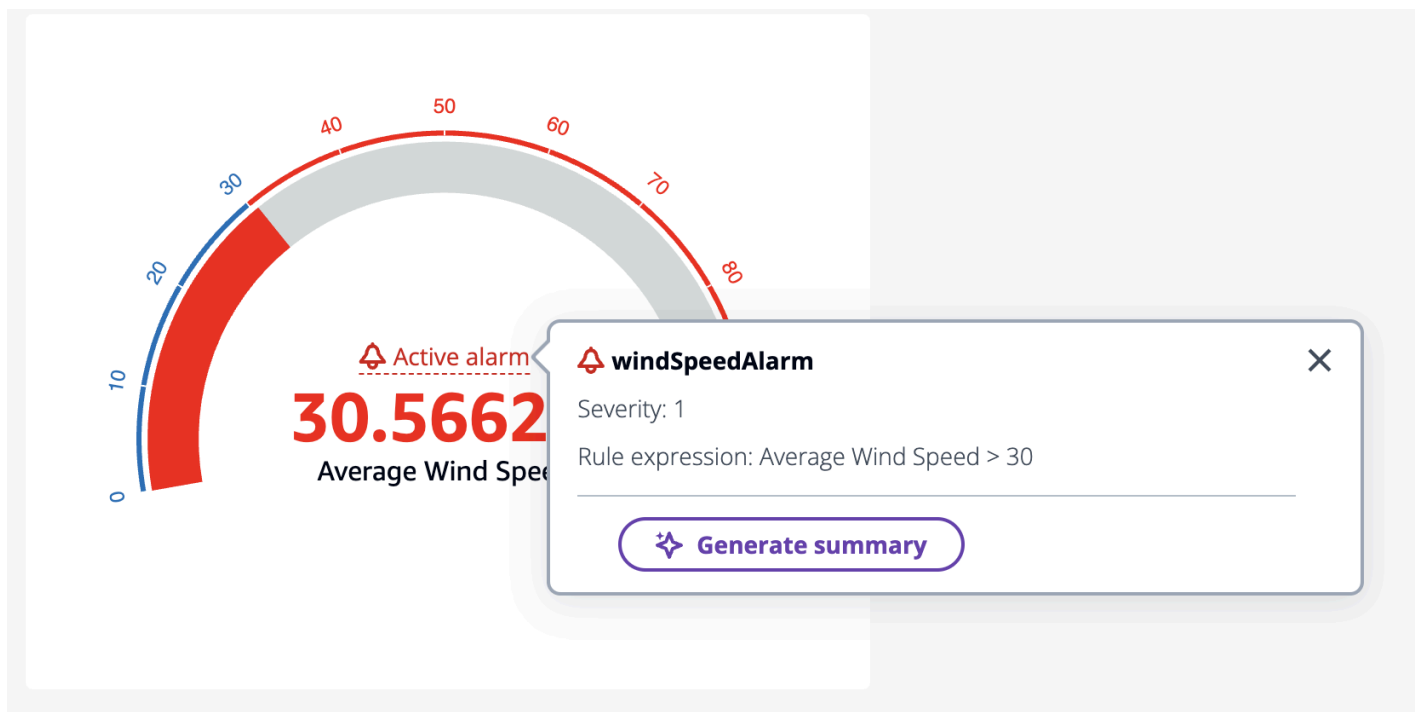
- Les paramètres des propriétés du flux de données dépendent du type de propriété ajouté à un widget. Les propriétés du flux de données sont entièrement prises en charge, tandis que les propriétés des alarmes ne permettent actuellement pas de configurer les paramètres de propriétés.
- Si vous ajoutez un flux de données d'alarme, le flux de données de propriété d'entrée associé est également ajouté au graphique. Si vous supprimez le flux de données d'alarme, sa propriété d'entrée est également supprimée.
- Pour contrôler individuellement le flux de données de propriétés d'entrée d'une alarme, vous devez les ajouter séparément.

Les exemples ci-dessous montrent comment certains widgets utilisent les alarmes.

- Graphique linéaire
 - L'alarme et son flux de données de propriétés d'entrée sont ajoutés au graphique.
 - Vous pouvez voir l'état de l'alarme dans la légende du graphique et sous forme d'icônes survolant le flux de données lorsque l'alarme change d'état.
 - Vous pouvez désactiver les icônes d'alarme dans les paramètres du graphique.

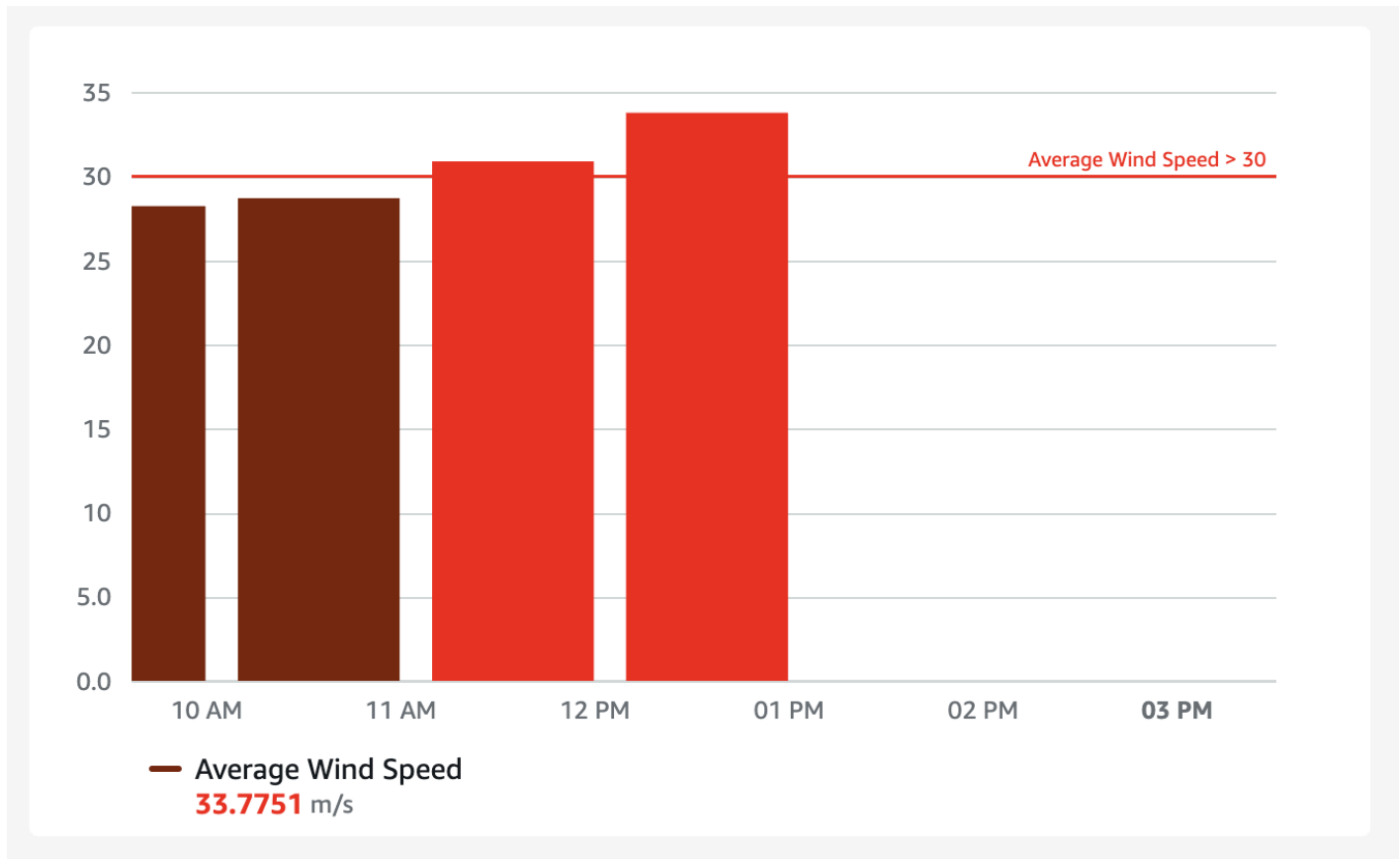


- KPI et jauge
 - L'alarme et son flux de données de propriété d'entrée sont ajoutés au widget choisi.
 - Le seuil d'alarme est ajouté au widget, qui change de couleur en fonction de sa configuration.
 - Vous pouvez sélectionner l'état de l'alarme sur le widget, voir les détails de l'alarme et cliquer sur Générer un résumé pour appeler le et AWS IoT SiteWise obtenir un résumé de l'alarme.



- Tableau

- L'alarme et sa propriété d'entrée sont ajoutées sous forme de ligne dans le tableau.
- Graphique à barres
 - L'alarme est ajoutée en tant que seuil au graphique, ce qui modifie la couleur de tout flux de données dépassant le seuil.
 - Vous pouvez ajouter les flux de données associés séparément.
 - Vous ne pouvez pas interagir avec l' AWS IoT SiteWise Assistant depuis le widget.



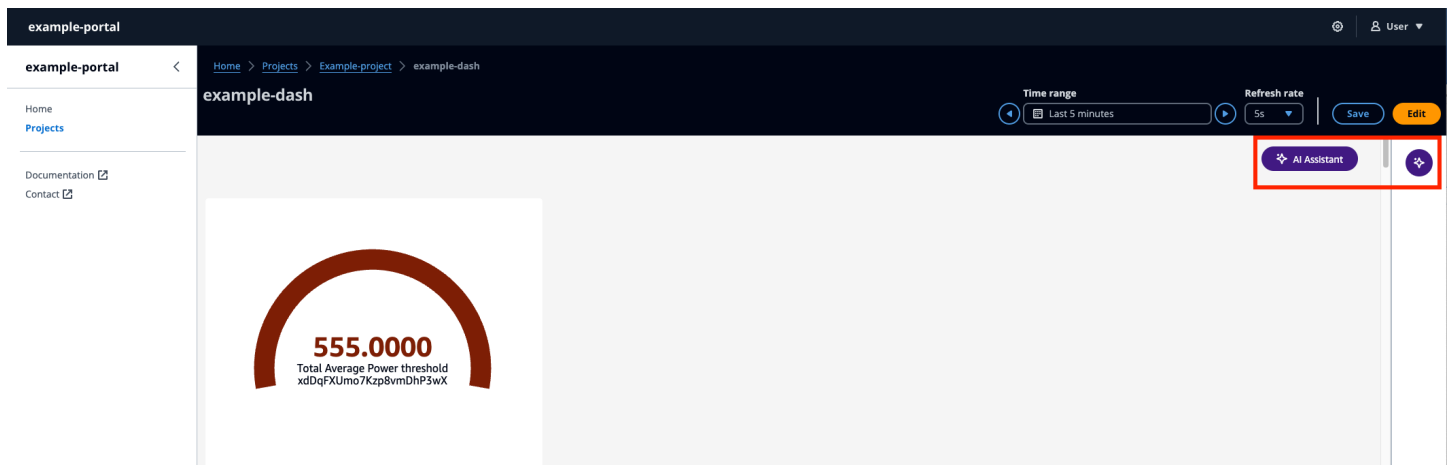
- Chronologie du statut
 - L'alarme est ajoutée en tant que seuil à la chronologie.
 - L'ajout de l'état de l'alarme et de ses données de propriété d'entrée à la chronologie est en cours.
 - Vous ne pouvez pas interagir avec l' AWS IoT SiteWise Assistant depuis le widget.

AWS IoT SiteWise Utilisation de l'assistant dans les widgets

L' AWS IoT SiteWise assistant est un assistant génératif alimenté par l'IA. Il permet aux utilisateurs tels que les directeurs d'usine, les ingénieurs qualité et les techniciens de maintenance d'obtenir des

informations, de résoudre des problèmes et de prendre des mesures directement à partir de leurs données opérationnelles et d'entreprise. L' AWS IoT SiteWise assistant consolide les informations issues AWS IoT des données, des modèles d'actifs, des manuels et de la documentation dans des résumés compréhensibles des événements critiques. Il permet également des sessions interactives de questions-réponses approfondies pour faciliter le diagnostic, l'exploration des causes profondes et les recommandations guidées.

Le bouton AWS IoT SiteWise Assistant se trouve dans le coin supérieur droit du tableau de bord. Cliquez dessus pour activer l'Assistant. Ne peut être utilisé qu'avec le mode Aperçu du tableau de bord.



Utilisez l' AWS IoT SiteWise Assistant dans les scénarios suivants :

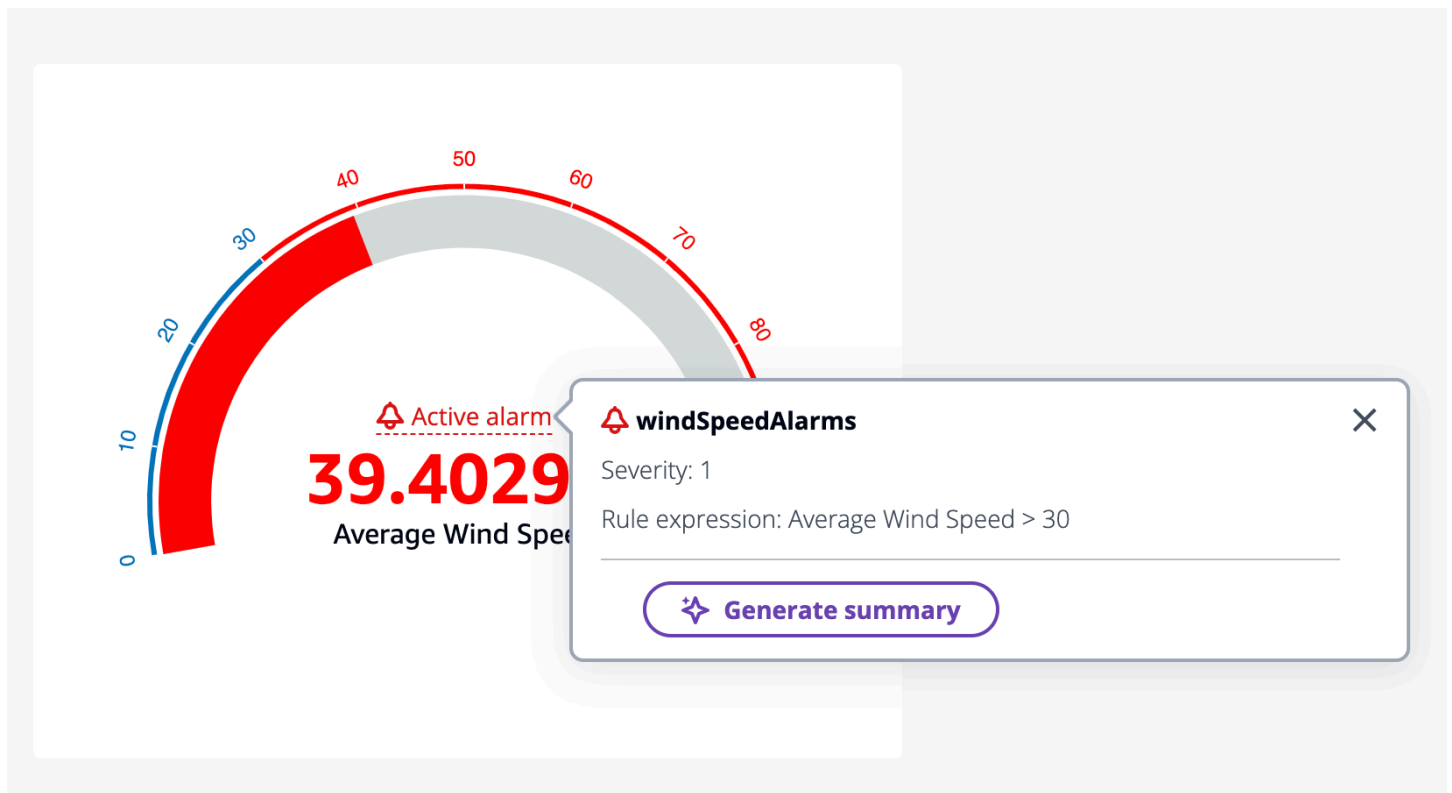
Rubriques

- [Cas d'utilisation - Résumés des alarmes](#)
- [Cas d'utilisation - Résumés situationnels](#)
- [Cas d'utilisation - Résumés détaillés](#)

Cas d'utilisation - Résumés des alarmes

Résumez l'alarme en cours pour un panneau sélectionné sur le tableau de bord. Les alarmes sont prises en charge par les widgets Line, KPI, Gauge et Table. Choisissez un widget avec une alarme et résumez-le.

- Sélectionnez Alarme active sur le widget.
- L'expression Severity and Rule s'affiche pour l'alarme.
- Choisissez Générer un résumé pour générer un résumé.



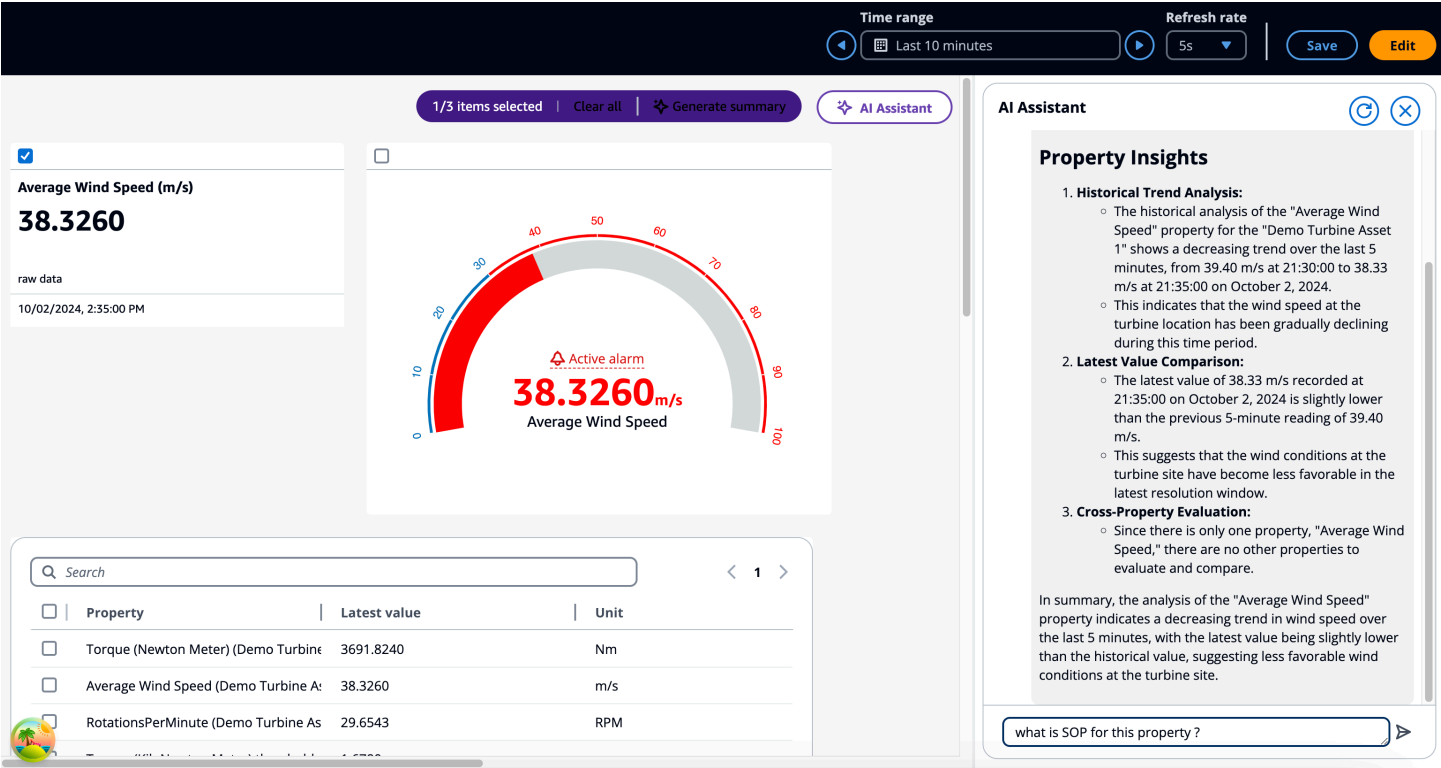
Cas d'utilisation - Résumés situationnels

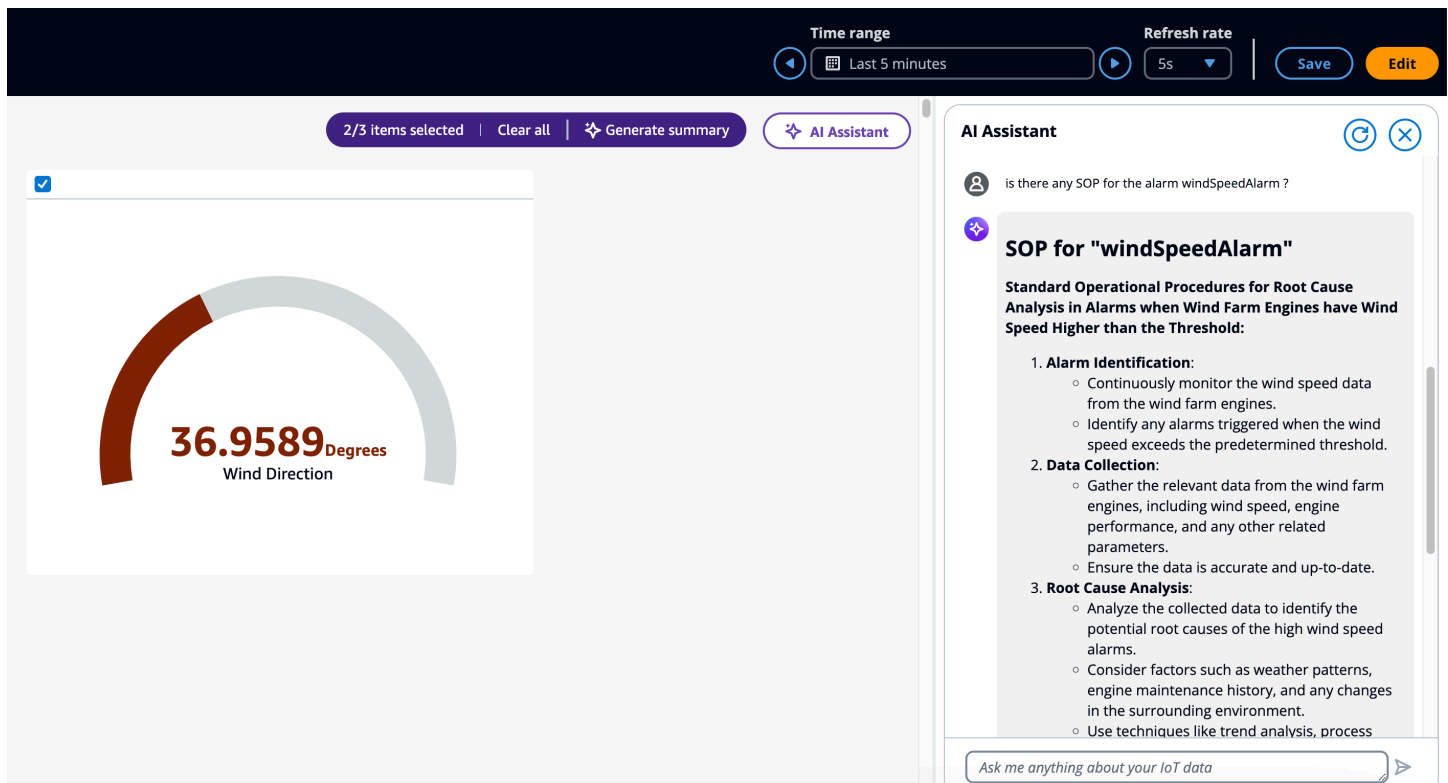
Sélectionnez jusqu'à trois widgets pour résumer. Il peut s'agir d'une combinaison de widgets et de propriétés. Si plus de trois sont sélectionnés, l'assistant renvoie un message d'erreur.

Générer un résumé de la situation avec AWS IoT SiteWise Assistant

1. Cliquez sur AI Assistant. Il affiche un menu avec trois options.
 - a. Éléments sélectionnés — N'en sélectionnez que trois. Vous ne pouvez pas en sélectionner plus de trois.
 - b. Tout effacer — Effacez votre sélection.
 - c. Générer un résumé : génère un résumé des éléments sélectionnés.
2. Choisissez Générer un résumé pour générer le résumé des éléments sélectionnés.

L'image ci-dessous contient un widget sélectionné et un résumé de l'AWS IoT SiteWise Assistant.





Exemples de questions à poser à l' AWS IoT SiteWise assistant

Note

- L' AWS IoT SiteWise assistant doit utiliser un ensemble de données avec un index [Amazon Kendra](#) pour obtenir des connaissances et des conseils au niveau de l'entreprise. Si vous ne possédez pas d'index Amazon Kendra, consultez [Création d'un index pour en créer un](#). L'ajout d'un [jeu de données](#) améliore la qualité de la réponse de l'assistant. Voir [Créer un jeu de données](#) pour en savoir plus.
- Certaines questions nécessitent une AWS IoT TwinMaker intégration. Voir [Intégrer AWS IoT TwinMaker et AWS IoT SiteWise](#) pour plus de détails.

Quelques questions de suivi à poser à l'assistant après avoir reçu un résumé des alarmes dans le tableau de bord, dans le cadre de la même conversation.

- Afficher les détails de l'actif dans le résumé ci-dessus ?
- Quel est le chemin hiérarchique entre la racine et l'actif mentionné ?
- Quels sont les actifs descendants dépendants de l'actif mentionné ?

- Quels sont les actifs dépendants de l'actif mentionné qui ont des alarmes actives ?
- Trouvez tous les actifs dotés d'alarmes actives.

Quelques questions de suivi à poser à l'assistant après avoir obtenu un résumé de la propriété dans le tableau de bord, dans le cadre de la même conversation.

- Effectuez la même analyse au cours des dernières 24 heures.
- Trouvez la documentation relative aux propriétés mentionnées ci-dessus.
- Fournissez les détails de l'identifiant de l'actif 1da67d28-14f8-4f71-a06a-386f0425a21d/du nom de l'actif Demo Turbine Asset 1.

Appelez l' AWS IoT SiteWise assistant depuis l'API.

- Générez un résumé de l'alarme pour le nom de l'alarme windSpeedAlarm dans l'identifiant de l'actif d591e153-e5cf-4206-96bb-ce3c119d9d2d.
- Générez un résumé des alarmes des 12 dernières heures/2 jours/1 semaine pour le nom windSpeedAlarm de l'alarme dans l'identifiant de l'actif. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Générer un résumé de propriété pour l'identifiant de propriété ab187fb7-d74b-44d9-bd9b-f2f19a9137cc dans l'identifiant de l'actif d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Générez un résumé de la propriété pour les 12 dernières heures/2 jours/1 semaine pour l'identifiant de la propriété dans l'identifiant ab187fb7-d74b-44d9-bd9b-f2f19a9137cc de l'actif. d591e153-e5cf-4206-96bb-ce3c119d9d2d
- Trouvez les actifs dont le nom est Turbine.
- Donnez-moi les valeurs de propriété actuelles de l'identifiant de propriété 5356168c-3390-456f-802c-9f6e047810d4 dans l'identifiant de l'actif d591e153-e5cf-4206-96bb-ce3c119d9d2d, 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Quelle est la relation entre l'identifiant de l'actif d591e153-e5cf-4206-96bb-ce3c119d9d2d et l'identifiant de l'actif 3cbb084e-1ded-4b08-9f21-1b47b2fb86fd.
- Trouvez de la documentation sur la façon de résoudre le problème de faible régime des éoliennes.
- Générez un résumé de propriété pour l'alias de propriété WindSpeed.
- Quels sont les contrôles préopérateurs selon ma base de connaissances ?

Interrogez les données de AWS IoT SiteWise

Vous pouvez utiliser les opérations de l' AWS IoT SiteWise API pour interroger les valeurs actuelles, les valeurs historiques et les agrégats des propriétés de vos actifs sur des intervalles de temps spécifiques.

Utilisez ces fonctionnalités pour mieux comprendre vos données. Par exemple, découvrez tous vos actifs avec une valeur de propriété donnée ou créez une représentation personnalisée de vos données. Vous pouvez également utiliser les opérations d'API pour développer des solutions logicielles qui s'intègrent aux données industrielles stockées dans vos AWS IoT SiteWise actifs. Vous pouvez également explorer les données de vos ressources en direct dans AWS IoT SiteWise Monitor. Pour savoir comment configurer le SiteWise moniteur, consultez [Surveillez les données avec AWS IoT SiteWise Monitor](#).

Les opérations décrites dans cette section renvoient des objets contenant des valeurs de propriété contenant des structures d'horodatage, de qualité et de valeur (TQV) :

- `timestamp` contient l'heure d'époque Unix actuelle en secondes avec un décalage en nanosecondes.
- La métrique `quality` contient l'une des chaînes suivantes, qui indiquent la qualité du point de données :
 - `GOOD`— Les données ne sont affectées par aucun problème.
 - `BAD`— Les données sont affectées par un problème tel qu'une défaillance du capteur.
 - `UNCERTAIN`— Les données sont affectées par un problème tel que l'imprécision du capteur.
- `value` contient l'un des champs suivants, selon le type de la propriété :
 - `booleanValue`
 - `doubleValue`
 - `integerValue`
 - `stringValue`
 - `nullValue`

Rubriques

- [Interrogez les valeurs actuelles des propriétés des actifs dans AWS IoT SiteWise](#)
- [Rechercher les valeurs historiques des propriétés des actifs dans AWS IoT SiteWise](#)

- [Interrogez les agrégats de propriétés des actifs dans AWS IoT SiteWise](#)
- [AWS IoT SiteWise langage de requête](#)

Interrogez les valeurs actuelles des propriétés des actifs dans AWS IoT SiteWise

Ce didacticiel montre deux méthodes pour obtenir la valeur actuelle d'une propriété d'actif. Vous pouvez utiliser la AWS IoT SiteWise console ou utiliser l'API dans le AWS Command Line Interface (AWS CLI).

Rubriques

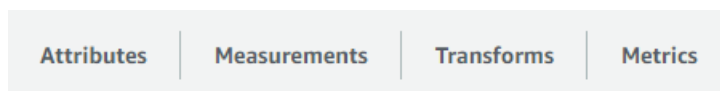
- [Rechercher la valeur actuelle d'une propriété d'actif \(console\)](#)
- [Rechercher la valeur actuelle d'une propriété d'actif \(AWS CLI\)](#)

Rechercher la valeur actuelle d'une propriété d'actif (console)

Vous pouvez utiliser la AWS IoT SiteWise console pour afficher la valeur actuelle d'une propriété d'actif.

Pour obtenir la valeur actuelle d'une propriété de ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource avec la propriété à interroger.
4. Cliquez sur l'icône en forme de flèche pour développer une hiérarchie d'actifs afin de trouver votre actif.
5. Choisissez l'onglet correspondant au type de propriété. Par exemple, choisissez Mesures pour afficher la valeur actuelle d'une propriété de mesure.



6. Trouvez la propriété à afficher. La valeur actuelle apparaît dans la colonne Dernière valeur.

Rechercher la valeur actuelle d'une propriété d'actif (AWS CLI)

Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour rechercher la valeur actuelle d'une propriété d'actif.

Utilisez l'[GetAssetPropertyValue](#) opération pour interroger la valeur actuelle d'une propriété d'actif.

Pour identifier une propriété d'actif, spécifiez l'une des options suivantes :

- La `assetId` fin `propertyId` de la propriété de l'actif à laquelle les données sont envoyées.
- Le `propertyAlias`, qui est un alias de flux de données (par exemple, `/company/windfarm/3/turbine/7/temperature`). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour définir des alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Pour obtenir la valeur actuelle d'un actif (AWS CLI)

- Exécutez la commande suivante pour obtenir la valeur actuelle de la propriété de ressource. Remplacez *asset-id* par l'ID de l'actif et *property-id* par l'ID de la propriété.

```
aws iotsitewise get-asset-property-value \  
  --asset-id asset-id \  
  --property-id property-id
```

L'opération renvoie une réponse qui contient les données TQV actuelles de la propriété au format suivant.

```
{  
  "propertyValue": {  
    "value": {  
      "booleanValue": Boolean,  
      "doubleValue": Number,  
      "integerValue": Number,  
      "stringValue": "String",  
      "nullValue": {  
        "valueType": "String"  
      }  
    },  
    "timestamp": {  
      "timeInSeconds": Number,
```

```
    "offsetInNanos": Number
  },
  "quality": "String"
}
```

Rechercher les valeurs historiques des propriétés des actifs dans AWS IoT SiteWise

Vous pouvez utiliser l'[GetAssetPropertyValueHistory](#) opération AWS IoT SiteWise API pour interroger les valeurs historiques d'une propriété d'actif.

Pour identifier une propriété d'actif, spécifiez l'une des options suivantes :

- La `assetId` fin `propertyId` de la propriété de l'actif à laquelle les données sont envoyées.
- Le `propertyAlias`, qui est un alias de flux de données (par exemple, `/company/windfarm/3/turbine/7/temperature`). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour définir des alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Passez les paramètres suivants pour affiner vos résultats :

- `startDate`— Le début exclusif de la plage à partir de laquelle interroger les données historiques, exprimé en secondes à l'époque Unix.
- `endDate`— La fin de la plage inclusive à partir de laquelle interroger les données historiques, exprimée en secondes à l'époque Unix.
- `maxResults`— Le nombre maximum de résultats à renvoyer par requête. Par défaut, ce sont les 20 résultats.
- `nextToken`— Un jeton de pagination renvoyé lors d'un précédent appel de cette opération.
- `timeOrdering`— La commande à appliquer aux valeurs renvoyées : `ASCENDING` ou `DESCENDING`.
- `qualities`— La qualité permettant de filtrer les résultats par : `GOODBAD`, ou `UNCERTAIN`.

Pour consulter l'historique des valeurs d'une propriété d'actif (AWS CLI)

1. Exécutez la commande suivante pour obtenir l'historique des valeurs de la propriété de ressource. Cette commande interroge l'historique de la propriété sur un intervalle spécifique de 10 minutes. Remplacez *asset-id* par l'ID de l'actif et *property-id* par l'ID de la propriété. Remplacez les paramètres de date par l'intervalle à interroger.

```
aws iotsitewise get-asset-property-value-history \  
  --asset-id asset-id \  
  --property-id property-id \  
  --start-date 1575216000 \  
  --end-date 1575216600
```

L'opération renvoie une réponse contenant l'historique TQVs de la propriété au format suivant :

```
{  
  "assetPropertyValueHistory": [  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    },  
    {  
      "value": {  
        "booleanValue": Boolean,  
        "doubleValue": Number,  
        "integerValue": Number,  
        "stringValue": "String",  
        "nullValue": {  
          "valueType": "String"  
        }  
      },  
      "timestamp": {  
        "timeInSeconds": Number,  
        "offsetInNanos": Number  
      },  
      "quality": "String"  
    }  
  ],  
  "nextToken": "String"  
}
```

2. S'il existe d'autres entrées de valeur, vous pouvez transmettre le jeton de pagination du `nextToken` champ à un appel ultérieur à l'[GetAssetPropertyValueHistory](#) opération.

Interrogez les agrégats de propriétés des actifs dans AWS IoT SiteWise

AWS IoT SiteWise calcule automatiquement les valeurs agrégées des propriétés des actifs, qui sont un ensemble de mesures de base calculées sur plusieurs intervalles de temps. AWS IoT SiteWise calcule les agrégats suivants chaque minute, heure et jour pour les propriétés de vos actifs :

- **average** — La moyenne (moyenne) des valeurs d'une propriété sur un intervalle de temps.
- **count** — Le nombre de points de données pour une propriété sur un intervalle de temps.
- **maximum** — Le maximum des valeurs d'une propriété sur un intervalle de temps.
- **minimum** — Le minimum des valeurs d'une propriété sur un intervalle de temps.
- **écart type** : écart type des valeurs d'une propriété sur un intervalle de temps.
- **sum** — Somme des valeurs d'une propriété sur un intervalle de temps.

Pour les propriétés non numériques, telles que les chaînes et les booléens, AWS IoT SiteWise calcule uniquement le nombre agrégé.

Vous pouvez également calculer des métriques personnalisées pour vos données d'actif. Les propriétés des métriques vous permettent de définir des agrégations spécifiques à votre opération. Les propriétés métriques offrent des fonctions d'agrégation et des intervalles de temps supplémentaires qui ne sont pas précalculés pour l' AWS IoT SiteWise API. Pour de plus amples informations, veuillez consulter [Données agrégées provenant de propriétés et d'autres actifs \(métriques\)](#).

Rubriques

- [Agrégats pour une propriété d'actif \(API\)](#)
- [Agrégats pour une propriété d'actif \(AWS CLI\)](#)

Agrégats pour une propriété d'actif (API)

Utilisez l' AWS IoT SiteWise API pour obtenir des agrégats pour une propriété d'actif.

Utilisez l'[GetAssetPropertyAggregates](#) opération pour interroger les agrégats d'une propriété d'actif.

Pour identifier une propriété d'actif, spécifiez l'une des options suivantes :

- La `assetId` fin `propertyId` de la propriété de l'actif à laquelle les données sont envoyées.
- Le `propertyAlias`, qui est un alias de flux de données (par exemple, `/company/windfarm/3/turbine/7/temperature`). Pour utiliser cette option, vous devez d'abord définir l'alias de votre propriété de ressource. Pour définir des alias de propriété, consultez [Gérez les flux de données pour AWS IoT SiteWise](#).

Vous devez transmettre les paramètres obligatoires suivants :

- `aggregateTypes`— La liste des agrégats à récupérer. Vous pouvez spécifier n'importe quel élément : AVERAGE, COUNT, MAXIMUM, MINIMUM, STANDARD_DEVIATION et SUM.
- `resolution`— Intervalle de temps pendant lequel la métrique doit être 1m récupérée : (1 minute), 15m (15 minutes), 1h (1 heure) ou 1d (1 jour).
- `startDate`— Le début exclusif de la plage à partir de laquelle interroger les données historiques, exprimé en secondes à l'époque Unix.
- `endDate`— La fin de la plage inclusive à partir de laquelle interroger les données historiques, exprimée en secondes à l'époque Unix.

Vous pouvez également passer l'un des paramètres suivants pour affiner vos résultats :

- `maxResults`— Le nombre maximum de résultats à renvoyer par requête. Par défaut, ce sont les 20 résultats.
- `nextToken`— Un jeton de pagination renvoyé lors d'un précédent appel de cette opération.
- `timeOrdering`— La commande à appliquer aux valeurs renvoyées : ASCENDING ou DESCENDING.
- `qualities`— La qualité permettant de filtrer les résultats par : GOODBAD, ou UNCERTAIN.

Note

L'[GetAssetPropertyAggregates](#) opération renvoie un TQV avec un format différent de celui des autres opérations décrites dans cette section. La structure `value` contient un champ pour chacun des éléments `aggregateTypes` de la demande. Le `timestamp` contient l'heure à laquelle l'agrégation s'est produite, en secondes (heure UNIX Epoch).

Agrégats pour une propriété d'actif (AWS CLI)

Pour interroger des agrégats pour une propriété d'actif (AWS CLI)

1. Exécutez la commande suivante pour obtenir des agrégats pour la propriété de ressource. Cette commande interroge la moyenne et la somme avec une résolution d'1 heure pour un intervalle spécifique d'1 heure. Remplacez *asset-id* par l'ID de l'actif et *property-id* par l'ID de la propriété. Remplacez les paramètres par les agrégats et l'intervalle à interroger.

```
aws iotsitewise get-asset-property-aggregates \
  --asset-id asset-id \
  --property-id property-id \
  --start-date 1575216000 \
  --end-date 1575219600 \
  --aggregate-types AVERAGE SUM \
  --resolution 1h
```

L'opération renvoie une réponse contenant l'historique TQVs de la propriété au format suivant. La réponse inclut uniquement les agrégats demandés.

```
{
  "aggregatedValues": [
    {
      "timestamp": Number,
      "quality": "String",
      "value": {
        "average": Number,
        "count": Number,
        "maximum": Number,
        "minimum": Number,
        "standardDeviation": Number,
        "sum": Number
      }
    }
  ],
  "nextToken": "String"
}
```

2. S'il existe d'autres entrées de valeur, vous pouvez transmettre le jeton de pagination du `nextToken` champ à un appel ultérieur à l'[GetAssetPropertyAggregates](#) opération.

Note

Si votre plage de requêtes contient une `null` valeur TQVs, consultez [AssetPropertyValueAPI](#). Toutes les statistiques, à l'exception du nombre, donnent lieu à une `null` réponse, similaire aux statistiques de String TQVs. Si votre plage de requêtes contient `Double.NaN` un type double TQVs, tous les calculs, à l'exception du nombre, aboutiront à `unDouble.NaN`.

AWS IoT SiteWise langage de requête

Grâce à l'opération d'[ExecuteQuery](#) API de récupération de AWS IoT SiteWise données, vous pouvez récupérer des informations sur les définitions structurelles déclaratives et les données de séries chronologiques qui leur sont associées, à partir des informations suivantes :

- des modèles
- actifs
- mesures
- métriques
- transforme
- agrégats

Cela peut être fait avec des instructions de requête de type SQL, dans une seule requête d'API.

Note

Cette fonctionnalité est disponible dans toutes les régions où AWS IoT SiteWise les deux AWS IoT TwinMaker sont disponibles, sauf AWS GovCloud dans l'ouest des États-Unis.

Rubriques

- [Prérequis](#)
- [Référence du langage de requête pour AWS IoT SiteWise](#)

Prérequis

AWS IoT SiteWise nécessite des autorisations d'intégration AWS IoT TwinMaker afin de pouvoir organiser et modéliser les données industrielles.

Avant de pouvoir récupérer des informations sur les modèles, les actifs, les mesures, les métriques, les transformations et les agrégats, assurez-vous que les conditions préalables suivantes sont remplies :

- Rôles liés au service pour les deux AWS IoT SiteWise et AWS IoT TwinMaker configurez-les dans votre AWS compte. Pour plus d'informations sur les rôles liés à un service, voir [Créer un rôle lié à un service dans le Guide de l'utilisateur](#) d'IAM.
- Une AWS IoT SiteWise intégration activée pour votre rôle IAM. Pour de plus amples informations, veuillez consulter [Intégrer AWS IoT SiteWise et AWS IoT TwinMaker](#).
- Un AWS IoT TwinMaker espace de travail avec un identifiant `IoTSiteWiseDefaultWorkspace` dans votre compte dans la région. Pour plus d'informations, consultez [Utilisation d'IoTSiteWiseDefaultWorkspace](#) dans le Guide de l'utilisateur AWS IoT TwinMaker .
- Les modes de tarification groupée standard ou échelonnés sont AWS IoT TwinMaker activés. Pour plus d'informations, consultez la section [Changer de mode de AWS IoT TwinMaker tarification](#) dans le guide de AWS IoT TwinMaker l'utilisateur.

Référence du langage de requête pour AWS IoT SiteWise

AWS IoT SiteWise prend en charge un langage de requête riche pour travailler avec vos données. Les types de données, les opérateurs, les fonctions et les constructions disponibles sont décrits dans les rubriques suivantes.

Voir [Exemples de requêtes](#) pour écrire des requêtes avec le langage de AWS IoT SiteWise requête.

Rubriques

- [Comprendre les vues de référence des requêtes](#)
- [Types de données pris en charge](#)
- [Récupérez des données avec une instruction SELECT](#)
- [Opérateurs logiques](#)
- [Opérateurs de comparaison](#)
- [Exemples de requêtes](#)

Comprendre les vues de référence des requêtes

Cette section fournit des informations pour vous aider à comprendre les vues AWS IoT SiteWise, telles que les métadonnées de processus et les données de télémétrie.

Les tableaux suivants fournissent les noms et les descriptions des vues.

Modèle de données

Nom de la vue	Description de la vue
asset	Contient des informations sur l'actif et la dérivation du modèle.
propriété_actif	Contient des informations sur la structure de la propriété de l'actif.
Série Raw_Time	Contient les données historiques de la série chronologique.
dernière_value_time_series	Contient la dernière valeur de la série chronologique.
agrégats_précalculés	Contient les valeurs agrégées des propriétés des actifs calculées automatiquement. Il s'agit d'un ensemble de mesures de base calculées sur plusieurs intervalles de temps.

Les vues suivantes répertorient les noms de colonnes pour les requêtes ainsi que des exemples de données.

Voir : actif

identifiant_actif	nom_actif	description_de l'actif	identifiant_modèle d'actif
88898498-0b8b-42b5-bf57-16180bc3d3a0	WindTurbine A	WindTurbine Actif A	17847250-5bf0-4f74-b775-cc03f05e7cb8

identifiant_actif	nom_actif	description_de l'actif	identifiant_modèle d'actif
17847250-5bf0-4f74-b775-cc03f05e7cb8	Modèle d'actifs d'éoliennes	Représente une turbine dans un parc éolien.	

Voir : asset_property

identifiant_propriété	identifiant_actif	nom_propriété	alias de propriété	asset_composite_model_id	
b29be434-b000-4d74-b809-75287d83bcd6	88898498-0b8b-42b5-bf57-16180bc3d3a0	température du moteur	Rochester2/44///Line-5/Bus-2/Machine-5/Temperature		
3b458f00-24e7-458a-b4e8-c6026eff654a	88898498-0b8b-42b5-bf57-16180bc3d3a0	direction du vent	/company/windfarm/3/turbine/7/winddirection	2f458n00-56e7-458h-b4e8-c6026eff985 g	

Voir : RAW_TIME_SERIES

identifiant_actif	identifiant_propriété	alias de propriété	horodatage de l'événement	qualité	valeur_boléenne	int_value	valeur_double	valeur_chaine
88898498-0b8b-42b5-bf57-161	b29be434-b000-4d74-b809-752	Rochester2/44///Line-5/	157521960	BON			115,0	

identifiant_actif	identifiant_propriété	alias de propriété	horodatage de l'événement	qualité	valeur_boléenne	int_value	valeur_double	valeur_chaîne
80bc3d3a	87d83bcd	Bus-2/ Machine-5/ Temperature						
888984980b8b-42b- bf57-16180bc3d3a	3b458f00-24e7-458a-b4e8-c6026eff654a	/ company, windfarm 3/ turbine /7/ winddirection	157521937	BON			348,75	

Note

Vous devez inclure une clause de filtre dans la `event_timestamp` colonne pour interroger la `raw_time_series` vue. Il s'agit d'un filtre obligatoire, et la requête échouera sans ce filtre.

Example query

```
SELECT event_timestamp, double_value FROM raw_time_series WHERE event_timestamp > 1234567890
```


Voir : LATEST_VALUE_TIME_SERIES

identifiant_actif	identifiant_propriété	alias de propriété	horodatage de l'événement	qualité	valeur_boléenne	int_value	valeur_double	valeur_chaine
888984980b8b-42b1-bf57-16180bc3d3a	3b458f00-24e7-4581-b4e8-c6026eff654a	/ company, windfarm 3/ turbine /7/ winddirection	15752196	BON			355,39	

Voir : precomputed_aggregates

identifiant_actif	identifiant_propriété	alias de propriété	horodatage de l'événement	qualité	résolution	valeur_mme	valeur_mptage	valeur_yenne	valeur_ximale	valeur_nimale	stdev_value
888984980b8b-42b1-bf57-16180bc3d3a	b29be4b000-4b809-787d83b	Roche 2/44/, Li ne-5/ Bus-2/ Machi -5/ Tempe ature	157521	BON	15 min	1105,4	15	73,4	80,6	68	3,64

Types de données pris en charge

AWS IoT SiteWise le langage de requête prend en charge les types de données suivants.

Valeur scalaire

Type de données	Description
STRING	Chaîne d'une longueur maximale de 1024 octets.
INTEGER	Un entier signé de 32 bits dont la plage est comprise -2,147,483,648 to 2,147,483,647 entre.
DOUBLE	Nombre à virgule flottante dont la plage - 10^{100} to 10^{100} est comprise entre ou Nan avec une IEEE 754 double précision.
BOOLEAN	true ou false.

Nullvaleur : un booléen true indiquant l'absence de données définies.

Note

Les données de double précision ne sont pas exactes. Certaines valeurs ne sont pas converties exactement et ne représenteront pas tous les nombres réels en raison d'une précision limitée. Les données à virgule flottante de la requête peuvent ne pas avoir la même valeur que celle représentée en interne. La valeur est arrondie si la précision d'un nombre saisi est trop élevée.

Récupérez des données avec une instruction SELECT

L'SELECT instruction est utilisée pour récupérer des données à partir d'une ou de plusieurs vues. AWS IoT SiteWise soutient un point JOIN de vue implicite. Vous pouvez répertorier les vues à joindre (dans la FROM clause de l'SELECT instruction) en les séparant par des virgules.

Exemple

Utilisez l'SELECT énoncé suivant :

```
SELECT select_expr [, ...]  
[ FROM from_item [, ...] ]  
[ WHERE [LIKE condition ESCAPE condition] ]
```

Dans l'exemple précédent, la LIKE clause spécifie les conditions de recherche et de filtrage à l'aide de caractères génériques. AWS IoT SiteWise supporte en pourcentage (%) tant que caractère joker.

Exemple à utiliser % dans les conditions suivantes :

```
Prefix search: String%  
Infix search: %String%  
Suffix search: %String
```

Exemple pour rechercher un actif :

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'Wind%'
```

Exemple pour rechercher un actif à l'aide d'une condition ESCAPE :

```
SELECT asset_name, asset_description FROM asset WHERE asset_name LIKE 'room\%' ESCAPE  
'\'
```

Opérateurs logiques

AWS IoT SiteWise prend en charge les opérateurs logiques suivants.

Opérateurs logiques

Opérateur	Description	Exemple
AND	TRUE si les deux valeurs sont vraies	un AND b

Si a ou b l'est FALSE, l'expression précédente est considérée comme fausse. Pour qu'un AND opérateur soit évalué comme vrai, a et b doivent tous deux être vrais.

Exemple

```
SELECT a.asset_name
FROM asset as a, latest_value_time_series as t
WHERE t.int_value > 30 AND t.event_timestamp > 1234567890
```

Opérateurs de comparaison

AWS IoT SiteWise prend en charge les opérateurs de comparaison suivants. Toutes les opérations de comparaison sont disponibles pour les types de données intégrés et sont évaluées selon un booléen.

Opérateurs logiques

Opérateur	Description
<	Inférieur à
>	Supérieure à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
=	Égal à
!=	Non égal à

Table de vérité des opérations de comparaison pour les valeurs non numériques

Type	Tapez >= x	Type <= x	Tapez > x	Type < x	Type = x	Tapez != x
NaN	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
NULL	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE

Certains prédicats se comportent comme des opérateurs mais ont une syntaxe spéciale. Voir les exemples ci-dessous.

Prédicats de comparaison

Opérateur	Description
IS NULL	Teste si une valeur estNULL.
IS NOT NULL	Teste si une valeur ne l'est pasNULL.
IS NaN	Teste si une valeur estNaN.
IS NOT NaN	Teste si une valeur ne l'est pasNaN.

Exemples de requêtes

Filtrage des métadonnées

L'exemple suivant concerne le filtrage des métadonnées à l'aide d'une SELECT instruction contenant le langage de AWS IoT SiteWise requête :

```
SELECT a.asset_name, p.property_name
FROM asset a, asset_property p
WHERE a.asset_id = p.asset_id AND a.asset_name LIKE '%windmill%'
```

Filtrage des valeurs

Voici un exemple de filtrage de valeurs à l'aide d'une SELECT instruction utilisant le langage de AWS IoT SiteWise requête :

```
SELECT a.asset_name FROM asset a, raw_time_series r
WHERE a.asset_id = r.asset_id AND r.int_value > 30 AND r.event_timestamp > 1234567890
AND r.event_timestamp < 1234567891
```

Interagissez avec d'autres AWS services

AWS IoT SiteWise peut publier des données d'actifs sur le courtier de messages de publication/abonnement AWS IoT MQTT, afin que vous puissiez interagir avec les données de vos actifs provenant d'autres services. AWS IoT SiteWise attribue à chaque propriété d'actif un sujet MQTT unique que vous pouvez utiliser pour acheminer les données de vos actifs vers d'autres AWS services à l'aide des règles de AWS IoT base. Par exemple, vous pouvez configurer les règles de AWS IoT base pour effectuer les tâches suivantes :

- Identifier les défaillances de l'équipement et aviser le personnel approprié en envoyant des données à [AWS IoT Events](#).
- Historisez certaines données d'actifs à utiliser dans des solutions logicielles externes en envoyant des données à [Amazon DynamoDB](#).
- Générez des rapports hebdomadaires en déclenchant une fonction [AWS Lambda](#).

Vous pouvez suivre un didacticiel qui décrit les étapes nécessaires pour configurer une règle qui stocke les valeurs des propriétés dans DynamoDB. Pour de plus amples informations, veuillez consulter [Publier les mises à jour de la valeur des propriétés sur Amazon DynamoDB](#).

Pour plus d'informations sur la configuration d'une règle, consultez la section [Règles](#) du guide du AWS IoT développeur.

Vous pouvez également réutiliser les données d'autres AWS services dans AWS IoT SiteWise. Pour ingérer des données par le biais de l'action de la AWS IoT SiteWise règle, consultez [Ingérer des données pour AWS IoT SiteWise utiliser des règles AWS IoT Core](#).

Rubriques

- [Comprendre les propriétés des actifs dans les rubriques MQTT](#)
- [Activez les notifications relatives aux propriétés des actifs dans AWS IoT SiteWise](#)
- [Interrogez les notifications relatives aux propriétés des actifs dans AWS IoT SiteWise](#)
- [Exportez des données vers Amazon S3 avec des notifications relatives aux propriétés des actifs](#)
- [Intégrez AWS IoT SiteWise Grafana](#)
- [Intégrer AWS IoT SiteWise et AWS IoT TwinMaker](#)
- [Déectez les anomalies avec Lookout for Equipment](#)

Comprendre les propriétés des actifs dans les rubriques MQTT

Chaque propriété de ressource possède un chemin d'accès de rubrique MQTT unique au format suivant.

```
$aws/sitewise/asset-models/assetModelId/assets/assetId/properties/propertyId
```

Note

AWS IoT SiteWise ne prend pas en charge le caractère générique du filtre de rubrique # (à plusieurs niveaux) dans le moteur de règles de AWS IoT base. Vous pouvez utiliser le caractère générique + (niveau unique). Par exemple, vous pouvez utiliser le filtre de rubrique suivant pour faire correspondre toutes les mises à jour d'un modèle de ressource particulier.

```
$aws/sitewise/asset-models/assetModelId/assets/+ /properties/+
```

Pour en savoir plus sur les caractères génériques utilisés pour filtrer les [rubriques](#), consultez la section Rubriques du Guide du développeur AWS IoT principal.

Activez les notifications relatives aux propriétés des actifs dans AWS IoT SiteWise

Vous pouvez activer les notifications de propriété pour publier les mises à jour des données des actifs AWS IoT Core, puis exécuter des requêtes sur ces données. Avec les notifications relatives aux propriétés des actifs, AWS IoT SiteWise fournit un AWS CloudFormation modèle que vous pouvez utiliser pour exporter AWS IoT SiteWise des données vers Amazon S3.

Note

Les données relatives aux actifs sont envoyées à AWS IoT Core chaque fois qu'elles sont reçues AWS IoT SiteWise, que leur valeur ait changé ou non.

Rubriques

- [Activer les notifications relatives aux propriétés des actifs \(console\)](#)
- [Activer les notifications relatives aux propriétés des actifs \(AWS CLI\)](#)

Activer les notifications relatives aux propriétés des actifs (console)

Par défaut, AWS IoT SiteWise ne publie pas les mises à jour de la valeur des propriétés. Vous pouvez utiliser la AWS IoT SiteWise console pour activer les notifications relatives à une propriété d'actif.

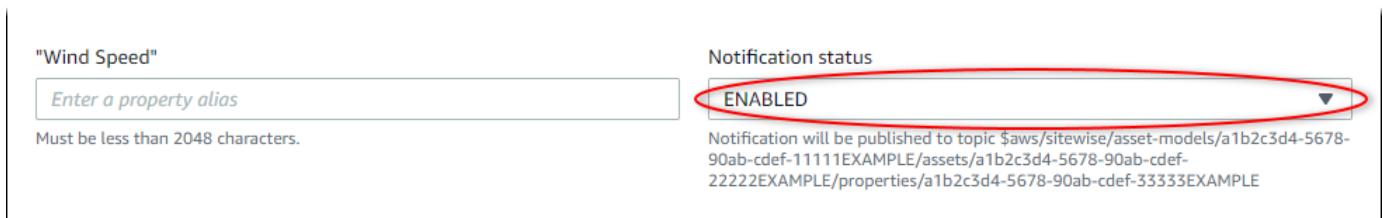
Pour activer ou désactiver les notifications pour une propriété de ressource (console)

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Ressources.
3. Choisissez la ressource pour activer les notifications d'une propriété.

Tip

Vous pouvez cliquer sur l'icône en forme de flèche pour développer une hiérarchie de ressources afin de trouver votre ressource.

4. Choisissez Modifier.
5. Pour le statut de notification de la propriété de ressource, choisissez ACTIVÉ.



The screenshot shows the AWS IoT SiteWise console interface. On the left, there is a text input field labeled '"Wind Speed"' with a placeholder text 'Enter a property alias' and a note 'Must be less than 2048 characters.' To the right of this field is a dropdown menu titled 'Notification status'. The dropdown is currently set to 'ENABLED', which is highlighted with a red oval. Below the dropdown, there is a detailed notification topic string: 'Notification will be published to topic \$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'.

Vous pouvez également choisir DÉSACTIVÉ pour désactiver les notifications pour la propriété de ressource.

6. Choisissez Save (Enregistrer).

Activer les notifications relatives aux propriétés des actifs (AWS CLI)

Par défaut, AWS IoT SiteWise ne publie pas les mises à jour de la valeur des propriétés. Vous pouvez utiliser le AWS Command Line Interface (AWS CLI) pour activer ou désactiver les notifications relatives à une propriété d'actif.

Vous devez connaître l'assetId de votre ressource et le propertyId de la propriété pour effectuer cette procédure. Vous pouvez également utiliser l'identifiant externe. Si vous avez créé un actif et

que vous ne le connaissez pas `assetId`, utilisez l'[ListAssets](#) API pour répertorier tous les actifs d'un modèle spécifique. Utilisez cette [DescribeAsset](#) opération pour afficher les propriétés de votre actif, y compris les propriétés IDs.

Utilisez cette [UpdateAssetProperty](#) opération pour activer ou désactiver les notifications pour une propriété d'actif. Spécifiez les paramètres suivants :

- `assetId`— L'identifiant de l'actif.
- `propertyId`— L'ID de la propriété de l'actif.
- `propertyNotificationState`— État de notification de la valeur de la propriété : `ENABLED` ou `DISABLED`.
- `propertyAlias`— L'alias de la propriété. Spécifiez l'alias existant de la propriété lorsque vous mettez à jour l'état de notification. Si vous omettez ce paramètre, l'alias existant de la propriété est supprimé.

Pour activer ou désactiver les notifications pour une propriété de ressource (interface de ligne de commande)

1. Exécutez la commande suivante pour récupérer l'alias de la propriété de ressource. Remplacez *asset-id* par l'ID de l'actif et *property-id* par l'ID de la propriété.

```
aws iotsitewise describe-asset-property \  
  --asset-id asset-id \  
  --property-id property-id
```

L'opération renvoie une réponse qui contient les informations de propriété de ressource au format suivant. L'alias de propriété se trouve dans `assetProperty.alias` dans l'objet JSON.

```
{  
  "assetId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",  
  "assetName": "Wind Turbine 7",  
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-1111EXAMPLE",  
  "assetProperty": {  
    "id": "a1b2c3d4-5678-90ab-cdef-3333EXAMPLE",  
    "name": "Wind Speed",  
    "alias": "/company/windfarm/3/turbine/7/windspeed",  
    "notification": {
```

```

    "topic": "$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/
assets/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE/properties/a1b2c3d4-5678-90ab-
cdef-33333EXAMPLE",
    "state": "DISABLED"
  },
  "dataType": "DOUBLE",
  "unit": "m/s",
  "type": {
    "measurement": {}
  }
}
}

```

2. Exécutez la commande suivante pour activer les notifications pour la propriété de ressource. Remplacez-le *property-alias* par l'alias de propriété indiqué dans la réponse de la commande précédente, ou omettez `--property-alias` de mettre à jour la propriété sans alias.

```

aws iotsitewise update-asset-property \
  --asset-id asset-id \
  --property-id property-id \
  --property-notification-state ENABLED \
  --property-alias property-alias

```

Vous pouvez également passer `--property-notification-state DISABLED` pour désactiver les notifications pour la propriété de ressource.

Interrogez les notifications relatives aux propriétés des actifs dans AWS IoT SiteWise

Pour interroger les notifications relatives aux propriétés des actifs, créez des AWS IoT Core règles composées d'instructions SQL.

AWS IoT SiteWise publie les mises à jour des données relatives aux propriétés des actifs dans AWS IoT Core au format suivant.

```

{
  "type": "PropertyValueUpdate",
  "payload": {
    "assetId": "String",

```

```
"propertyId": "String",
"values": [
  {
    "timestamp": {
      "timeInSeconds": Number,
      "offsetInNanos": Number
    },
    "quality": "String",
    "value": {
      "booleanValue": Boolean,
      "doubleValue": Number,
      "integerValue": Number,
      "stringValue": "String",
      "nullValue": {
        "valueType": "String"
      }
    }
  }
]
```

Chaque structure de la values liste est une structure timestamp-quality-value (TQV).

- timestamp contient l'heure d'époque Unix actuelle en secondes avec un décalage en nanosecondes.
- La métrique quality contient l'une des chaînes suivantes, qui indiquent la qualité du point de données :
 - GOOD— Les données ne sont affectées par aucun problème.
 - BAD— Les données sont affectées par un problème tel qu'une défaillance du capteur.
 - UNCERTAIN— Les données sont affectées par un problème tel que l'imprécision du capteur.
- value contient l'un des champs suivants, selon le type de la propriété :
 - booleanValue
 - doubleValue
 - integerValue
 - stringValue
 - nullValue

`nullValue`— Une structure dont le champ suivant indique le type de la valeur de propriété avec la valeur Null et la qualité de BAD ou UNCERTAIN.

- `valueType`— Enum de {"B », « D », « S », « I"}

Pour analyser les valeurs sorties du tableau `values`, vous devez utiliser des requêtes d'objets imbriqués complexes dans les instructions SQL de vos règles. Pour plus d'informations, consultez la section [Requêtes d'objets imbriqués](#) dans le guide du AWS IoT développeur ou consultez le [Publier les mises à jour de la valeur des propriétés sur Amazon DynamoDB](#) didacticiel pour un exemple spécifique d'analyse des messages de notification relatifs aux propriétés des actifs.

Exemple Exemple d'interrogation pour extraire le tableau de valeurs

L'instruction suivante montre comment interroger le tableau de valeurs de propriété mises à jour pour une propriété de type double spécifique sur toutes les ressources avec cette propriété.

```
SELECT
  (SELECT VALUE (value.doubleValue) FROM payload.values) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

L'instruction d'interrogation de la règle précédente génère les données dans le format suivant.

```
{
  "windspeed": [
    26.32020195042838,
    26.282584572975477,
    26.352566977372508,
    26.283084346171442,
    26.571883739599322,
    26.60684140743005,
    26.628738636715045,
    26.273486932802125,
    26.436379105473964,
    26.600590095377303
  ]
}
```

Exemple Exemple d'interrogation pour extraire une valeur unique

L'instruction suivante montre comment interroger la première valeur du tableau de valeurs de propriété pour une propriété de type double spécifique sur tous les actifs avec cette propriété.

```
SELECT
  get((SELECT VALUE (value.doubleValue) FROM payload.values), 0) AS windspeed
FROM
  '$aws/sitewise/asset-models/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE/assets/+/
properties/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE'
WHERE
  type = 'PropertyValueUpdate'
```

L'instruction d'interrogation de la règle précédente génère les données dans le format suivant.

```
{
  "windspeed": 26.32020195042838
}
```

Important

Cette instruction de requête de règle ignore les mises à jour de valeurs autres que la première dans chaque lot. Chaque lot peut contenir jusqu'à 10 valeurs. Si vous devez inclure les valeurs restantes, vous devez configurer une solution plus complexe pour générer des valeurs de propriété de ressources vers d'autres services. Par exemple, vous pouvez définir une règle avec une AWS Lambda action permettant de republier chaque valeur du tableau dans un autre sujet, et définir une autre règle pour interroger ce sujet et publier chaque valeur selon l'action de règle souhaitée.

Exportez des données vers Amazon S3 avec des notifications relatives aux propriétés des actifs

Vous pouvez exporter les données entrantes depuis AWS IoT SiteWise un compartiment Amazon S3 de votre compte. Vous pouvez sauvegarder vos données dans un format que vous pouvez utiliser pour créer des rapports historiques ou pour analyser vos données à l'aide de méthodes complexes.

Pour exporter des données de séries chronologiques AWS IoT SiteWise, activez la fonctionnalité Cold Tier afin que les données soient stockées dans un compartiment Amazon S3. Voir [Gérer le stockage des données AWS IoT SiteWise](#) pour plus de détails.

Pour exporter le modèle d'actif et les métadonnées des actifs à partir de AWS IoT SiteWise, utilisez la fonctionnalité d'opérations groupées pour exporter les métadonnées vers un compartiment Amazon S3. Voir [Opérations groupées avec des actifs et des modèles](#) pour plus de détails.

Intégrez AWS IoT SiteWise Grafana

Grafana est une plateforme de visualisation de données utilisée pour visualiser et surveiller les données dans des tableaux de bord. Dans les versions 10.4.0 et ultérieures de Grafana, utilisez le AWS IoT SiteWise plugin pour visualiser les données de vos AWS IoT SiteWise actifs dans les tableaux de bord Grafana. Les utilisateurs peuvent visualiser les données provenant de plusieurs AWS sources (telles qu' AWS IoT SiteWise Amazon Timestream et CloudWatch Amazon) et d'autres sources de données à l'aide d'un seul tableau de bord Grafana.

Deux options s'offrent à vous pour utiliser le AWS IoT SiteWise plugin :

- Serveurs Grafana locaux

Vous pouvez configurer le AWS IoT SiteWise plugin sur un serveur Grafana que vous gérez. Pour plus d'informations sur l'ajout et l'utilisation du plugin, consultez le fichier [AWS IoT SiteWise Datasource README](#) sur le GitHub site Web.

- AWS Service géré pour Grafana

Vous pouvez utiliser le AWS IoT SiteWise plugin dans le AWS Managed Service for Grafana (AMG). AMG gère les serveurs Grafana pour vous afin que vous puissiez visualiser vos données sans avoir à créer, emballer ou déployer du matériel ou toute autre infrastructure Grafana. Pour plus d'informations, consultez les rubriques suivantes dans le guide de l'utilisateur de AWS Managed Service for Grafana :

- [Qu'est-ce qu'Amazon Managed Service for Grafana \(AMG\) ?](#)
- [Utilisation de la source AWS IoT SiteWise de données](#)

Exemple Exemple de tableau de bord Grafana

Le tableau de bord Grafana suivant permet de visualiser le parc éolien de [démonstration](#). Vous pouvez accéder à ce tableau de bord de démonstration sur le site Web de [Grafana Play](#).



Intégrer AWS IoT SiteWise et AWS IoT TwinMaker

L'intégration AWS IoT TwinMaker permet d'accéder à des fonctionnalités robustes AWS IoT SiteWise, telles que l'ExecuteQueryAPI de récupération de AWS IoT SiteWise données et la recherche avancée d'actifs dans la AWS IoT SiteWise console. Pour intégrer les services et utiliser ces fonctionnalités, vous devez d'abord activer l'intégration.

Rubriques

- [Activation de l'intégration](#)
- [Intégration AWS IoT SiteWise et AWS IoT TwinMaker](#)

Activation de l'intégration

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. Pour plus d'informations sur les actions AWS IoT SiteWise prises en charge, consultez la section [Actions définies par AWS IoT SiteWise](#) dans la référence d'autorisation de service.

Pour plus d'informations sur les rôles AWS IoT TwinMaker liés à un service, consultez la section [Rôles liés à un service AWS IoT TwinMaker dans le Guide de l'utilisateur](#). AWS IoT TwinMaker

Avant de pouvoir intégrer AWS IoT SiteWise et AWS IoT TwinMaker, vous devez accorder les autorisations suivantes qui permettent l'intégration AWS IoT SiteWise à un espace de travail AWS IoT TwinMaker lié :

- **iotsitewise:EnableSiteWiseIntegration**— Permet AWS IoT SiteWise de s'intégrer à un AWS IoT TwinMaker espace de travail lié. Cette intégration permet AWS IoT TwinMaker de lire toutes vos informations de modélisation AWS IoT SiteWise via un rôle AWS IoT TwinMaker lié à un service. Pour activer cette autorisation, ajoutez la politique suivante à votre rôle IAM :

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

Intégration AWS IoT SiteWise et AWS IoT TwinMaker

Pour intégrer AWS IoT SiteWise et AWS IoT TwinMaker, vous devez disposer des éléments suivants :

- AWS IoT SiteWise rôle lié au service configuré dans votre compte

- AWS IoT TwinMaker rôle lié au service configuré dans votre compte
- AWS IoT TwinMaker espace de travail avec identifiant IoTSiteWiseDefaultWorkspace dans votre compte dans la région.

Pour intégrer à l'aide de la AWS IoT SiteWise console

Lorsque la AWS IoT TwinMaker bannière Intégration avec apparaît dans la console, choisissez Accorder l'autorisation. Les prérequis sont créés dans votre compte.

Pour effectuer l'intégration à l'aide du AWS CLI

Pour intégrer AWS IoT SiteWise et AWS IoT TwinMaker en utilisant le AWS CLI, entrez les commandes suivantes :

1. Appelez `CreateServiceLinkedRole` avec un `AWSServiceName` de `iotsitewise.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iotsitewise.amazonaws.com
```

2. Appelez `CreateServiceLinkedRole` avec un `AWSServiceName` de `iottwinmaker.amazonaws.com`.

```
aws iam create-service-linked-role --aws-service-name iottwinmaker.amazonaws.com
```

3. Appelez `CreateWorkspace` avec un ID de `IoTSiteWiseDefaultWorkspace`.

```
aws iottwinmaker create-workspace --workspace-id IoTSiteWiseDefaultWorkspace
```

Détectez les anomalies avec Lookout for Equipment

Note

La détection des anomalies n'est disponible que dans les régions où Amazon Lookout for Equipment est disponible.

Vous pouvez intégrer AWS IoT SiteWise Amazon Lookout for Equipment pour obtenir des informations sur votre équipement industriel grâce à la détection des anomalies et à la maintenance

prédictive des équipements industriels. Lookout for Equipment est un service d'apprentissage automatique (ML) pour la surveillance des équipements industriels qui détecte le comportement anormal des équipements et identifie les défaillances potentielles. Lookout for Equipment vous permet de mettre en œuvre des programmes de maintenance prédictive et d'identifier les processus sous-optimaux liés aux équipements. Pour plus d'informations sur Lookout for Equipment, [consultez l'article Qu'est-ce qu'Amazon Lookout](#) for Equipment ? dans le guide de l'utilisateur d'Amazon Lookout for Equipment.

Lorsque vous créez une prédiction pour entraîner un modèle d'apprentissage automatique afin de détecter le comportement anormal de l'équipement, vous envoyez les valeurs AWS IoT SiteWise des propriétés des actifs à Lookout for Equipment pour entraîner un modèle d'apprentissage automatique afin de détecter le comportement anormal de l'équipement. Pour définir une définition de prédiction sur un modèle d'actif, vous devez spécifier les rôles IAM nécessaires pour que Lookout for Equipment accède à vos données et les propriétés à envoyer à Lookout for Equipment et à envoyer les données traitées à Amazon S3. Pour de plus amples informations, veuillez consulter [Créez des modèles d'actifs dans AWS IoT SiteWise](#).

Pour intégrer AWS IoT SiteWise Lookout for Equipment à Lookout for Equipment, vous devez suivre les étapes de haut niveau suivantes :

- Ajoutez une définition de prédiction sur un modèle d'actif qui décrit les propriétés que vous souhaitez suivre. La définition de prédiction est un ensemble réutilisable de mesures, de transformations et de métriques qui est utilisé pour créer des prédictions sur les actifs basées sur ce modèle d'actif.
- Entraînez la prédiction en fonction des données historiques que vous fournissez.
- Planifier l'inférence, qui indique à AWS IoT SiteWise quelle fréquence une prédiction spécifique doit être exécutée.

Une fois l'inférence planifiée, le modèle Lookout for Equipment surveille les données qu'il reçoit de votre équipement et recherche les anomalies de comportement de l'équipement. Vous pouvez consulter et analyser les résultats dans SiteWise Monitor, à l'aide des opérations de l'API AWS IoT SiteWise GET ou de la console Lookout for Equipment. Vous pouvez également créer des alarmes à l'aide des détecteurs d'alarme du modèle d'équipement pour vous avertir en cas de comportement anormal de l'équipement.

Rubriques

- [Ajouter une définition de prédiction \(console\)](#)

- [Entraînez une prédiction \(console\)](#)
- [Démarrer ou arrêter l'inférence sur une prédiction \(console\)](#)
- [Ajouter une définition de prédiction \(CLI\)](#)
- [Entraînez une prédiction et démarrez l'inférence \(CLI\)](#)
- [Entraînez une prédiction \(CLI\)](#)
- [Démarrer ou arrêter l'inférence sur une prédiction \(CLI\)](#)

Ajouter une définition de prédiction (console)

Pour commencer à envoyer les données collectées par AWS IoT SiteWise Lookout for Equipment, vous devez ajouter AWS IoT SiteWise une définition de prédiction à un modèle d'actif.

Pour ajouter une définition de prédiction à un modèle AWS IoT SiteWise d'actif

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Models et sélectionnez le modèle de ressource auquel vous souhaitez ajouter la définition de prédiction.
3. Choisissez Prédiction.
4. Choisissez Ajouter une définition de prédiction.
5. Définissez les détails de la définition de la prédiction.
 - a. Entrez un nom unique et une description pour la définition de votre prédiction. Choisissez le nom avec soin, car une fois que vous avez créé la définition de prédiction, vous ne pouvez pas le modifier.
 - b. Créez ou sélectionnez un rôle d'autorisation IAM qui permet de AWS IoT SiteWise partager les données de vos actifs avec Amazon Lookout for Equipment. Le rôle doit respecter les politiques IAM et de confiance suivantes. Pour obtenir de l'aide sur la création du rôle, voir [Création d'un rôle à l'aide de politiques de confiance personnalisées \(console\)](#).

Politique IAM

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "L4EPermissions",
    "Effect": "Allow",
    "Action": [
```

```

        "lookoutequipment:CreateDataset",
        "lookoutequipment:CreateModel",
        "lookoutequipment:CreateInferenceScheduler",
        "lookoutequipment:DescribeDataset",
        "lookoutequipment:DescribeModel",
        "lookoutequipment:DescribeInferenceScheduler",
        "lookoutequipment:ListInferenceExecutions",
        "lookoutequipment:StartDataIngestionJob",
        "lookoutequipment:StartInferenceScheduler",
        "lookoutequipment:UpdateInferenceScheduler",
        "lookoutequipment:StopInferenceScheduler"
    ],
    "Resource": [
        "arn:aws:lookoutequipment:Region:Account_ID:inference-
scheduler/IoTSiteWise_*",
        "arn:aws:lookoutequipment:Region:Account_ID:model/
IoTSiteWise_*",
        "arn:aws:lookoutequipment:Region:Account_ID:dataset/
IoTSiteWise_*"
    ]
},
{
    "Sid": "L4EPermissions2",
    "Effect": "Allow",
    "Action": [
        "lookoutequipment:DescribeDataIngestionJob"
    ],
    "Resource": "*"
},
{
    "Sid": "S3Permissions",
    "Effect": "Allow",
    "Action": [
        "s3:CreateBucket",
        "s3:ListBucket",
        "s3:PutObject",
        "s3:GetObject"
    ],
    "Resource": ["arn:aws:s3:::iotsitewise-*"]
},
{
    "Sid": "IAMPermissions",
    "Effect": "Allow",
    "Action": [

```

```

        "iam:GetRole",
        "iam:PassRole"
    ],
    "Resource": "arn:aws:iam::Account_ID:role/Role_name"
}
]
}

```

Politique d'approbation

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "Account_ID"
      },
      "ArnEquals": {
        "aws:SourceArn":
"arn:aws:iotsitewise:Region:Account_ID:asset/*"
      }
    }
  },
  {
    "Effect": "Allow",
    "Principal": {
      "Service": "lookoutequipment.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceAccount": "Account_ID"
      },
      "ArnEquals": {
        "aws:SourceArn":
"arn:aws:lookoutequipment:Region:Account_ID:*"
      }
    }
  }
}

```

```
}  
  ]  
}
```

- c. Choisissez Suivant.
6. Sélectionnez les attributs de données (mesures, transformations et métriques) que vous souhaitez envoyer à Lookout for Equipment.
 - a. (Facultatif) Sélectionnez les mesures.
 - b. (Facultatif) Sélectionnez les transformations.
 - c. (Facultatif) Sélectionnez les métriques.
 - d. Choisissez Suivant.
7. Passez en revue vos sélections. Pour ajouter la définition de prédiction au modèle de ressource, sur la page de résumé, choisissez Ajouter une définition de prédiction.

Vous pouvez également modifier ou supprimer une définition de prédiction existante à laquelle sont associées des prédictions actives.

Entraînez une prédiction (console)

Après avoir ajouté une définition de prédiction à un modèle d'actifs, vous pouvez entraîner les prédictions relatives à vos actifs.

Pour entraîner une prédiction dans AWS IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Assets, puis sélectionnez l'actif que vous souhaitez surveiller.
3. Choisissez Prédiction.
4. Sélectionnez les prédictions que vous souhaitez entraîner.
5. Sous Actions, choisissez Commencer l'entraînement, puis procédez comme suit :
 - a. Sous Détails de la prédiction, sélectionnez un rôle d'autorisation IAM qui permet de AWS IoT SiteWise partager les données de vos actifs avec Lookout for Equipment. Si vous devez créer un nouveau rôle, choisissez Créer un nouveau rôle.
 - b. Pour les paramètres des données d'entraînement, entrez une plage temporelle des données d'entraînement pour sélectionner les données à utiliser pour entraîner la prédiction.

- c. (Facultatif) Sélectionnez le taux d'échantillonnage des données après le post-traitement.
 - d. (Facultatif) Pour les étiquettes de données, fournissez un compartiment Amazon S3 et un préfixe contenant vos données d'étiquetage. Pour plus d'informations sur les données d'étiquetage, consultez la section [Étiquetage de vos données](#) dans le guide de l'utilisateur d'Amazon Lookout for Equipment.
 - e. Choisissez Suivant.
6. (Facultatif) Si vous souhaitez que la prédiction soit active dès la fin de l'entraînement, sous Paramètres avancés, sélectionnez Activer automatiquement la prédiction après l'entraînement, puis procédez comme suit :
- a. Sous Données d'entrée, pour Fréquence de téléchargement des données, définissez la fréquence à laquelle les données sont téléchargées, et pour Temps de décalage, définissez la quantité de mémoire tampon à utiliser.
 - b. Choisissez Suivant.
7. Passez en revue les détails de la prédiction et choisissez Enregistrer et démarrer.

Démarrer ou arrêter l'inférence sur une prédiction (console)

Note

Les frais de Lookout for Equipment s'appliquent aux inférences planifiées avec les données transférées AWS IoT SiteWise entre Lookout for Equipment et Lookout for Equipment. Pour plus d'informations, consultez les [tarifs d'Amazon Lookout for Equipment](#).

Si vous avez ajouté la prédiction `lookoutequipment:CreateDataset`, mais que vous n'avez pas choisi de l'activer après l'entraînement, vous devez l'activer pour commencer à surveiller vos actifs.

Pour démarrer l'inférence d'une prédiction

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Assets, puis sélectionnez l'actif auquel la prédiction est ajoutée.
3. Choisissez Prédictions.
4. Sélectionnez les prédictions que vous souhaitez activer.

5. Sous Actions, choisissez Démarrer l'inférence, puis procédez comme suit :
 - a. Sous Données d'entrée, pour Fréquence de téléchargement des données, définissez la fréquence à laquelle les données sont téléchargées, et pour Temps de décalage, définissez la quantité de mémoire tampon à utiliser.
 - b. Choisissez Enregistrer et commencez.

Pour arrêter l'inférence pour une prédiction

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Assets, puis sélectionnez l'actif auquel la prédiction est ajoutée.
3. Choisissez Prédictions.
4. Sélectionnez les prédictions que vous souhaitez arrêter.
5. Sous Actions, choisissez Arrêter l'inférence.

Ajouter une définition de prédiction (CLI)

Pour définir une définition de prédiction sur un modèle d'actif nouveau ou existant, vous pouvez utiliser le AWS Command Line Interface (AWS CLI). Après avoir défini la définition de prédiction sur le modèle d'actif, vous entraînez et planifiez l'inférence pour une prédiction sur un actif AWS IoT SiteWise afin de détecter les anomalies avec Lookout for Equipment.

Prérequis

Pour effectuer ces étapes, vous devez avoir créé un modèle d'actif et au moins un actif. Pour plus d'informations, consultez [Création d'un modèle d'actifs \(AWS CLI\)](#) et [Création d'un actif \(AWS CLI\)](#).

Si vous débutez dans ce AWS IoT SiteWise domaine, vous devez appeler l'opération `CreateBulkImportJob` API pour y importer les valeurs des propriétés des actifs AWS IoT SiteWise, qui seront utilisées pour entraîner le modèle. Pour de plus amples informations, veuillez consulter [Création d'une tâche d'importation AWS IoT SiteWise en bloc \(AWS CLI\)](#).

Pour ajouter une définition de prédiction

1. Créez un fichier, appelé `asset-model-payload.json`. Suivez les étapes décrites dans ces autres sections pour ajouter les détails de votre modèle d'actif au fichier, mais ne soumettez pas de demande de création ou de mise à jour du modèle d'actif.

- Pour plus d'informations sur la création d'un modèle d'actifs, voir [Création d'un modèle d'actifs \(AWS CLI\)](#)
 - Pour plus d'informations sur la mise à jour d'un modèle d'actif existant, voir [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#)
2. Ajoutez un modèle composite Lookout for Equipment `assetModelCompositeModels ()` au modèle d'actif en ajoutant le code suivant.
- **Property** Remplacez-le par l'ID des propriétés que vous souhaitez inclure. Pour les obtenir IDs, appelez-moi [DescribeAssetModel](#).
 - **RoleARN** Remplacez-le par l'ARN d'un rôle IAM qui permet à Lookout for Equipment d'accéder AWS IoT SiteWise à vos données.

```
{
  ...
  "assetModelCompositeModels": [
    {
      "name": "L4Epredictiondefinition",
      "type": "AWS/L4E_ANOMALY",
      "properties": [
        {
          "name": "AWS/L4E_ANOMALY_RESULT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_RESULT",
          "unit": "none",
          "type": {
            "measurement": {}
          }
        },
        {
          "name": "AWS/L4E_ANOMALY_INPUT",
          "dataType": "STRUCT",
          "dataTypeSpec": "AWS/L4E_ANOMALY_INPUT",
          "type": {
            "attribute": {
              "defaultValue": "{\"properties\": [\"Property1\", \"Property2\"]}"
            }
          }
        }
      ]
    }
  ]
}
```

```

    "name": "AWS/L4E_ANOMALY_PERMISSIONS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_PERMISSIONS",
    "type": {
      "attribute": {
        "defaultValue": "{\"roleArn\": \"RoleARN\"}"
      }
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_DATASET",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_DATASET",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_MODEL",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_MODEL",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_INFERENCE",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE",
    "type": {
      "attribute": {}
    }
  },
  {
    "name": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_TRAINING_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  }
},
{

```

```

    "name": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "dataType": "STRUCT",
    "dataTypeSpec": "AWS/L4E_ANOMALY_INFERENCE_STATUS",
    "type": {
      "attribute": {
        "defaultValue": "{}"
      }
    }
  }
]
}

```

3. Créez le modèle d'actif ou mettez à jour le modèle d'actif existant. Effectuez l'une des actions suivantes :

- Pour créer le modèle d'actif, exécutez la commande suivante :

```
aws iotsitewise create-asset-model --cli-input-json file://asset-model-payload.json
```

- Pour mettre à jour le modèle d'actif existant, exécutez la commande suivante. *asset-model-id* Remplacez-le par l'ID du modèle d'actif que vous souhaitez mettre à jour.

```
aws iotsitewise update-asset-model \
  --asset-model-id asset-model-id \
  --cli-input-json file://asset-model-payload.json
```

Après avoir exécuté la commande, notez le `assetModelId` dans la réponse.

Entraînez une prédiction et démarrez l'inférence (CLI)

Maintenant que la définition de la prédiction est définie, vous pouvez entraîner les actifs en fonction de celle-ci et commencer l'inférence. Si vous souhaitez affiner votre prédiction sans commencer l'inférence, passez directement à [Entraînez une prédiction \(CLI\)](#). Pour entraîner la prédiction et commencer à inférer sur l'actif, vous aurez besoin `assetId` de celui de la ressource cible.

Pour entraîner et démarrer l'inférence de la prédiction

1. Exécutez la commande suivante pour trouver le `assetModelCompositeModelId` dessous `assetModelCompositeModelSummaries`. *asset-model-id* Remplacez-le par

l'ID du modèle d'actif que vous avez créé dans [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Exécutez la commande suivante pour trouver `actionDefinitionId` l'`TrainingWithInference` action. Remplacez *asset-model-id* par l'ID utilisé à l'étape précédente et remplacez *asset-model-composite-model-id* par l'ID renvoyé à l'étape précédente.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Créez un fichier appelé `train-start-inference-prediction.json` et ajoutez-y le code suivant, en remplaçant le suivant :
 - *asset-id* avec l'ID de l'actif cible
 - *action-definition-id* avec l'identifiant de l' `TrainingWithInference` action
 - *StartTime* avec le début des données d'entraînement, fournies en secondes
 - *EndTime* avec les données de fin d'entraînement, fournies en secondes d'époque
 - *TargetSamplingRate* avec le taux d'échantillonnage des données après post-traitement par Lookout for Equipment. Les valeurs autorisées sont : PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload": {
    "stringValue": "{\"l4ETrainingWithInference\":{\"trainingWithInferenceMode\": \"START\", \"trainingPayload\": {\"exportDataStartTime\": StartTime, \"exportDataEndTime\": EndTime}, \"targetSamplingRate\": \"TargetSamplingRate\", \"inferencePayload\": {\"dataDelayOffsetInMinutes\": 0, \"dataUploadFrequency\": \"PT5M\"}}}"
  }
}
```

}

4. Exécutez la commande suivante pour démarrer l'entraînement et l'inférence :

```
aws iotsitewise execute-action --cli-input-json file://train-start-inference-
prediction.json
```

Entraînez une prédiction (CLI)

Maintenant que la définition de la prédiction est définie, vous pouvez entraîner les actifs en fonction de celle-ci. Pour entraîner la prédiction sur l'actif, vous aurez besoin `assetId` de celui de la ressource cible.

Pour entraîner la prédiction

1. Exécutez la commande suivante pour trouver le `assetModelCompositeModelId` dessous `assetModelCompositeModelSummaries`. *asset-model-id* Remplacez-le par l'ID du modèle d'actif que vous avez créé dans [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

- Exécutez la commande suivante pour trouver `actionDefinitionId` l'`Trainingaction`. Remplacez `asset-model-id` par l'ID utilisé à l'étape précédente et remplacez `asset-model-composite-model-id` par l'ID renvoyé à l'étape précédente.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Créez un fichier appelé `train-prediction.json` et ajoutez-y le code suivant, en remplaçant le suivant :
 - *asset-id* avec l'ID de l'actif cible
 - *action-definition-id* avec l'identifiant de l'action de formation
 - *StartTime* avec le début des données d'entraînement, fournies en secondes
 - *EndTime* avec les données de fin d'entraînement, fournies en secondes d'époque

- (Facultatif) *BucketName* avec le nom du compartiment Amazon S3 qui contient vos données d'étiquette
- (Facultatif) *Prefix* avec le préfixe associé au compartiment Amazon S3.
- *TargetSamplingRate* avec le taux d'échantillonnage des données après post-traitement par Lookout for Equipment. Les valeurs autorisées sont :PT1S | PT5S | PT10S | PT15S | PT30S | PT1M | PT5M | PT10M | PT15M | PT30M | PT1H.

 Note

Incluez à la fois le nom et le préfixe du compartiment, ou aucun des deux.

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-Id",
  "actionPayload":{ "stringValue": "{\\"l4ETraining\\": {\\"trainingMode\\":
\\"START\\",\\"exportDataStartTime\\": StartTime, \\"exportDataEndTime\\": EndTime,
\\"targetSamplingRate\\":\\"TargetSamplingRate\\", \\"labelInputConfiguration\\":
{\\"bucketName\\": \\"BucketName\\", \\"prefix\\": \\"Prefix\\"}}}"
```

4. Exécutez la commande suivante pour démarrer l'entraînement :

```
aws iotsitewise execute-action --cli-input-json file://train-prediction.json
```

Avant de pouvoir commencer l'inférence, vous devez suivre une formation. Pour vérifier le statut de la formation, effectuez l'une des opérations suivantes :

- Depuis la console, naviguez jusqu'à l'actif sur lequel porte la prédiction.
- À partir du AWS CLI, appelez BatchGetAssetPropertyValue en utilisant le `propertyId` nom de la `trainingStatus` propriété.

Démarrer ou arrêter l'inférence sur une prédiction (CLI)

Une fois la prédiction établie, vous pouvez commencer l'inférence pour demander à Lookout for Equipment de commencer à surveiller vos actifs. Pour démarrer ou arrêter l'inférence, vous aurez besoin `assetId` de la ressource cible.

Pour démarrer l'inférence

1. Exécutez la commande suivante pour trouver le `assetModelCompositeModelId` `dessousassetModelCompositeModelSummaries`. *asset-model-id* Remplacez-le par l'ID du modèle d'actif que vous avez créé dans [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --output format \  
  --query query
```

2. Exécutez la commande suivante pour trouver `actionDefinitionId` l'`Inferenceaction`. Remplacez *asset-model-id* par l'ID utilisé à l'étape précédente et remplacez *asset-model-composite-model-id* par l'ID renvoyé à l'étape précédente.

```
aws iotsitewise describe-asset-model-composite-model \  
  --asset-model-id asset-model-id \  
  --asset-model-composite-model-id asset-model-composite-model-id \  
  --output format \  
  --query query
```

3. Créez un fichier appelé `start-inference.json` et ajoutez-y le code suivant, en remplaçant le suivant :

- *asset-id* avec l'ID de l'actif cible
- *action-definition-id* avec l'ID de l'action d'inférence de départ
- *Offset* avec la quantité de mémoire tampon à utiliser
- *Frequency* avec la fréquence à laquelle les données sont téléchargées

```
{  
  "targetResource": {  
    "assetId": "asset-id"  
  },  
  "actionDefinitionId": "action-definition-id",  
  "actionPayload": {  
    "stringValue": "{ \"inferenceMode\": \"START\",  
    \"dataDelayOffsetInMinutes\": Offset,  
    \"dataUploadFrequency\": \"Frequency\" } }"
```

```
}}
```

4. Exécutez la commande suivante pour démarrer l'inférence :

```
aws iotsitewise execute-action --cli-input-json file://start-inference.json
```

Pour arrêter l'inférence

1. Exécutez la commande suivante pour trouver le `assetModelCompositeModelId` dessous `assetModelCompositeModelSummaries`. *asset-model-id* Remplacez-le par l'ID du modèle d'actif que vous avez créé dans [Mettre à jour un modèle d'actif ou de composant \(AWS CLI\)](#).

```
aws iotsitewise describe-asset-model \
  --asset-model-id asset-model-id \
```

2. Exécutez la commande suivante pour trouver `actionDefinitionId` l'`Inferenceaction`. Remplacez *asset-model-id* par l'ID utilisé à l'étape précédente et remplacez *asset-model-composite-model-id* par l'ID renvoyé à l'étape précédente.

```
aws iotsitewise describe-asset-model-composite-model \
  --asset-model-id asset-model-id \
  --asset-model-composite-model-id asset-model-composite-model-id \
```

3. Créez un fichier appelé `stop-inference.json` et ajoutez-y le code suivant, en remplaçant le suivant :

- *asset-id* avec l'ID de l'actif cible
- *action-definition-id* avec l'ID de l'action d'inférence de départ

```
{
  "targetResource": {
    "assetId": "asset-id"
  },
  "actionDefinitionId": "action-definition-id",
  "actionPayload": { "stringValue": "{\"l4EInference\":{\"inferenceMode\":\"STOP\"}}"}
}
```


4. Exécutez la commande suivante pour arrêter l'inférence :

```
aws iotsitewise execute-action --cli-input-json file://stop-inference.json
```

Gérez le stockage des données dans AWS IoT SiteWise

Vous pouvez configurer AWS IoT SiteWise pour enregistrer vos données dans les niveaux de stockage suivants :

Niveau chaud

Le niveau de stockage à chaud est un stockage de séries chronologiques AWS IoT SiteWise géré. Le mode Hot Tier est particulièrement efficace pour les données fréquemment consultées, avec une faible write-to-read latence. Les données stockées dans le hot tier sont utilisées par les applications industrielles qui ont besoin d'un accès rapide aux dernières valeurs de mesure de votre équipement. Cela inclut les applications qui visualisent les métriques en temps réel à l'aide d'un tableau de bord interactif, ou les applications qui surveillent les opérations et lancent des alarmes pour identifier les problèmes de performance.

Par défaut, les données ingérées sont AWS IoT SiteWise stockées dans le hot tier. Vous pouvez définir une période de conservation pour le niveau chaud, après quoi les AWS IoT SiteWise données du niveau chaud sont transférées vers le niveau de stockage chaud ou froid, en fonction de votre configuration. Pour optimiser les performances et la rentabilité, définissez la période de rétention de votre hot tier de manière à ce qu'elle soit plus longue que le temps nécessaire pour récupérer souvent les données. Ceci est utilisé pour les métriques en temps réel, les alarmes et les scénarios de surveillance. Si aucune période de conservation n'est définie, vos données sont stockées indéfiniment dans le hot tier.

Niveau chaud

Le niveau de stockage à chaud est un niveau AWS IoT SiteWise géré efficace pour le stockage rentable des données historiques. Il est préférable de l'utiliser pour récupérer de gros volumes de données présentant des caractéristiques de write-to-read latence moyenne. Utilisez le niveau chaud pour stocker les données historiques nécessaires aux charges de travail importantes. Par exemple, il est utilisé pour la récupération de données pour les analyses, les applications de business intelligence (BI), les outils de reporting et la formation de modèles d'apprentissage automatique (ML). Si vous activez le niveau de stockage à froid, vous pouvez définir une période de rétention du niveau chaud. Une fois la période de conservation terminée, AWS IoT SiteWise les données sont supprimées du niveau chaud.

Niveau froid

Le niveau de stockage à froid utilise un compartiment Amazon S3 pour stocker des données rarement utilisées. Lorsque le niveau froid est activé, AWS IoT SiteWise reproduit les séries

chronologiques, y compris les mesures, les métriques, les transformations et les agrégats, ainsi que les définitions des modèles d'actifs toutes les 6 heures. Le niveau froid est utilisé pour stocker des données qui tolèrent une latence de lecture élevée pour les rapports historiques et les sauvegardes.

Rubriques

- [Configurer les paramètres de stockage dans AWS IoT SiteWise](#)
- [Résoudre les problèmes liés aux paramètres de stockage pour AWS IoT SiteWise](#)
- [Chemins de fichiers et schémas de données enregistrés dans le niveau froid](#)

Configurer les paramètres de stockage dans AWS IoT SiteWise

Vous pouvez configurer les paramètres de stockage pour opter pour le stockage de niveau chaud géré par service et également pour répliquer les données vers le niveau froid. Pour en savoir plus sur la durée de conservation des niveaux chaud et froid, consultez [Impact sur la conservation des données](#). Lors de la configuration des paramètres de stockage, procédez comme suit :

- **Rétention de niveau chaud** : définissez une période de conservation pour la durée pendant laquelle vos données sont stockées dans le niveau chaud avant d'être supprimées, puis transférées vers le stockage de niveau chaud ou le stockage de niveau froid géré par le service en fonction de vos paramètres de stockage. AWS IoT SiteWise supprimera toutes les données du hot tier qui existaient avant la fin de la période de conservation. Si vous ne définissez pas de période de conservation, vos données sont stockées indéfiniment dans le hot tier.
- **Rétention au niveau chaud** : définissez une période de conservation pour la durée pendant laquelle vos données sont stockées dans le niveau chaud avant qu'elles ne soient supprimées du AWS IoT SiteWise stockage et transférées vers le stockage à froid géré par le client. AWS IoT SiteWise supprime toutes les données du niveau chaud qui existaient avant la fin de la période de rétention. Si aucune période de conservation n'est définie, vos données sont stockées indéfiniment dans le niveau chaud.

Note

Pour améliorer les performances des requêtes, définissez une période de rétention de niveau chaud avec un stockage de niveau chaud.

Impact de la conservation des données dans les systèmes de stockage à chaud et à chaud

- Lorsque vous réduisez la durée de conservation du niveau de stockage chaud, les données sont définitivement déplacées du niveau chaud vers le niveau chaud ou froid. Lorsque vous réduisez la durée de conservation du niveau chaud, les données sont déplacées vers le niveau froid et définitivement supprimées du niveau chaud.
- Lorsque vous augmentez la durée de conservation du niveau de stockage chaud ou chaud, la modification affecte les données qui sont ensuite envoyées AWS IoT SiteWise . AWS IoT SiteWise ne récupère pas les données du stockage chaud ou froid pour alimenter le niveau chaud. Par exemple, si la période de conservation du stockage de niveau chaud est initialement fixée à 30 jours, puis portée à 60 jours, il faut 30 jours pour que le stockage de niveau chaud contienne 60 jours de données.

Rubriques

- [Configurer les paramètres de stockage pour Warm Tier \(console\)](#)
- [Configurer les paramètres de stockage pour Warm Tier \(AWS CLI\)](#)
- [Configuration des paramètres de stockage pour le niveau froid \(console\)](#)
- [Configurer les paramètres de stockage pour Cold Tier \(AWS CLI\)](#)

Configurer les paramètres de stockage pour Warm Tier (console)

La procédure suivante explique comment configurer les paramètres de stockage pour répliquer les données vers le niveau chaud de la AWS IoT SiteWise console.

Pour configurer les paramètres de stockage dans la console

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, sous Paramètres, choisissez Stockage.
3. Dans le coin supérieur droit, choisissez Modifier.
4. Sur la page Modifier le stockage, procédez comme suit :
5. Pour les paramètres Hot Tier, procédez comme suit :
 - Si vous souhaitez définir une période de conservation correspondant à la durée pendant laquelle vos données sont stockées dans le niveau chaud avant leur suppression et leur

transfert vers le stockage à chaud géré par le service, choisissez Activer la période de conservation.

- Pour configurer une période de rétention, entrez un nombre entier et choisissez une unité. La durée de conservation doit être supérieure ou égale à 30 jours.

AWS IoT SiteWise supprime toutes les données du hot tier qui sont antérieures à la période de conservation. Si vous ne définissez pas de période de conservation, vos données sont stockées indéfiniment.

6. (Recommandé) Pour les paramètres du niveau Warm, procédez comme suit :

- Pour opter pour le stockage à niveau chaud, sélectionnez Je confirme pour activer le stockage à niveau chaud pour opter pour le stockage à niveau chaud.
- (Facultatif) Pour configurer une période de rétention, entrez un nombre entier et choisissez une unité. La durée de conservation doit être supérieure ou égale à 365 jours.

AWS IoT SiteWise supprime les données du niveau chaud qui existaient avant la période de rétention. Si vous ne définissez pas de période de conservation, vos données sont stockées indéfiniment.

 Note

- Lorsque vous optez pour le niveau chaud, la configuration ne s'affiche qu'une seule fois.
- Pour définir le niveau de rétention à chaud, vous devez disposer d'un niveau de stockage à chaud ou à froid. Pour des raisons de rentabilité et de récupération des données historiques, il est AWS IoT SiteWise recommandé de stocker les données à long terme dans le niveau chaud.
- Pour définir le niveau de rétention à chaud, vous devez disposer d'un niveau de stockage à froid.

7. Choisissez Enregistrer pour enregistrer vos paramètres de stockage.

Dans la section AWS IoT SiteWise stockage, le stockage de niveau Warm se trouve dans l'un des états suivants :

- **Activé** : si vos données existaient avant la période de rétention du niveau chaud, AWS IoT SiteWise déplacez-les vers le niveau chaud. »
- **Désactivé** : le stockage à chaud est désactivé.

Configurer les paramètres de stockage pour Warm Tier (AWS CLI)

Vous pouvez configurer les paramètres de stockage pour déplacer les données vers le niveau chaud à l' AWS CLI aide des commandes suivantes.

Pour éviter de remplacer la configuration existante, récupérez les informations de configuration de stockage actuelles en exécutant la commande suivante :

```
aws iotsitewise describe-storage-configuration
```

Exemple réponse sans configuration de niveau froid existante

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "configurationStatus": {
    "state": "ACTIVE"
  },
  "lastUpdateDate": "2021-10-14T15:53:35-07:00",
  "warmTier": "DISABLED"
}
```

Exemple réponse avec la configuration de niveau froid existante

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": retention-in-days
  }
}
```

```
    },
    "configurationStatus": {
      "state": "ACTIVE"
    },
    "lastUpdateDate": "2023-10-25T15:59:46-07:00",
    "warmTier": "DISABLED"
  }
```

Configurez les paramètres de stockage pour le niveau chaud avec AWS CLI

Exécutez la commande suivante pour configurer les paramètres de stockage. Remplacez file-name par le nom du fichier contenant la configuration AWS IoT SiteWise de stockage.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Exemple AWS IoT SiteWise configuration avec niveau chaud et niveau chaud

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "disassociatedDataStorage": "ENABLED",
  "warmTier": "ENABLED",
  "retentionPeriod": {
    "numberOfDays": hot-tier-retention-in-days
  }
}
```

hot-tier-retention-in-days doit être un nombre entier supérieur ou égal à 30 jours.

Exemple réponse

```
{
  "storageType": "SITEWISE_DEFAULT_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

Si le stockage à froid est activé, consultez [Configuration des paramètres de stockage avec un AWS CLI niveau froid existant](#).

Configuration des paramètres de stockage avec un AWS CLI niveau froid existant

Configuration des paramètres de stockage à l' AWS CLI aide du stockage à froid existant

- Exécutez la commande suivante pour configurer les paramètres de stockage. Remplacez *file-name* par le nom du fichier contenant la configuration AWS IoT SiteWise de stockage.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Exemple AWS IoT SiteWise configuration de stockage

- amzn-s3-demo-bucket* Remplacez-le par le nom de votre compartiment Amazon S3.
- prefix* Remplacez-le par votre préfixe Amazon S3.
- Remplacez *aws-account-id* par votre identifiant de AWS compte.
- Remplacez *role-name* par le nom du rôle d'accès Amazon S3 qui permet AWS IoT SiteWise d'envoyer des données à Amazon S3.
- Remplacez *hot-tier-retention-in-days* par un nombre entier supérieur ou égal à 30 jours.
- Remplacez *warm-tier-retention-in-days* par un nombre entier supérieur ou égal à 365 jours.

Note

AWS IoT SiteWise supprimera toutes les données du niveau chaud qui sont antérieures à la période de conservation du niveau froid. Si vous ne définissez pas de période de conservation, vos données sont stockées indéfiniment.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix/",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "disassociatedDataStorage": "ENABLED",
```



```
"retentionPeriod": {
  "numberOfDays": hot-tier-retention-in-days
},
"warmTier": "ENABLED",
"warmTierRetentionPeriod": {
  "numberOfDays": warm-tier-retention-in-days
}
}
```

Exemple réponse

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}
```

Configuration des paramètres de stockage pour le niveau froid (console)

La procédure suivante explique comment configurer les paramètres de stockage pour répliquer les données vers le niveau froid de la AWS IoT SiteWise console.

Pour configurer les paramètres de stockage dans la console

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, sous Paramètres, choisissez Stockage.
3. Dans le coin supérieur droit, choisissez Modifier.
4. Sur la page Modifier le stockage, procédez comme suit :
 - a. Pour les paramètres de stockage, choisissez Activer le stockage à froid. Le stockage à froid est désactivé par défaut.
 - b. Pour l'emplacement du compartiment S3, entrez le nom d'un compartiment Amazon S3 existant et un préfixe.

 Note

- Amazon S3 utilise le préfixe comme nom de dossier dans le compartiment Amazon S3. Le préfixe doit comporter de 1 à 255 caractères et se terminer par une barre oblique (/). Vos AWS IoT SiteWise données sont enregistrées dans ce dossier.
- Si vous n'avez pas de compartiment Amazon S3, choisissez View, puis créez-en un dans la console Amazon S3. Pour plus d'informations, consultez la section [Création de votre premier compartiment S3](#) dans le guide de l'utilisateur Amazon S3.

c. Pour le rôle d'accès S3, effectuez l'une des opérations suivantes :

- Choisissez Create a role from an AWS managed template, puis créez AWS automatiquement un rôle IAM qui permet d' AWS IoT SiteWise envoyer des données à Amazon S3.
- Choisissez Utiliser un rôle existant, puis choisissez le rôle que vous avez créé dans la liste.

 Note

- Vous devez utiliser le même nom de compartiment Amazon S3 pour l'emplacement du compartiment S3 que celui que vous avez utilisé à l'étape précédente et dans votre politique IAM.
- Assurez-vous que votre rôle dispose des autorisations indiquées dans l'exemple suivant.

Exemple politique d'autorisations :

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

```
        "s3:DeleteObject",
        "s3:GetBucketLocation",
        "s3:ListBucket"
    ],
    "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket",
        "arn:aws:s3:::amzn-s3-demo-bucket/*"
    ]
}
]
```

Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment Amazon S3.

- Si le compartiment Amazon S3 est chiffré à l'aide d'une clé KMS gérée par le client, la clé KMS doit avoir une politique d'accès avec un rôle IAM pour les `kms:GenerateDataKey` opérations `kms:Decrypt` et les opérations.

- d. Pour configurer le hot tier, reportez-vous à l'étape 5 de [Configurer les paramètres de stockage pour Warm Tier \(console\)](#).
- e. (Facultatif) Pour AWS IoT Analytics l'intégration, procédez comme suit.

 Note

Avis de fin de support : le 15 décembre 2025, AWS le support de AWS IoT Analytics. Après le 15 décembre 2025, vous ne pourrez plus accéder à la AWS IoT Analytics console ni aux AWS IoT Analytics ressources. Pour plus d'informations, voir [AWS IoT Analytics fin du support](#).

- i. Si vous souhaitez l'utiliser AWS IoT Analytics pour interroger vos données, choisissez Enabled AWS IoT Analytics data store.
- ii. AWS IoT SiteWise génère un nom pour votre banque de données ou vous pouvez saisir un autre nom.

AWS IoT SiteWise crée automatiquement un magasin de données AWS IoT Analytics pour enregistrer vos données. Pour interroger les données, vous pouvez AWS IoT Analytics créer

des ensembles de données. Pour plus d'informations, consultez la section [Utilisation des AWS IoT SiteWise données](#) dans le Guide de AWS IoT Analytics l'utilisateur.

- f. Choisissez Enregistrer.

Dans la section AWS IoT SiteWise de stockage, le stockage à froid peut prendre l'une des valeurs suivantes :

- **Activé** : AWS IoT SiteWise réplique vos données dans le compartiment Amazon S3 spécifié.
- **Activation** : AWS IoT SiteWise traite votre demande pour activer le stockage à froid. Ce processus peut prendre plusieurs minutes.
- **Enable_Failed** : AWS IoT SiteWise impossible de traiter votre demande d'activation du stockage à froid. Si vous avez activé AWS IoT SiteWise l'envoi de journaux vers Amazon CloudWatch Logs, vous pouvez utiliser ces journaux pour résoudre les problèmes. Pour de plus amples informations, veuillez consulter [Surveillez avec Amazon CloudWatch Logs](#).
- **Désactivé** : le stockage à froid est désactivé.

Configurer les paramètres de stockage pour Cold Tier (AWS CLI)

La procédure suivante explique comment configurer les paramètres de stockage pour répliquer les données vers le niveau froid à l'aide AWS CLI de.

Pour configurer les paramètres de stockage à l'aide de AWS CLI

1. Pour exporter des données vers un compartiment Amazon S3 de votre compte, exécutez la commande suivante pour configurer les paramètres de stockage. Remplacez *file-name* par le nom du fichier contenant la configuration AWS IoT SiteWise de stockage.

```
aws iotsitewise put-storage-configuration --cli-input-json file://file-name.json
```

Exemple AWS IoT SiteWise configuration de stockage

- *amzn-s3-demo-bucket* Remplacez-le par le nom de votre compartiment Amazon S3.
- *prefix* Remplacez-le par votre préfixe Amazon S3.
- Remplacez *aws-account-id* par votre identifiant de AWS compte.
- Remplacez *role-name* par le nom du rôle d'accès Amazon S3 qui permet AWS IoT SiteWise d'envoyer des données à Amazon S3.

- Remplacez *retention-in-days* par un nombre entier supérieur ou égal à 30 jours.

```
{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {
      "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/prefix",
      "roleArn": "arn:aws:iam::aws-account-id:role/role-name"
    }
  },
  "retentionPeriod": {
    "numberOfDays": retention-in-days,
    "unlimited": false
  }
}
```

Note

- Vous devez utiliser le même nom de compartiment Amazon S3 dans la configuration de AWS IoT SiteWise stockage et dans la politique IAM.
- Assurez-vous que votre rôle dispose des autorisations indiquées dans l'exemple suivant.

Exemple politique d'autorisations :

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

```

    ]
  }
]
}

```

Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment Amazon S3.

- Si le compartiment Amazon S3 est chiffré à l'aide d'une clé KMS gérée par le client, la clé KMS doit avoir une politique d'accès avec un rôle IAM pour les `kms:GenerateDataKey` opérations `kms:Decrypt` et les opérations.

Exemple réponse

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "retentionPeriod": {
    "numberOfDays": 100,
    "unlimited": false
  },
  "configurationStatus": {
    "state": "UPDATE_IN_PROGRESS"
  }
}

```

Note

La mise à jour de la configuration AWS IoT SiteWise de stockage peut prendre quelques minutes.

2. Pour récupérer les informations de configuration du stockage, exécutez la commande suivante.

```
aws iotsitewise describe-storage-configuration
```

Exemple réponse

```

{
  "storageType": "MULTI_LAYER_STORAGE",
  "multiLayerStorage": {
    "customerManagedS3Storage": {

```

```

        "s3ResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket/torque/",
        "roleArn": "arn:aws:iam::123456789012:role/SWAccessS3Role"
    },
    "retentionPeriod": {
        "numberOfDays": 100,
        "unlimited": false
    },
    "configurationStatus": {
        "state": "ACTIVE"
    },
    "lastUpdateDate": "2021-03-30T15:54:14-07:00"
}

```

3. Pour arrêter d'exporter des données vers le compartiment Amazon S3, exécutez la commande suivante pour configurer les paramètres de stockage.

```
aws iotsitewise put-storage-configuration --storage-type SITEWISE_DEFAULT_STORAGE
```

Note

Par défaut, vos données ne sont stockées que dans le niveau chaud de AWS IoT SiteWise.

Exemple réponse

```

{
    "storageType": "SITEWISE_DEFAULT_STORAGE",
    "configurationStatus": {
        "state": "UPDATE_IN_PROGRESS"
    }
}

```

4. Pour récupérer les informations de configuration du stockage, exécutez la commande suivante.

```
aws iotsitewise describe-storage-configuration
```

Exemple réponse

```
{
```

```
"storageType": "SITEWISE_DEFAULT_STORAGE",
"configurationStatus": {
  "state": "ACTIVE"
},
"lastUpdateDate": "2021-03-30T15:57:14-07:00"
}
```

(Facultatif) Créez un magasin de AWS IoT Analytics données (AWS CLI)

Note

Avis de fin de support : le 15 décembre 2025, AWS le support de AWS IoT Analytics. Après le 15 décembre 2025, vous ne pourrez plus accéder à la AWS IoT Analytics console ni aux AWS IoT Analytics ressources. Pour plus d'informations, voir [AWS IoT Analytics fin du support](#).

Un magasin de AWS IoT Analytics données est un référentiel évolutif et interrogeable qui reçoit et stocke des données. Vous pouvez utiliser la AWS IoT SiteWise console ou AWS IoT Analytics APIs créer un magasin de AWS IoT Analytics données pour enregistrer vos AWS IoT SiteWise données. Pour interroger les données, vous créez des ensembles de données à l'aide AWS IoT Analytics de. Pour plus d'informations, consultez la section [Utilisation des AWS IoT SiteWise données](#) dans le Guide de AWS IoT Analytics l'utilisateur.

Les étapes suivantes permettent AWS CLI de créer un magasin de données dans AWS IoT Analytics.

Pour créer un magasin de données, exécutez la commande suivante. Remplacez *file-name* par le nom du fichier contenant la configuration du magasin de données.

```
aws iotanalytics create-datastore --cli-input-json file://file-name.json
```

Note

- Vous devez spécifier le nom d'un compartiment Amazon S3 existant. Si vous n'avez pas de compartiment Amazon S3, créez-en un d'abord. Pour plus d'informations, consultez la section [Création de votre premier compartiment S3](#) dans le guide de l'utilisateur Amazon S3.

- Vous devez utiliser le même nom de compartiment Amazon S3 dans la configuration du AWS IoT SiteWise stockage, la politique IAM et la configuration du magasin de AWS IoT Analytics données.

Exemple AWS IoT Analytics configuration du magasin de données

Remplacez *data-store-name* et *amzn-s3-demo-bucket* par le nom de votre banque de AWS IoT Analytics données et le nom du compartiment Amazon S3.

```
{
  "datastoreName": "data-store-name",
  "datastoreStorage": {
    "iotSiteWiseMultiLayerStorage": {
      "customerManagedS3Storage": {
        "bucket": "amzn-s3-demo-bucket"
      }
    }
  },
  "retentionPeriod": {
    "numberOfDays": 90
  }
}
```

Exemple réponse

```
{
  "datastoreName": "datastore_IoTSiteWise_demo",
  "datastoreArn": "arn:aws:iotanalytics:us-west-2:123456789012:datastore/
datastore_IoTSiteWise_demo",
  "retentionPeriod": {
    "numberOfDays": 90,
    "unlimited": false
  }
}
```

Résoudre les problèmes liés aux paramètres de stockage pour AWS IoT SiteWise

Utilisez les informations suivantes pour résoudre les problèmes liés à la configuration du stockage.

Problèmes

- [Erreur : le compartiment n'existe pas](#)
- [Erreur : accès refusé au chemin Amazon S3](#)
- [Erreur : l'ARN du rôle ne peut pas être assumé](#)
- [Erreur : Impossible d'accéder au compartiment Amazon S3 interrégional](#)

Erreur : le compartiment n'existe pas

Solution : AWS IoT SiteWise impossible de trouver votre compartiment Amazon S3. Assurez-vous de saisir le nom d'un compartiment Amazon S3 existant dans la région actuelle.

Erreur : accès refusé au chemin Amazon S3

Solution : AWS IoT SiteWise impossible d'accéder à votre compartiment Amazon S3. Procédez comme suit :

- Assurez-vous d'utiliser le même compartiment Amazon S3 que celui que vous avez spécifié dans la politique IAM.
- Assurez-vous que votre rôle dispose des autorisations indiquées dans l'exemple suivant.

Exemple stratégie d'autorisation

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

```
}
```

Remplacez `amzn-s3-demo-bucket` par le nom de votre compartiment Amazon S3.

Erreur : l'ARN du rôle ne peut pas être assumé

Solution : AWS IoT SiteWise ne peut pas assumer le rôle IAM en votre nom. Assurez-vous que votre rôle fait confiance au service suivant : `iotsitewise.amazonaws.com`. Pour plus d'informations, voir [Je ne peux pas assumer un rôle, voir le](#) guide de l'utilisateur IAM.

Erreur : Impossible d'accéder au compartiment Amazon S3 interrégional

Solution : Le compartiment Amazon S3 que vous avez spécifié se trouve dans une autre AWS région. Assurez-vous que votre compartiment Amazon S3 et vos AWS IoT SiteWise actifs se trouvent dans la même région.

Chemins de fichiers et schémas de données enregistrés dans le niveau froid

AWS IoT SiteWise stocke vos données dans la couche froide en répliquant des séries chronologiques, notamment des mesures, des métriques, des transformations et des agrégats, ainsi que des définitions d'actifs et de modèles d'actifs. Ce qui suit décrit les chemins de fichiers et les schémas de données envoyés au niveau froid.

Rubriques

- [Données relatives à l'équipement \(mesures\)](#)
- [Métriques, transformations et agrégats](#)
- [Métadonnées des actifs](#)
- [Métadonnées de hiérarchie des actifs](#)
- [Fichiers d'index des données de stockage](#)

Données relatives à l'équipement (mesures)

AWS IoT SiteWise exporte les données de l'équipement (mesures) vers le niveau froid une fois toutes les six heures. Les données brutes sont enregistrées dans le niveau froid au format [Apache AVRO](#) (`.avro`).

Chemin d'accès du fichier

AWS IoT SiteWise stocke les données de l'équipement (mesures) dans le niveau froid à l'aide du modèle suivant.

```
{keyPrefix}/raw/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/  
seriesBucket={seriesBucket}/raw_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Chaque chemin de fichier vers les données brutes dans Amazon S3 contient les composants suivants.

Composant Path	Description
keyPrefix	Le préfixe Amazon S3 que vous avez spécifié dans la configuration AWS IoT SiteWise de stockage. Amazon S3 utilise le préfixe comme nom de dossier dans le compartiment.
raw	Le dossier qui stocke les séries chronologiques des équipements (mesures). Le raw dossier est enregistré dans le dossier des préfixes.
seriesBucket	Nombre hexadécimal compris entre 00 et ff. Ce numéro est dérivé de <code>timeSeriesId</code>. Cette partition est utilisée pour augmenter le débit lors des AWS IoT SiteWise écritures sur le niveau froid. Lorsque vous utilisez Amazon Athena pour exécuter des requêtes, vous pouvez utiliser la partition pour un partitionnement précis afin d'améliorer les performances des requêtes. <code>seriesBucket</code> et <code>timeSeriesBucket</code> dans les métadonnées des actifs figurent le même numéro.
startYear	L'année de l'heure de début exclusive associée aux données de la série chronologique.

Composant Path	Description
startMonth	Le mois de l'heure de début exclusive associée aux données de la série chronologique.
startDay	Le jour du mois de l'heure de début exclusive associée aux données de la série chronologique.
fileName	Le nom du fichier utilise le trait de soulignement (_) comme séparateur pour séparer les éléments suivants : • Le raw préfixe. • La <code>timeSeriesId</code> valeur. • L'horodatage de l'heure de début exclusive associée aux données de la série chronologique. • La qualité des données. Valeurs valides : GOODBAD, etUNCERTAIN . Pour plus d'informations, consultez AssetPropertyValue dans la Référence d'API AWS IoT SiteWise . Le fichier est enregistré au .avro format à l'aide de la compression Snappy.

Exemple chemin du fichier vers les données brutes dans le niveau froid

```
keyPrefix/raw/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/
raw_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-
bc6f-1b490154b07a_1609577700_GOOD.avro
```

Champs

Le schéma des données brutes exportées vers le niveau froid contient les champs suivants.

AWS IoT SiteWise conseille aux clients de mettre en œuvre le support pour l'évolution des schémas sur les systèmes qui lisent les données brutes depuis le niveau froid, car des champs supplémentaires pourraient être introduits à l'avenir.

Les données nulles sont représentées par le fait que tous les champs de valeur sont nuls. Toutefois, les clients recevront toujours le type de données correct lorsqu'ils effectueront des AWS IoT SiteWise APIs requêtes.

Nom de champ	Types pris en charge	Type par défaut	Description
seriesId	string	N/A	L'identifiant qui identifie les séries chronologiques de l'équipement (mesures). Vous pouvez utiliser ce champ pour joindre des données brutes et des métadonnées d'actifs dans des requêtes.
timeInSeconds	long	N/A	Date d'horodatage, en secondes, au format Unix Epoch. Les données de nanosecondes fractionnaires sont fournies par. offsetInNanos
offsetInNanos	long	N/A	Le décalage de nanosecondes par rapport à. timeInSeconds

Nom de champ	Types pris en charge	Type par défaut	Description
quality	string	N/A	Qualité de la valeur de la série chronologique.
doubleValue	double ou null	null	Données de séries chronologiques de type double (nombre à virgule flottante).
stringValue	string ou null	null	Données de séries chronologiques de type chaîne (séquence de caractères).
integerValue	int ou null	null	Données de séries chronologiques de type entier (nombre entier).
booleanValue	boolean ou null	null	Données de séries chronologiques de type booléen (vrai ou faux).
jsonValue	string ou null	null	Données de séries chronologiques de type JSON (types de données complexes stockés sous forme de chaîne).

Nom de champ	Types pris en charge	Type par défaut	Description
recordVersion	long ou null	null	Le numéro de version de l'enregistrement. Vous pouvez utiliser le numéro de version pour sélectionner le dernier enregistrement. Les nouveaux enregistrements ont des numéros de version plus grands.

Exemple données brutes dans la couche froide

```
{
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675887,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.75
  },
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 1
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675889,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.69
  },
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 2
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675890,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.66
  },
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 3
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675891,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.92
  },
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 4
}, {
  "seriesId": "e9687d2a-0dbe-4f65-9ed6-6f443cba41f7_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeInSeconds": 1625675892,
  "offsetInNanos": 0,
  "quality": "GOOD",
  "doubleValue": {
    "double": 0.73
  },
  "stringValue": null,
  "integerValue": null,
  "booleanValue": null,
  "jsonValue": null,
  "recordVersion": 5
}
```

Métriques, transformations et agrégats

AWS IoT SiteWise exporte les métriques, les transformations et les agrégats vers le niveau froid une fois toutes les six heures. Les métriques, les transformations et les agrégats sont enregistrés dans le niveau froid au format [Apache AVRO](#) (.avro).

Chemin d'accès du fichier

AWS IoT SiteWise stocke les métriques, les transformations et les agrégats dans le niveau froid à l'aide du modèle suivant.

```
{keyPrefix}/agg/startYear={startYear}/startMonth={startMonth}/startDay={startDay}/  
seriesBucket={seriesBucket}/agg_{timeseriesId}_{startTimestamp}_{quality}.avro
```

Chaque chemin de fichier vers les métriques, les transformations et les agrégats dans Amazon S3 contient les composants suivants.

Composant Path	Description
keyPrefix	Le préfixe Amazon S3 que vous avez spécifié dans la configuration AWS IoT SiteWise de stockage. Amazon S3 utilise le préfixe comme nom de dossier dans le compartiment.
agg	Le dossier qui stocke les données de séries chronologiques issues des métriques. Le agg dossier est enregistré dans le dossier des préfixes.
seriesBucket	Nombre hexadécimal compris entre 00 et ff. Ce numéro est dérivé de <code>timeSeriesId</code>. Cette partition est utilisée pour augmenter le débit lors des AWS IoT SiteWise écritures sur le niveau froid. Lorsque vous utilisez Amazon Athena pour exécuter des requêtes, vous pouvez utiliser la partition pour un partitionnement précis afin d'améliorer les performances des requêtes. <code>seriesBucket</code> et <code>timeSeriesBucket</code> dans les métadonnées des actifs figurent le même numéro.

Composant Path	Description
startYear	L'année de l'heure de début exclusive associée aux données de la série chronologique.
startMonth	Le mois de l'heure de début exclusive associée aux données de la série chronologique.
startDay	Le jour du mois de l'heure de début exclusive associée aux données de la série chronologique.
fileName	Le nom du fichier utilise le trait de soulignement (_) comme séparateur pour séparer les éléments suivants : • Le raw préfixe. • La timeSeriesId valeur. • L'horodatage de l'heure de début exclusive associée aux données de la série chronologique. • La qualité des données. Valeurs valides : GOODBAD, etUNCERTAIN . Pour plus d'informations, consultez AssetPropertyValue dans la Référence d'API AWS IoT SiteWise . Le fichier est enregistré au .avro format à l'aide de la compression Snappy.

Exemple chemin du fichier vers les métriques dans le niveau froid

```
keyPrefix/agg/startYear=2021/startMonth=1/startDay=2/seriesBucket=a2/agg_7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a_1609577700_GOOD.avro
```

Champs

Le schéma des métriques, des transformations et des agrégats exportés vers le niveau froid contient les champs suivants.

Nom de champ	Types pris en charge	Type par défaut	Description
seriesId	string	N/A	L'identifiant qui identifie les séries chronologiques issues de l'équipement, des métriques ou des transformations. Vous pouvez utiliser ce champ pour joindre des données brutes et des métadonnées d'actifs dans des requêtes.
timeInSeconds	long	N/A	Date d'horodatage, en secondes, au format Unix Epoch. Les données de nanoseconde fractionnaires sont fournies par. <code>offsetInNanos</code>
offsetInNanos	long	N/A	Le décalage de nanosecondes par rapport à. <code>timeInSeconds</code>
quality	string	N/A	La qualité selon laquelle les données

Nom de champ	Types pris en charge	Type par défaut	Description
			relatives aux actifs doivent être filtrées.
resolution	string	N/A	Intervalle de temps pendant lequel les données doivent être agrégées.
count	double ou null	null	Le nombre total de points de données pour les variables données sur l'intervalle de temps actuel.
average	double ou null	null	La moyenne des valeurs des variables données sur l'intervalle de temps actuel.
min	double ou null	null	Le minimum des valeurs des variables données sur l'intervalle de temps actuel.
max	boolean ou null	null	Le maximum des valeurs des variables données sur l'intervalle de temps actuel.
sum	string ou null	null	Somme des valeurs des variables données sur l'intervalle de temps actuel.

Nom de champ	Types pris en charge	Type par défaut	Description
recordVersion	long ou null	null	Le numéro de version de l'enregistrement. Vous pouvez utiliser le numéro de version pour sélectionner le dernier enregistrement. Les nouveaux enregistrements ont des numéros de version plus grands.

Exemple Données métriques dans la couche froide

```

{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334060,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":496.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334120,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":1334.0},"min":{"double":32.0},"max":{"double":60.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334540,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":496.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637334600,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":1334.0},"min":{"double":32.0},"max":{"double":60.0},"recordVersion":null}
{"seriesId":"f74c2828-5317-4df3-ba16-6d41b5bcb531","timeInSeconds":1637335020,"offsetInNanos":0,"quality":"GOOD","resolution":1000,"sum":{"double":496.0},"min":{"double":1.0},"max":{"double":31.0},"recordVersion":null}

```

Métadonnées des actifs

Lorsque vous activez l'exportation de données AWS IoT SiteWise vers le niveau froid pour la première fois, les métadonnées des actifs sont exportées vers le niveau froid. Après la configuration initiale, AWS IoT SiteWise exporte les métadonnées des actifs vers le niveau uniquement lorsque vous modifiez les définitions des modèles d'actifs ou les définitions des actifs. Les métadonnées des actifs sont enregistrées dans le niveau froid au format JSON (.ndjson) délimité par une nouvelle ligne.

Chemin d'accès du fichier

AWS IoT SiteWise stocke les métadonnées des actifs dans le niveau froid à l'aide du modèle suivant.

```
{keyPrefix}/asset_metadata/asset_{assetId}.ndjson
```

Chaque chemin de fichier vers les métadonnées des actifs dans le niveau froid contient les composants suivants.

Composant Path	Description
keyPrefix	Le préfixe Amazon S3 que vous avez spécifié dans la configuration de stockage AWS IoT SiteWise s. Amazon S3 utilise le préfixe comme nom de dossier dans le compartiment.
asset_metadata	Le dossier qui stocke les métadonnées des actifs. Le asset_metadata dossier est enregistré dans le dossier des préfixes.
fileName	Le nom du fichier utilise le trait de soulignement (_) comme séparateur pour séparer les éléments suivants : • Le asset préfixe. • La assetId valeur. Le fichier est enregistré au .ndjson format.

Exemple chemin du fichier vers les métadonnées des actifs dans le niveau le plus froid

keyPrefix/asset_metadata/asset_35901915-d476-4dca-8637-d9ed4df939ed.ndjson

Champs

Le schéma des métadonnées des actifs exportées vers le niveau froid contient les champs suivants.

Nom de champ	Description
assetId	ID de la ressource .
assetName	Le nom de l'actif.
assetExternalId	ID externe de la ressource.
assetModelId	ID du modèle d'actif utilisé pour créer cet actif.
assetModelName	Nom du modèle d'actif.
assetModelExternalId	ID externe du modèle d'actif.
assetPropertyId	ID de la propriété de ressource.
assetPropertyName	Nom de la propriété de l'actif.
assetPropertyExternalId	ID externe de la propriété de l'actif.
assetPropertyDataType	Type de données de la propriété de l'actif.
assetPropertyUnit	Unité de la propriété de l'actif (par exemple, Newtons etRPM).
assetPropertyAlias	Alias qui identifie la propriété de l'actif, tel que le chemin de flux de données d'un serveur OPC UA (par exemple,/company/windfarm/3/turbine/7/temperature).
timeSeriesId	L'identifiant qui identifie les séries chronologiques issues de l'équipement, des métriques ou des transformations. Vous pouvez utiliser ce

Nom de champ	Description
	champ pour joindre des données brutes et des métadonnées d'actifs dans des requêtes.
<code>timeSeriesBucket</code>	Nombre hexadécimal compris entre 00 et ff. Ce numéro est dérivé de <code>timeSeriesId</code> . Cette partition est utilisée pour augmenter le débit lors des AWS IoT SiteWise écritures sur le niveau froid. Lorsque vous utilisez Amazon Athena pour exécuter des requêtes, vous pouvez utiliser la partition pour un partitionnement précis afin d'améliorer les performances des requêtes. <code>timeSeriesBucket</code> et <code>seriesBucket</code> le chemin du fichier vers les données brutes contient le même numéro.
<code>assetCompositeModelId</code>	ID du modèle composite.
<code>assetCompositeModelExternalId</code>	ID externe du modèle composite.
<code>assetCompositeModelDescription</code>	Description du modèle composite.
<code>assetCompositeModelName</code>	Nom du modèle composite.
<code>assetCompositeModelType</code>	Type du modèle composite. Pour les modèles composites d'alarme, ce type est <code>AWS/ALARM</code> .
<code>assetCreationDate</code>	Date de création de la ressource, à l'époque Unix.
<code>assetLastUpdateDate</code>	Date à laquelle la ressource a été mise à jour pour la dernière fois, à l'époque Unix.
<code>assetStatusErrorCode</code>	Code de l'erreur.
<code>assetStatusErrorMessage</code>	Message d'erreur.

Nom de champ	Description
assetStatusState	État actuel de l'actif.

Exemple métadonnées des actifs dans le niveau froid

```
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Temperature",
  "assetPropertyData": {
    "value": 100,
    "unit": "Celsius",
    "type": "DOUBLE"
  },
  "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_95e63da7-d34e-43e1-bc6f-1b490154b07a",
  "timeSeriesBucket": "f6",
  "assetArn": null,
  "assetCompositeModelDescription": null
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "c706d54d-4c11-42dc-9a01-63662fc697b4",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Pressure",
  "assetPropertyData": {
    "value": 100,
    "unit": "Pascals",
    "type": "DOUBLE"
  },
  "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_c706d54d-4c11-42dc-9a01-63662fc697b4",
  "timeSeriesBucket": "1e",
  "assetArn": null,
  "assetCompositeModelDescription": null
},
{
  "assetId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d",
  "assetExternalId": null,
  "assetName": "Wind Turbine Asset 2",
  "assetModelId": "ec1d924f-f07d-444f-b072-e2994c165d35",
  "assetModelExternalId": null,
  "assetModelName": "Wind Turbine Asset Model",
  "assetPropertyId": "8cf1162f-dead-4fbe-b468-c8e24cde9f50",
  "assetPropertyExternalId": null,
  "assetPropertyName": "Max Temperature",
  "assetPropertyDataType": "DOUBLE",
  "assetPropertyUnit": null,
  "assetPropertyAlias": null,
  "timeSeriesId": "7020c8e2-e6db-40fa-9845-ed0ddddd4c77d_8cf1162f-dead-4fbe-b468-c8e24cde9f50",
  "timeSeriesBucket": "d7",
  "assetArn": null,
  "assetCompositeModelDescription": null,
  "assetStatusState": "ON"
},
{
  "assetId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab",
  "assetExternalId": null,
  "assetName": "BatchAssessment",
  "assetModelExternalId": null,
  "assetModelName": "FlashTestAssetModelDouble",
  "assetPropertyId": "ab401a9176ed",
  "assetPropertyExternalId": null,
  "assetPropertyName": "measurementProperty",
  "assetPropertyData": {
    "value": 100,
    "unit": "Celsius",
    "type": "DOUBLE"
  },
  "timeSeriesId": "3a5f2a22-3b37-4332-9c1c-404ea1d73fab_ab401a9176ed",
  "timeSeriesBucket": "af",
  "assetArn": null,
  "assetCompositeModelDescription": null,
  "assetStatusState": "ON"
}
```

Métadonnées de hiérarchie des actifs

Lorsque vous activez AWS IoT SiteWise l'enregistrement des données dans le niveau froid pour la première fois, les métadonnées de la hiérarchie des actifs sont exportées vers le niveau froid. Après la configuration initiale, AWS IoT SiteWise exporte les métadonnées de la hiérarchie des actifs vers le niveau froid uniquement lorsque vous modifiez le modèle d'actif ou les définitions des actifs. Les métadonnées de la hiérarchie des actifs sont enregistrées dans le niveau froid au format JSON (.ndjson) délimité par une nouvelle ligne.

Un identifiant externe pour la hiérarchie, la ressource cible ou la ressource source est récupéré en appelant l'[DescribeAssetAPI](#).

Chemin d'accès du fichier

AWS IoT SiteWise stocke les métadonnées de la hiérarchie des actifs dans le niveau froid à l'aide du modèle suivant.

```
{keyPrefix}/asset_hierarchy_metadata/{parentAssetId}_{hierarchyId}.ndjson
```

Chaque chemin de fichier vers les métadonnées de la hiérarchie des actifs dans le niveau froid contient les composants suivants.

Composant Path	Description
keyPrefix	Le préfixe Amazon S3 que vous avez spécifié dans la configuration AWS IoT SiteWise de stockage. Amazon S3 utilise le préfixe comme nom de dossier dans le compartiment.
asset_hierarchy_metadata	Le dossier qui stocke les métadonnées de la hiérarchie des actifs. Le asset_hierarchy_metadata dossier est enregistré dans le dossier des préfixes.
fileName	Le nom du fichier utilise le trait de soulignement (_) comme séparateur pour séparer les éléments suivants : La parentAssetId valeur.

Composant Path	Description
	La hierarchyId valeur. Le fichier est enregistré au .ndjson format.

Exemple chemin du fichier vers les métadonnées de la hiérarchie des actifs dans le niveau froid

```
keyPrefix/asset_hierarchy_metadata/35901915-d476-4dca-8637-
d9ed4df939ed_c5b3ced8-589a-48c7-9998-cdcccfc9747a0.ndjson
```

Champs

Le schéma des métadonnées de hiérarchie des actifs exportées vers le niveau froid contient les champs suivants.

Nom de champ	Description
sourceAssetId	L'ID de l'actif source dans cette relation d'actif.
targetAssetId	L'ID de l'actif cible dans cette relation d'actif.
hierarchyId	ID de la hiérarchie.
associationType	Type d'association de cette relation patrimoniale. La valeur doit êtreCHILD. L'actif cible est un actif enfant de l'actif source.

Exemple métadonnées de hiérarchie des actifs dans le niveau froid

```
{
  "sourceAssetId": "80388e72-2284-44fb-9c89-bfbaf0dfedd2",
  "targetAssetId": "2b866c25-0c74-4750-bdf5-b73683c8a2a2",
  "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee",
  "associationType": "CHILD"
}
```

```
bc0b-470ea47d1e31", "hierarchyId": "bbed9f59-0412-4585-a61d-6044db526aee", "associationType": "CHILD"}
```

Pour consulter vos données en mode froid

1. Accédez à la [console Amazon S3](#).
2. Dans le volet de navigation, choisissez Buckets, puis choisissez votre compartiment Amazon S3.
3. Accédez au dossier contenant les données brutes, les métadonnées des actifs ou les métadonnées de la hiérarchie des actifs.
4. Sélectionnez les fichiers, puis dans Actions, choisissez Télécharger.

Fichiers d'index des données de stockage

AWS IoT SiteWise utilise ces fichiers pour optimiser les performances des requêtes de données. Ils apparaissent dans votre compartiment Amazon S3, mais vous n'avez pas besoin de les utiliser.

Chemin d'accès du fichier

AWS IoT SiteWise stocke les fichiers d'index de données dans le niveau froid en utilisant le modèle suivant.

```
keyPrefix/index/series=timeseriesId/startYear=startYear/startMonth=startMonth/  
startDay=startDay/index_timeseriesId_startTimestamp_quality
```

Exemple chemin du fichier vers le fichier d'index de stockage des données

```
keyPrefix/index/series=7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-  
d34e-43e1-bc6f-1b490154b07a/startYear=2022/startMonth=02/startDay=03/  
index_7020c8e2-e6db-40fa-9845-ed0dddd4c77d_95e63da7-d34e-43e1-  
bc6f-1b490154b07a_1643846400_G00D
```

Exemples de code pour AWS IoT SiteWise l'utilisation AWS SDKs

Les exemples de code suivants montrent comment utiliser AWS IoT SiteWise un kit de développement AWS logiciel (SDK).

Les principes de base sont des exemples de code qui vous montrent comment effectuer les opérations essentielles au sein d'un service.

Les actions sont des extraits de code de programmes plus larges et doivent être exécutées dans leur contexte. Alors que les actions vous indiquent comment appeler des fonctions de service individuelles, vous pouvez les voir en contexte dans leurs scénarios associés.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Mise en route

Bonjour AWS IoT SiteWise

Les exemples de code suivants montrent comment démarrer avec AWS IoT SiteWise.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public class HelloSitewise {
    private static final Logger logger =
        LoggerFactory.getLogger(HelloSitewise.class);
    public static void main(String[] args) {
        fetchAssetModels();
    }
}
```

```

/**
 * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
 */
public static void fetchAssetModels() {
    IoTSiteWiseAsyncClient siteWiseAsyncClient =
IoTSiteWiseAsyncClient.create();
    ListAssetModelsRequest assetModelsRequest =
ListAssetModelsRequest.builder()
        .assetModelTypes(AssetModelType.ASSET_MODEL)
        .build();

    // Asynchronous paginator - process paginated results.
    ListAssetModelsPublisher listModelsPaginator =
siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
    CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
        response.assetModelSummaries().forEach(assetSummary ->
            logger.info("Asset Model Name: {} ", assetSummary.name())
        );
    });

    // Wait for the asynchronous operation to complete
    future.join();
}
}

```

- Pour plus de détails sur l'API, voir [ListAssetModels](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [réfrentiel d'exemples de code AWS](#).

```
import {
```

```
paginateListAssetModels,
IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new IoTSiteWiseClient();
  const listAssetModelsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
    for await (const page of paginator) {
      listAssetModelsPaginated.push(...page.assetModelSummaries);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }
  for (const { name, creationDate } of listAssetModelsPaginated) {
    console.log(`${name} - ${creationDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Pour plus de détails sur l'API, voir [ListAssetModels](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
    object wraps
                                the low-level AWS IoT SiteWise service API.
    """
    print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
    paginator = iot_sitewise_client.get_paginator("list_asset_models")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

    asset_model_names: [str] = []
    for page in page_iterator:
        for asset_model in page["assetModelSummaries"]:
            asset_model_names.append(asset_model["name"])

    print(f"{len(asset_model_names)} asset model(s) retrieved.")
    for asset_model_name in asset_model_names:
        print(f"\t{asset_model_name}")

if __name__ == "__main__":
    hello_iot_sitewise(boto3.client("iotsitewise"))
```


- Pour plus de détails sur l'API, consultez [ListAssetModels](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Exemples de code

- [Exemples de base pour AWS IoT SiteWise l'utilisation AWS SDKs](#)
 - [Bonjour AWS IoT SiteWise](#)
 - [Apprenez les bases AWS IoT SiteWise d'un AWS SDK](#)
 - [Actions d' AWS IoT SiteWise utilisation AWS SDKs](#)
 - [Utilisation BatchPutAssetPropertyValue avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateAsset avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateAssetModel avec un AWS SDK ou une CLI](#)
 - [Utilisation CreateGateway avec un AWS SDK ou une CLI](#)
 - [Utilisation CreatePortal avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteAsset avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteAssetModel avec un AWS SDK ou une CLI](#)
 - [Utilisation DeleteGateway avec un AWS SDK ou une CLI](#)
 - [Utilisation DeletePortal avec un AWS SDK ou une CLI](#)
 - [Utilisation DescribeAssetModel avec un AWS SDK ou une CLI](#)
 - [Utilisation DescribeGateway avec un AWS SDK ou une CLI](#)
 - [Utilisation DescribePortal avec un AWS SDK ou une CLI](#)
 - [Utilisation GetAssetPropertyValue avec un AWS SDK ou une CLI](#)
 - [Utilisation ListAssetModels avec un AWS SDK ou une CLI](#)

Exemples de base pour AWS IoT SiteWise l'utilisation AWS SDKs

Les exemples de code suivants montrent comment utiliser les principes de base de AWS IoT SiteWise with AWS SDKs.

Exemples

- [Bonjour AWS IoT SiteWise](#)
- [Apprenez les bases AWS IoT SiteWise d'un AWS SDK](#)
- [Actions d' AWS IoT SiteWise utilisation AWS SDKs](#)

- [Utilisation BatchPutAssetPropertyValue avec un AWS SDK ou une CLI](#)
- [Utilisation CreateAsset avec un AWS SDK ou une CLI](#)
- [Utilisation CreateAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation CreateGateway avec un AWS SDK ou une CLI](#)
- [Utilisation CreatePortal avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteAsset avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteGateway avec un AWS SDK ou une CLI](#)
- [Utilisation DeletePortal avec un AWS SDK ou une CLI](#)
- [Utilisation DescribeAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation DescribeGateway avec un AWS SDK ou une CLI](#)
- [Utilisation DescribePortal avec un AWS SDK ou une CLI](#)
- [Utilisation GetAssetPropertyValue avec un AWS SDK ou une CLI](#)
- [Utilisation ListAssetModels avec un AWS SDK ou une CLI](#)

Bonjour AWS IoT SiteWise

Les exemples de code suivants montrent comment démarrer avec AWS IoT SiteWise.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet [GitHub](#). Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
public class HelloSitewise {  
    private static final Logger logger =  
        LoggerFactory.getLogger(HelloSitewise.class);  
    public static void main(String[] args) {  
        fetchAssetModels();  
    }  
}
```

```
/**
 * Fetches asset models using the provided {@link IoTSiteWiseAsyncClient}.
 */
public static void fetchAssetModels() {
    IoTSiteWiseAsyncClient siteWiseAsyncClient =
IoTSiteWiseAsyncClient.create();
    ListAssetModelsRequest assetModelsRequest =
ListAssetModelsRequest.builder()
        .assetModelTypes(AssetModelType.ASSET_MODEL)
        .build();

    // Asynchronous paginator - process paginated results.
    ListAssetModelsPublisher listModelsPaginator =
siteWiseAsyncClient.listAssetModelsPaginator(assetModelsRequest);
    CompletableFuture<Void> future = listModelsPaginator.subscribe(response -
> {
        response.assetModelSummaries().forEach(assetSummary ->
            logger.info("Asset Model Name: {} ", assetSummary.name())
        );
    });

    // Wait for the asynchronous operation to complete
    future.join();
}
}
```

- Pour plus de détails sur l'API, voir [ListAssetModels](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  paginateListAssetModels,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";

// Call ListDocuments and display the result.
export const main = async () => {
  const client = new IoTSiteWiseClient();
  const listAssetModelsPaginated = [];
  console.log(
    "Hello, AWS Systems Manager! Let's list some of your documents:\n",
  );
  try {
    // The paginate function is a wrapper around the base command.
    const paginator = paginateListAssetModels({ client }, { maxResults: 5 });
    for await (const page of paginator) {
      listAssetModelsPaginated.push(...page.assetModelSummaries);
    }
  } catch (caught) {
    console.error(`There was a problem saying hello: ${caught.message}`);
    throw caught;
  }
  for (const { name, creationDate } of listAssetModelsPaginated) {
    console.log(`${name} - ${creationDate}`);
  }
};

// Call function if run directly.
import { fileURLToPath } from "node:url";
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  main();
}
```

- Pour plus de détails sur l'API, voir [ListAssetModels](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import boto3

def hello_iot_sitewise(iot_sitewise_client):
    """
    Use the AWS SDK for Python (Boto3) to create an AWS IoT SiteWise
    client and list the asset models in your account.
    This example uses the default settings specified in your shared credentials
    and config files.

    :param iot_sitewise_client: A Boto3 AWS IoT SiteWise Client object. This
    object wraps
                                the low-level AWS IoT SiteWise service API.
    """
    print("Hello, AWS IoT SiteWise! Let's list some of your asset models:\n")
    paginator = iot_sitewise_client.get_paginator("list_asset_models")
    page_iterator = paginator.paginate(PaginationConfig={"MaxItems": 10})

    asset_model_names: [str] = []
    for page in page_iterator:
        for asset_model in page["assetModelSummaries"]:
            asset_model_names.append(asset_model["name"])

    print(f"{len(asset_model_names)} asset model(s) retrieved.")
    for asset_model_name in asset_model_names:
        print(f"\t{asset_model_name}")

if __name__ == "__main__":
    hello_iot_sitewise(boto3.client("iotsitewise"))
```

- Pour plus de détails sur l'API, consultez [ListAssetModels](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Apprenez les bases AWS IoT SiteWise d'un AWS SDK

Les exemples de code suivants montrent comment :

- Créez un modèle AWS IoT SiteWise d'actifs.
- Créez un AWS IoT SiteWise actif.
- Récupérez les valeurs d'ID de propriété.
- Envoyez des données à un AWS IoT SiteWise actif.
- Récupérez la valeur de la propriété AWS IoT SiteWise Asset.
- Créez un AWS IoT SiteWise portail.
- Créez une AWS IoT SiteWise passerelle.
- Décrivez le AWS IoT SiteWise Gateway.
- Supprimez les AWS IoT SiteWise actifs.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Exécutez un scénario interactif illustrant AWS IoT SiteWise les fonctionnalités.

```
public class SitewiseScenario {  
  
    public static final String DASHES = new String(new char[80]).replace("\0",  
        "-");  
}
```

```

    private static final Logger logger =
LoggerFactory.getLogger(SitewiseScenario.class);
    static Scanner scanner = new Scanner(System.in);

    private static final String ROLES_STACK = "RoleSitewise";

    static SitewiseActions sitewiseActions = new SitewiseActions();

    public static void main(String[] args) throws Throwable {
        Scanner scanner = new Scanner(System.in);
        String contactEmail = "user@mydomain.com"; // Change email address.
        String assetModelName = "MyAssetModel1";
        String assetName = "MyAsset1" ;
        String portalName = "MyPortal1" ;
        String gatewayName = "MyGateway1" ;
        String myThing = "MyThing1" ;

        logger.info("""
            AWS IoT SiteWise is a fully managed software-as-a-service (SaaS)
that
            makes it easy to collect, store, organize, and monitor data from
industrial equipment and processes.
            It is designed to help industrial and manufacturing organizations
collect data from their equipment and
            processes, and use that data to make informed decisions about their
operations.

            One of the key features of AWS IoT SiteWise is its ability to connect
to a wide range of industrial
            equipment and systems, including programmable logic controllers
(PLCs), sensors, and other
            industrial devices. It can collect data from these devices and
organize it into a unified data model,
            making it easier to analyze and gain insights from the data. AWS IoT
SiteWise also provides tools for
            visualizing the data, setting up alarms and alerts, and generating
reports.

            Another key feature of AWS IoT SiteWise is its ability to scale to
handle large volumes of data.
            It can collect and store data from thousands of devices and process
millions of data points per second,

```

making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

Let's get started...

```
""");
```

```
waitForInputToContinue(scanner);
```

```
logger.info(DASHES);
```

```
try {
    runScenario(assetModelName, assetName, portalName, contactEmail,
gatewayName, myThing);
} catch (RuntimeException e) {
    logger.info(e.getMessage());
}
}
```

```
public static void runScenario(String assetModelName, String assetName,
String portalName, String contactEmail, String gatewayName, String myThing)
throws Throwable {
```

```
    logger.info("Use AWS CloudFormation to create an IAM role that is
required for this scenario.");
```

```
    CloudFormationHelper.deployCloudFormationStack(ROLES_STACK);
```

```
    Map<String, String> stackOutputs =
```

```
CloudFormationHelper.getStackOutputsAsync(ROLES_STACK).join();
```

```
    String iamRole = stackOutputs.get("SitewiseRoleArn");
```

```
    logger.info("The ARN of the IAM role is {}", iamRole);
```

```
    logger.info(DASHES);
```

```
    logger.info(DASHES);
```

```
    logger.info("1. Create an AWS SiteWise Asset Model");
```

```
    logger.info("""
```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment,

processes, and systems, that exist in an industrial environment.

This model provides a structured and

hierarchical representation of these assets, allowing users to define the relationships and properties

of each asset.


```

        This scenario creates two asset model properties: temperature and
        humidity.

        """);
        waitForInputToContinue(scanner);
        String assetModelId = null;
        try {
            CreateAssetModelResponse response =
sitewiseActions.createAssetModelAsync(assetModelName).join();
            assetModelId = response.assetModelId();
            logger.info("Asset Model successfully created. Asset Model ID: {}. ",
assetModelId);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceAlreadyExistsException) {
                try {
                    assetModelId =
sitewiseActions.getAssetModelIdAsync(assetModelName).join();
                    logger.info("The Asset Model {} already exists. The id of the
existing model is {}. Moving on...", assetModelName, assetModelId);
                } catch (CompletionException cex) {
                    logger.error("Exception thrown acquiring the asset model id:
{}", cex.getCause().getCause(), cex);
                    return;
                }
            } else {
                logger.info("An unexpected error occurred: " +
cause.getMessage(), cause);
                return;
            }
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("2. Create an AWS IoT SiteWise Asset");
        logger.info("""
            The IoT SiteWise model that we just created defines the structure
            and metadata for your physical assets.
            Now we create an asset from the asset model.

            """);
        logger.info("Let's wait 30 seconds for the asset to be ready.");
        countdown(30);
        waitForInputToContinue(scanner);
        String assetId;

```

```

    try {
        CreateAssetResponse response =
sitewiseActions.createAssetAsync(assetName, assetModelId).join();
        assetId = response.assetId();
        logger.info("Asset created with ID: {}", assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException) {
            logger.info("The asset model id was not found: {}",
cause.getMessage(), cause);
        } else {
            logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
        }
        return;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("3. Retrieve the property ID values");
    logger.info(""""
        To send data to an asset, we need to get the property ID values. In
this scenario, we access the
        temperature and humidity property ID values.
        """);
    waitForInputToContinue(scanner);
    Map<String, String> propertyIds = null;
    try {
        propertyIds = sitewiseActions.getPropertyIds(assetModelId).join();
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof IoTSiteWiseException) {
            logger.error("IoTSiteWiseException occurred: {}",
cause.getMessage(), ce);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage(), ce);
        }
        return;
    }
    String humPropId = propertyIds.get("Humidity");
    logger.info("The Humidity property Id is {}", humPropId);
    String tempPropId = propertyIds.get("Temperature");

```

```

logger.info("The Temperature property Id is {}", tempPropId);

waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("4. Send data to an AWS IoT SiteWise Asset");
logger.info("
    By sending data to an IoT SiteWise Asset, you can aggregate data
from
    multiple sources, normalize the data into a standard format, and
store it in a
    centralized location. This makes it easier to analyze and gain
insights from the data.

    In this example, we generate sample temperature and humidity data and
send it to the AWS IoT SiteWise asset.

    ");
waitForInputToContinue(scanner);
try {
    sitewiseActions.sendDataToSiteWiseAsync(assetId, tempPropId,
humPropId).join();
    logger.info("Data sent successfully.");
} catch (CompletionException ce) {
    Throwable cause = ce.getCause();
    if (cause instanceof ResourceNotFoundException) {
        logger.error("The AWS resource was not found: {}",
cause.getMessage(), cause);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
    }
    return;
}
waitForInputToContinue(scanner);
logger.info(DASHES);

logger.info(DASHES);
logger.info("5. Retrieve the value of the IoT SiteWise Asset property");
logger.info("
    IoT SiteWise is an AWS service that allows you to collect, process,
and analyze industrial data

```

```

        from connected equipment and sensors. One of the key benefits of
        reading an IoT SiteWise property
        is the ability to gain valuable insights from your industrial data.

        """);
        waitForInputToContinue(scanner);
        try {
            Double assetVal = sitewiseActions.getAssetPropValueAsync(tempPropId,
assetId).join();
            logger.info("The property name is: {}", "Temperature");
            logger.info("The value of this property is: {}", assetVal);

            waitForInputToContinue(scanner);

            assetVal = sitewiseActions.getAssetPropValueAsync(humPropId,
assetId).join();
            logger.info("The property name is: {}", "Humidity");
            logger.info("The value of this property is: {}", assetVal);
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException) {
                logger.info("The AWS resource was not found: {}",
cause.getMessage(), cause);
            } else {
                logger.info("An unexpected error occurred: {}",
cause.getMessage(), cause);
            }
            return;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);

        logger.info(DASHES);
        logger.info("6. Create an IoT SiteWise Portal");
        logger.info("""
            An IoT SiteWise Portal allows you to aggregate data from multiple
            industrial sources,
            such as sensors, equipment, and control systems, into a centralized
            platform.

            """);
        waitForInputToContinue(scanner);
        String portalId;
        try {

```

```

        portalId = sitewiseActions.createPortalAsync(portalName, iamRole,
contactEmail).join();
        logger.info("Portal created successfully. Portal ID {}", portalId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof IoTSiteWiseException siteWiseEx) {
            logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
                        siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
        return;
    }
    waitForInputToContinue(scanner);
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("7. Describe the Portal");
    logger.info(""""
        In this step, we get a description of the portal and display the
portal URL.
        """);
    waitForInputToContinue(scanner);
    try {
        String portalUrl =
sitewiseActions.describePortalAsync(portalId).join();
        logger.info("Portal URL: {}", portalUrl);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException) {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
        return;
    }
    waitForInputToContinue(scanner);

```

```

    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("8. Create an IoT SiteWise Gateway");
    logger.info(
        ""
        IoT SiteWise Gateway serves as the bridge between industrial
equipment, sensors, and the
        cloud-based IoT SiteWise service. It is responsible for securely
collecting, processing, and
        transmitting data from various industrial assets to the IoT
SiteWise platform,
        enabling real-time monitoring, analysis, and optimization of
industrial operations.

        "");
    waitForInputToContinue(scanner);
    String gatewayId = "";
    try {
        gatewayId = sitewiseActions.createGatewayAsync(gatewayName,
myThing).join();
        logger.info("Gateway creation completed successfully. id is {}",
gatewayId );
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof IoTSiteWiseException siteWiseEx) {
            logger.error("IoT SiteWise error occurred: Error message: {},
Error code {}",
                        siteWiseEx.getMessage(),
siteWiseEx.awsErrorDetails().errorCode(), siteWiseEx);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
        return;
    }
    logger.info(DASHES);
    logger.info(DASHES);

    logger.info("9. Describe the IoT SiteWise Gateway");
    waitForInputToContinue(scanner);
    try {
        sitewiseActions.describeGatewayAsync(gatewayId)
            .thenAccept(response -> {

```

```

        logger.info("Gateway Name: {}", response.gatewayName());
        logger.info("Gateway ARN: {}", response.gatewayArn());
        logger.info("Gateway Platform: {}",
response.gatewayPlatform());
        logger.info("Gateway Creation Date: {}",
response.creationDate());
    }).join();
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException) {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage(), cause);
        }
        return;
    }
    logger.info(DASHES);

    logger.info(DASHES);
    logger.info("10. Delete the AWS IoT SiteWise Assets");
    logger.info(
        ""
        Before you can delete the Asset Model, you must delete the assets.

        "");
    logger.info("Would you like to delete the IoT SiteWise Assets? (y/n)");
    String delAns = scanner.nextLine().trim();
    if (delAns.equalsIgnoreCase("y")) {
        logger.info("You selected to delete the SiteWise assets.");
        waitForInputToContinue(scanner);
        try {
            sitewiseActions.deletePortalAsync(portalId).join();
            logger.info("Portal {} was deleted successfully.", portalId);

        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException)
{
                logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",

```

```

        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
    } else {
        logger.error("An unexpected error occurred: {}",
cause.getMessage());
    }
}

    try {
        sitewiseActions.deleteGatewayAsync(gatewayId).join();
        logger.info("Gateway {} was deleted successfully.", gatewayId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
{
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }

    try {
        sitewiseActions.deleteAssetAsync(assetId).join();
        logger.info("Request to delete asset {} sent successfully",
assetId);
    } catch (CompletionException ce) {
        Throwable cause = ce.getCause();
        if (cause instanceof ResourceNotFoundException notFoundException)
{
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
        } else {
            logger.error("An unexpected error occurred: {}",
cause.getMessage());
        }
    }
    logger.info("Let's wait 1 minute for the asset to be deleted.");
    countdown(60);

```



```

        waitForInputToContinue(scanner);
        logger.info("Delete the AWS IoT SiteWise Asset Model");
        try {
            sitewiseActions.deleteAssetModelAsync(assetModelId).join();
            logger.info("Asset model deleted successfully.");
        } catch (CompletionException ce) {
            Throwable cause = ce.getCause();
            if (cause instanceof ResourceNotFoundException notFoundException)
        {
            logger.error("A ResourceNotFoundException occurred: Error
message: {}, Error code {}",
                        notFoundException.getMessage(),
notFoundException.awsErrorDetails().errorCode(), notFoundException);
            } else {
                logger.error("An unexpected error occurred: {}",
cause.getMessage());
            }
        }
        waitForInputToContinue(scanner);

    } else {
        logger.info("The resources will not be deleted.");
    }
    logger.info(DASHES);

    logger.info(DASHES);
    CloudFormationHelper.destroyCloudFormationStack(ROLES_STACK);
    logger.info("This concludes the AWS IoT SiteWise Scenario");
    logger.info(DASHES);
}

private static void waitForInputToContinue(Scanner scanner) {
    while (true) {
        logger.info("");
        logger.info("Enter 'c' followed by <ENTER> to continue:");
        String input = scanner.nextLine();

        if (input.trim().equalsIgnoreCase("c")) {
            logger.info("Continuing with the program...");
            logger.info("");
            break;
        } else {
            logger.info("Invalid input. Please try again.");
        }
    }
}

```

```

    }
}

public static void countdown(int totalSeconds) throws InterruptedException {
    for (int i = totalSeconds; i >= 0; i--) {
        int displayMinutes = i / 60;
        int displaySeconds = i % 60;
        System.out.printf("\r%02d:%02d", displayMinutes, displaySeconds);
        Thread.sleep(1000); // Wait for 1 second
    }
    System.out.println(); // Move to the next line after countdown
    logger.info("Countdown complete!");
}
}

```

Une classe wrapper pour les méthodes du AWS IoT SiteWise SDK.

```

public class SitewiseActions {

    private static final Logger logger =
        LoggerFactory.getLogger(SitewiseActions.class);

    private static IoTSiteWiseAsyncClient ioTSiteWiseAsyncClient;

    private static IoTSiteWiseAsyncClient getAsyncClient() {
        if (ioTSiteWiseAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
                ClientOverrideConfiguration.builder()
                    .apiCallTimeout(Duration.ofMinutes(2))
                    .apiCallAttemptTimeout(Duration.ofSeconds(90))
                    .retryStrategy(RetryMode.STANDARD)
                    .build();

            ioTSiteWiseAsyncClient = IoTSiteWiseAsyncClient.builder()
                .httpClient(httpClient)

```

```

        .overrideConfiguration(overrideConfig)
        .build();
    }
    return ioTSiteWiseAsyncClient;
}

/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *
 * <p>
 *
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
    PropertyType humidity = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    PropertyType temperaturePropertyType = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
        .name("Temperature")
        .dataType(PropertyDataType.DOUBLE)
        .type(temperaturePropertyType)
        .build();

    AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
        .name("Humidity")

```

```

        .dataType(PropertyDataType.DOUBLE)
        .type(humidity)
        .build();

    CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
        .assetModelName(name)
        .assetModelDescription("This is my asset model")
        .assetModelProperties(temperatureProperty, humidityProperty)
        .build();

    return getAsyncClient().createAssetModel(createAssetModelRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
            }
        });
}

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *      {@link CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *      available to the calling code as a {@link CompletionException}. By
calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")

```

```

        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}

/**
 * Sends data to the SiteWise service.
 *
 * @param assetId the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 * calling code can attach callbacks, then handle the result or
exception by calling
 * {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 * available to the calling code as a {@link CompletionException}. By
calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();

```

```

        BatchPutAssetPropertyValueRequest request =
BatchPutAssetPropertyValueRequest.builder()
    .entries(Arrays.asList(
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-3")
            .assetId(assetId)
            .propertyId(tempPropertyId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Temperature"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build(),
        PutAssetPropertyValueEntry.builder()
            .entryId("entry-4")
            .assetId(assetId)
            .propertyId(humidityPropId)
            .propertyValues(Arrays.asList(
                AssetPropertyValue.builder()
                    .value(Variant.builder()
                        .doubleValue(sampleData.get("Humidity"))
                        .build())
                    .timestamp(time)
                    .build()
            ))
            .build()
    ))
    .build();

return getAsyncClient().batchPutAssetPropertyValue(request)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("An exception occurred: {}",
exception.getCause().getMessage());
        }
    });
}

/**
 * Fetches the value of an asset property.
 *

```

```

    * @param propId the ID of the asset property to fetch.
    * @param assetId the ID of the asset to fetch the property value for.
    * @return a {@link CompletableFuture} that represents a {@link Double}
result. The calling code can attach
    *     callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
    *     {@link CompletableFuture#get()}.
    *
    * <p>
    *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *     it available to the calling code as a {@link CompletionException}.
By calling
    *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
        GetAssetPropertyValueRequest assetPropertyValueRequest =
GetAssetPropertyValueRequest.builder()
            .propertyId(propId)
            .assetId(assetId)
            .build();

        return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.propertyValue().value().doubleValue();
            });
    }

/**
    * Retrieves the property IDs associated with a specific asset model.
    *
    * @param assetModelId the ID of the asset model that defines the properties.
    * @return a {@link CompletableFuture} that represents a {@link Map} result
that associates the property name to the
    *     propert ID. The calling code can attach callbacks, then handle the
result or exception by calling
    *     {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.

```

```

    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
        ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
        return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
            .handle((response, throwable) -> {
                if (response != null) {
                    return response.assetModelPropertySummaries().stream()
                        .collect(Collectors
                            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
                } else {
                    logger.error("Error occurred while fetching property IDs:
{}.\"", throwable.getCause().getMessage());
                    throw (CompletionException) throwable;
                }
            });
    }

    /**
    * Deletes an asset.
    *
    * @param assetId the ID of the asset to be deleted.
    * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
    *      attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
By calling
    *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */

```



```

    public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
        DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
            .assetId(assetId)
            .build();

        return getAsyncClient().deleteAsset(deleteAssetRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred deleting asset with id: {}",
assetId);
                }
            });
    }

    /**
     * Deletes an Asset Model with the specified ID.
     *
     * @param assetModelId the ID of the Asset Model to delete.
     * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
     *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
     *         {@link CompletableFuture#get()}.
     *         <p>
     *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
     *         it available to the calling code as a {@link CompletionException}.
By calling
     *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
     */
    public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
        DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
            .assetModelId(assetModelId)
            .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());

```

```

        }
    });
}

/**
 * Creates a new IoT SiteWise portal.
 *
 * @param portalName the name of the portal to create.
 * @param iamRole the IAM role ARN to use for the portal.
 * @param contactEmail the email address of the portal contact.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal ID. The calling code
 * can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
    CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
        .portalName(portalName)
        .portalDescription("This is my custom IoT SiteWise portal.")
        .portalContactEmail(contactEmail)
        .roleArn(iamRole)
        .build();

    return getAsyncClient().createPortal(createPortalRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalId();
        });
}

/**

```

```

    * Deletes a portal.
    *
    * @param portalId the ID of the portal to be deleted.
    * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. The calling code can attach
    *     callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
    *     {@link CompletableFuture#get()}.
    *     <p>
    *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
    *     it available to the calling code as a {@link CompletionException}.
By calling
    *     {@link CompletionException#getCause()}, the calling code can
access the original exception.
    */
    public CompletableFuture<DeletePortalResponse> deletePortalAsync(String
portalId) {
        DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
            .portalId(portalId)
            .build();

        return getAsyncClient().deletePortal(deletePortalRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete portal with ID: {}. Error:
{}", portalId, exception.getCause().getMessage());
                }
            });
    }

    /**
    * Retrieves the asset model ID for the given asset model name.
    *
    * @param assetModelName the name of the asset model for the ID.
    * @return a {@link CompletableFuture} that represents a {@link String}
result of the asset model ID or null if the
    *     asset model cannot be found. The calling code can attach
callbacks, then handle the result or exception
    *     by calling {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
    *     <p>
    *     If any completion stage in this method throws an exception, the
method logs the exception cause and keeps

```

```

        *           it available to the calling code as a {@link CompletionException}.
By calling
        *           {@link CompletionException#getCause()}, the calling code can
access the original exception.
        */
    public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
    {
        ListAssetModelsRequest listAssetModelsRequest =
ListAssetModelsRequest.builder().build();
        return getAsyncClient().listAssetModels(listAssetModelsRequest)
            .handle((listAssetModelsResponse, exception) -> {
                if (exception != null) {
                    logger.error("Failed to retrieve Asset Model ID: {}",
exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                for (AssetModelSummary assetModelSummary :
listAssetModelsResponse.assetModelSummaries()) {
                    if (assetModelSummary.name().equals(assetModelName)) {
                        return assetModelSummary.id();
                    }
                }
                return null;
            });
    }

    /**
     * Retrieves a portal's description.
     *
     * @param portalId the ID of the portal to describe.
     * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal's start URL
     * (see: {@link DescribePortalResponse#portalStartUrl()}). The
calling code can attach callbacks, then handle the
     * result or exception by calling {@link CompletableFuture#join()} or
{@link CompletableFuture#get()}.
     * <p>
     * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
     * it available to the calling code as a {@link CompletionException}.
By calling
     * {@link CompletionException#getCause()}, the calling code can
access the original exception.
     */

```

```

public CompletableFuture<String> describePortalAsync(String portalId) {
    DescribePortalRequest request = DescribePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().describePortal(request)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
            return response.portalStartUrl();
        });
}

/**
 * Creates a new IoT SiteWise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
    GreengrassV2 gg = GreengrassV2.builder()
        .coreDeviceThingName(myThing)
        .build();

    GatewayPlatform platform = GatewayPlatform.builder()
        .greengrassV2(gg)

```

```

        .build();

        Map<String, String> tag = new HashMap<>();
        tag.put("Environment", "Production");

        CreateGatewayRequest createGatewayRequest =
        CreateGatewayRequest.builder()
            .gatewayName(gatewayName)
            .gatewayPlatform(platform)
            .tags(tag)
            .build();

        return getAsyncClient().createGateway(createGatewayRequest)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("Error creating the gateway.");
                    throw (CompletionException) exception;
                }
                logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
                return response.gatewayId();
            });
    }

    /**
     * Deletes the specified gateway.
     *
     * @param gatewayId the ID of the gateway to delete.
     * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
     *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
     *         {@link CompletableFuture#get()}.
     *         <p>
     *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
     *         it available to the calling code as a {@link CompletionException}.
By calling
     *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
     */
    public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {

```

```

        DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

        return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
    }

    /**
     * Describes the specified gateway.
     *
     * @param gatewayId the ID of the gateway to describe.
     * @return a {@link CompletableFuture} that represents a {@link
DescribeGatewayResponse} result. The calling code
     *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
     *         {@link CompletableFuture#get()}.
     *
     * <p>
     *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
     *         it available to the calling code as a {@link CompletionException}.
By calling
     *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
     */
    public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
        DescribeGatewayRequest request = DescribeGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

        return getAsyncClient().describeGateway(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
            }
        });
    }

```

```
}

private static Map<String, Double> generateSampleData() {
    Map<String, Double> data = new HashMap<>();
    data.put("Temperature", 23.5);
    data.put("Humidity", 65.0);
    return data;
}
}
```

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
    Scenario,
    ScenarioAction,
    ScenarioInput,
    ScenarioOutput,
    //} from "@aws-doc-sdk-examples/lib/scenario/index.js";
} from "../../libs/scenario/index.js";
import {
    IoTSiteWiseClient,
    CreateAssetModelCommand,
    CreateAssetCommand,
    ListAssetModelPropertiesCommand,
    BatchPutAssetPropertyValueCommand,
    GetAssetPropertyValueCommand,
    CreatePortalCommand,
    DescribePortalCommand,
    CreateGatewayCommand,
    DescribeGatewayCommand,
    DeletePortalCommand,
    DeleteGatewayCommand,
```



```

    DeleteAssetCommand,
    DeleteAssetModelCommand,
    DescribeAssetModelCommand,
} from "@aws-sdk/client-iotsitewise";
import {
    CloudFormationClient,
    CreateStackCommand,
    DeleteStackCommand,
    DescribeStacksCommand,
    waitUntilStackExists,
    waitUntilStackCreateComplete,
    waitUntilStackDeleteComplete,
} from "@aws-sdk/client-cloudformation";
import { wait } from "@aws-doc-sdk-examples/lib/utils/util-timers.js";
import { parseArgs } from "node:util";
import { readFileSync } from "node:fs";
import { fileURLToPath } from "node:url";
import { dirname } from "node:path";

const __filename = fileURLToPath(import.meta.url);
const __dirname = dirname(__filename);
const stackName = "SiteWiseBasicsStack";

/**
 * @typedef {{
 *   iotSiteWiseClient: import('@aws-sdk/client-iotsitewise').IotSiteWiseClient,
 *   cloudFormationClient: import('@aws-sdk/client-
cloudformation').CloudFormationClient,
 *   stackName,
 *   stack,
 *   askToDeleteResources: true,
 *   asset: {assetName: "MyAsset1"},
 *   assetModel: {assetModelName: "MyAssetModel1"},
 *   portal: {portalName: "MyPortal1"},
 *   gateway: {gatewayName: "MyGateway1"},
 *   propertyIds: [],
 *   contactEmail: "user@mydomain.com",
 *   thing: "MyThing1",
 *   sampleData: { temperature: 23.5, humidity: 65.0}
 * }} State
 */

/**
 * Used repeatedly to have the user press enter.

```

```

* @type {ScenarioInput}
*/
const pressEnter = new ScenarioInput("continue", "Press Enter to continue", {
  type: "confirm",
});

const greet = new ScenarioOutput(
  "greet",
  `AWS IoT SiteWise is a fully managed industrial software-as-a-service (SaaS)
  that makes it easy to collect, store, organize, and monitor data from industrial
  equipment and processes. It is designed to help industrial and manufacturing
  organizations collect data from their equipment and processes, and use that data
  to make informed decisions about their operations.
  One of the key features of AWS IoT SiteWise is its ability to connect to a
  wide range of industrial equipment and systems, including programmable logic
  controllers (PLCs), sensors, and other industrial devices. It can collect data
  from these devices and organize it into a unified data model, making it easier
  to analyze and gain insights from the data. AWS IoT SiteWise also provides tools
  for visualizing the data, setting up alarms and alerts, and generating reports.
  Another key feature of AWS IoT SiteWise is its ability to scale to handle large
  volumes of data. It can collect and store data from thousands of devices and
  process millions of data points per second, making it suitable for large-scale
  industrial operations. Additionally, AWS IoT SiteWise is designed to be secure
  and compliant, with features like role-based access controls, data encryption,
  and integration with other AWS services for additional security and compliance
  features.

  Let's get started...`,
  { header: true },
);

const displayBuildCloudFormationStack = new ScenarioOutput(
  "displayBuildCloudFormationStack",
  "This scenario uses AWS CloudFormation to create an IAM role that is required
  for this scenario. The stack will now be deployed.",
);

const sdkBuildCloudFormationStack = new ScenarioAction(
  "sdkBuildCloudFormationStack",
  async (** @type {State} */ state) => {
    try {
      const data = readFileSync(
        `${__dirname}/../../../../resources/cfn/iotsitewise_basics/SitewiseRoles-
        template.yml`,

```

```

    "utf8",
  );
  await state.cloudFormationClient.send(
    new CreateStackCommand({
      StackName: stackName,
      TemplateBody: data,
      Capabilities: ["CAPABILITY_IAM"],
    }),
  );
  await waitUntilStackExists(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  await waitUntilStackCreateComplete(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  const stack = await state.cloudFormationClient.send(
    new DescribeStacksCommand({
      StackName: stackName,
    }),
  );
  state.stack = stack.Stacks[0].Outputs[0];
  console.log(`The ARN of the IAM role is ${state.stack.OutputValue}`);
} catch (caught) {
  console.error(caught.message);
  throw caught;
}
},
);

```

```

const displayCreateAWSSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSSiteWiseAssetModel",
  `1. Create an AWS SiteWise Asset Model

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and properties of each asset.

This scenario creates two asset model properties: temperature and humidity.`,

```

);

```

```

const sdkCreateAWSSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSSiteWiseAssetModel",

```

```

async (/** @type {State} */ state) => {
  let assetModelResponse;
  try {
    assetModelResponse = await state.iotSiteWiseClient.send(
      new CreateAssetModelCommand({
        assetModelName: state.assetModel.assetModelName,
        assetModelProperties: [
          {
            name: "Temperature",
            dataType: "DOUBLE",
            type: {
              measurement: {},
            },
          },
          {
            name: "Humidity",
            dataType: "DOUBLE",
            type: {
              measurement: {},
            },
          },
        ],
      })),
    );
    state.assetModel.assetModelId = assetModelResponse.assetModelId;
    console.log(
      `Asset Model successfully created. Asset Model ID:
    ${state.assetModel.assetModelId}`,
    );
  } catch (caught) {
    if (caught.name === "ResourceAlreadyExistsException") {
      console.log(
        `The Asset Model ${state.assetModel.assetModelName} already exists.`,
      );
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displayCreateAWSIoTSiteWiseAssetModel = new ScenarioOutput(
  "displayCreateAWSIoTSiteWiseAssetModel",

```

`2. Create an AWS IoT SiteWise Asset

The IoT SiteWise model that we just created defines the structure and metadata for your physical assets. Now we create an asset from the asset model.

Let's wait 30 seconds for the asset to be ready.`,
);

```
const waitThirtySeconds = new ScenarioAction("waitThirtySeconds", async () => {
  await wait(30); // wait 30 seconds
  console.log("Time's up! Let's check the asset's status.");
});
```

```
const sdkCreateAWSIoTSiteWiseAssetModel = new ScenarioAction(
  "sdkCreateAWSIoTSiteWiseAssetModel",
  async (** @type {State} */ state) => {
    try {
      const assetResponse = await state.iotSiteWiseClient.send(
        new CreateAssetCommand({
          assetModelId: state.assetModel.assetModelId,
          assetName: state.asset.assetName,
        })),
      );
      state.asset.assetId = assetResponse.assetId;
      console.log(`Asset created with ID: ${state.asset.assetId}`);
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(
          `The Asset ${state.assetModel.assetModelName} was not found.`
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);
```

```
const displayRetrievePropertyId = new ScenarioOutput(
  "displayRetrievePropertyId",
  `3. Retrieve the property ID values
```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.`,
);

```
const sdkRetrievePropertyId = new ScenarioAction(
  "sdkRetrievePropertyId",
  async (state) => {
    try {
      const retrieveResponse = await state.iotSiteWiseClient.send(
        new ListAssetModelPropertiesCommand({
          assetModelId: state.assetModel.assetModelId,
        }),
      );
      for (const retrieveResponseKey in
retrieveResponse.assetModelPropertySummaries) {
        if (
          retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
            .name === "Humidity"
        ) {
          state.propertyIds.Humidity =
            retrieveResponse.assetModelPropertySummaries[
              retrieveResponseKey
            ].id;
        }
        if (
          retrieveResponse.assetModelPropertySummaries[retrieveResponseKey]
            .name === "Temperature"
        ) {
          state.propertyIds.Temperature =
            retrieveResponse.assetModelPropertySummaries[
              retrieveResponseKey
            ].id;
        }
      }
      console.log(`The Humidity propertyId is ${state.propertyIds.Humidity}`);
      console.log(
        `The Temperature propertyId is ${state.propertyIds.Temperature}`,
      );
    } catch (caught) {
      if (caught.name === "IoTSiteWiseException") {
        console.log(
          `There was a problem retrieving the properties: ${caught.message}`,
        );
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  }
);
```

```

    }
  },
);

const displaySendDataToIoTSiteWiseAsset = new ScenarioOutput(
  "displaySendDataToIoTSiteWiseAsset",
  `4. Send data to an AWS IoT SiteWise Asset

```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.`,

```

);

```

```

const sdkSendDataToIoTSiteWiseAsset = new ScenarioAction(
  "sdkSendDataToIoTSiteWiseAsset",
  async (state) => {
    try {
      const sendResponse = await state.iotSiteWiseClient.send(
        new BatchPutAssetPropertyValueCommand({
          entries: [
            {
              entryId: "entry-3",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Humidity,
              propertyValues: [
                {
                  value: {
                    doubleValue: state.sampleData.humidity,
                  },
                  timestamp: {
                    timeInSeconds: Math.floor(Date.now() / 1000),
                  },
                },
              ],
            },
            {
              entryId: "entry-4",
              assetId: state.asset.assetId,
              propertyId: state.propertyIds.Temperature,
              propertyValues: [

```

```

        {
            value: {
                doubleValue: state.sampleData.temperature,
            },
            timestamp: {
                timeInSeconds: Math.floor(Date.now() / 1000),
            },
        },
    ],
},
]),
}),
);
console.log("The data was sent successfully.");
} catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
        console.log(`The Asset ${state.asset.assetName} was not found.`);
        throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
}
},
);

const displayRetrieveValueOfIoTSiteWiseAsset = new ScenarioOutput(
    "displayRetrieveValueOfIoTSiteWiseAsset",
    `5. Retrieve the value of the IoT SiteWise Asset property

IoT SiteWise is an AWS service that allows you to collect, process, and analyze
industrial data from connected equipment and sensors. One of the key benefits of
reading an IoT SiteWise property is the ability to gain valuable insights from
your industrial data.`,
);

const sdkRetrieveValueOfIoTSiteWiseAsset = new ScenarioAction(
    "sdkRetrieveValueOfIoTSiteWiseAsset",
    async (/** @type {State} */ state) => {
        try {
            const temperatureResponse = await state.iotSiteWiseClient.send(
                new GetAssetPropertyValueCommand({
                    assetId: state.asset.assetId,
                    propertyId: state.propertyIds.Temperature,
                }),
            );

```



```

    );
    const humidityResponse = await state.iotSiteWiseClient.send(
      new GetAssetPropertyValueCommand({
        assetId: state.asset.assetId,
        propertyId: state.propertyIds.Humidity,
      }),
    );
    console.log(
      `The property value for Temperature is
    ${temperatureResponse.propertyValue.value.doubleValue}`,
    );
    console.log(
      `The property value for Humidity is
    ${humidityResponse.propertyValue.value.doubleValue}`,
    );
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
      throw caught;
    }
    console.error(`${caught.message}`);
    throw caught;
  }
},
);

const displayCreateIoTSiteWisePortal = new ScenarioOutput(
  "displayCreateIoTSiteWisePortal",
  `6. Create an IoT SiteWise Portal

An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources, such as sensors, equipment, and control systems, into a centralized
platform.`
);

const sdkCreateIoTSiteWisePortal = new ScenarioAction(
  "sdkCreateIoTSiteWisePortal",
  async (** @type {State} */ state) => {
    try {
      const createPortalResponse = await state.iotSiteWiseClient.send(
        new CreatePortalCommand({
          portalName: state.portal.portalName,
          portalContactEmail: state.contactEmail,
          roleArn: state.stack.OutputValue,

```

```

    })),
  );
  state.portal = { ...state.portal, ...createPortalResponse };
  await wait(5); // Allow the portal to properly propagate.
  console.log(
    `Portal created successfully. Portal ID
    ${createPortalResponse.portalId}`,
  );
} catch (caught) {
  if (caught.name === "IoTSiteWiseException") {
    console.log(
      `There was a problem creating the Portal: ${caught.message}.`,
    );
    throw caught;
  }
  console.error(`${caught.message}`);
  throw caught;
}
},
);

```

```

const displayDescribePortal = new ScenarioOutput(
  "displayDescribePortal",
  `7. Describe the Portal

```

In this step, we get a description of the portal and display the portal URL.`,

```

);

```

```

const sdkDescribePortal = new ScenarioAction(
  "sdkDescribePortal",
  async (/** @type {State} */ state) => {
    try {
      const describePortalResponse = await state.iotSiteWiseClient.send(
        new DescribePortalCommand({
          portalId: state.portal.portalId,
        }),
      );
      console.log(`Portal URL: ${describePortalResponse.portalStartUrl}`);
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Portal ${state.portal.portalName} was not found.`);
        throw caught;
      }
      console.error(`${caught.message}`);
    }
  }
);

```

```

        throw caught;
    }
},
);

```

```

const displayCreateIoTSiteWiseGateway = new ScenarioOutput(
    "displayCreateIoTSiteWiseGateway",
    `8. Create an IoT SiteWise Gateway

```

IoT SiteWise Gateway serves as the bridge between industrial equipment, sensors, and the cloud-based IoT SiteWise service. It is responsible for securely collecting, processing, and transmitting data from various industrial assets to the IoT SiteWise platform, enabling real-time monitoring, analysis, and optimization of industrial operations.`,

```

);

```

```

const sdkCreateIoTSiteWiseGateway = new ScenarioAction(
    "sdkCreateIoTSiteWiseGateway",
    async (/** @type {State} */ state) => {
        try {
            const createGatewayResponse = await state.iotSiteWiseClient.send(
                new CreateGatewayCommand({
                    gatewayName: state.gateway.gatewayName,
                    gatewayPlatform: {
                        greengrassV2: {
                            coreDeviceThingName: state.thing,
                        },
                    },
                }),
            );
            console.log(
                `Gateway creation completed successfully. ID is
                ${createGatewayResponse.gatewayId}`,
            );
            state.gateway.gatewayId = createGatewayResponse.gatewayId;
        } catch (caught) {
            if (caught.name === "IoTSiteWiseException") {
                console.log(
                    `There was a problem creating the gateway: ${caught.message}.`,
                );
                throw caught;
            }
            console.error(`${caught.message}`);
            throw caught;
        }
    }
);

```

```

    }
  },
);

const displayDescribeIoTSiteWiseGateway = new ScenarioOutput(
  "displayDescribeIoTSiteWiseGateway",
  "9. Describe the IoT SiteWise Gateway",
);

const sdkDescribeIoTSiteWiseGateway = new ScenarioAction(
  "sdkDescribeIoTSiteWiseGateway",
  async (** @type {State} */ state) => {
    try {
      const describeGatewayResponse = await state.iotSiteWiseClient.send(
        new DescribeGatewayCommand({
          gatewayId: state.gateway.gatewayId,
        }),
      );
      console.log("Gateway creation completed successfully.");
      console.log(`Gateway Name: ${describeGatewayResponse.gatewayName}`);
      console.log(`Gateway ARN: ${describeGatewayResponse.gatewayArn}`);
      console.log(
        `Gateway Platform:
${Object.keys(describeGatewayResponse.gatewayPlatform)}`,
      );
      console.log(
        `Gateway Creation Date: ${describeGatewayResponse.creationDate}`,
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
        throw caught;
      }
      console.error(`${caught.message}`);
      throw caught;
    }
  },
);

const askToDeleteResources = new ScenarioInput(
  "askToDeleteResources",
  `10. Delete the AWS IoT SiteWise Assets

Before you can delete the Asset Model, you must delete the assets.`

```

```
    { type: "confirm" },
  );

const displayConfirmDeleteResources = new ScenarioAction(
  "displayConfirmDeleteResources",
  async (** @type {State} */ state) => {
    if (state.askToDeleteResources) {
      return "You selected to delete the SiteWise assets.";
    }
    return "The resources will not be deleted. Please delete them manually to avoid charges.";
  },
);

const sdkDeleteResources = new ScenarioAction(
  "sdkDeleteResources",
  async (** @type {State} */ state) => {
    await wait(10); // Give the portal status time to catch up.
    try {
      await state.iotSiteWiseClient.send(
        new DeletePortalCommand({
          portalId: state.portal.portalId,
        }),
      );
      console.log(
        `Portal ${state.portal.portalName} was deleted successfully.`,
      );
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Portal ${state.portal.portalName} was not found.`);
      } else {
        console.log(`When trying to delete the portal: ${caught.message}`);
      }
    }

    try {
      await state.iotSiteWiseClient.send(
        new DeleteGatewayCommand({
          gatewayId: state.gateway.gatewayId,
        }),
      );
      console.log(
        `Gateway ${state.gateway.gatewayName} was deleted successfully.`,
      );
    }
  }
);
```

```
    } catch (caught) {
      if (caught.name === "ResourceNotFoundException") {
        console.log(`The Gateway ${state.gateway.gatewayId} was not found.`);
      } else {
        console.log(`When trying to delete the gateway: ${caught.message}`);
      }
    }
  }

  try {
    await state.iotSiteWiseClient.send(
      new DeleteAssetCommand({
        assetId: state.asset.assetId,
      }),
    );
    await wait(5); // Allow the delete to finish.
    console.log(`Asset ${state.asset.assetName} was deleted successfully.`);
  } catch (caught) {
    if (caught.name === "ResourceNotFoundException") {
      console.log(`The Asset ${state.asset.assetName} was not found.`);
    } else {
      console.log(`When deleting the asset: ${caught.message}`);
    }
  }
}

await wait(30); // Allow asset deletion to finish.
try {
  await state.iotSiteWiseClient.send(
    new DeleteAssetModelCommand({
      assetModelId: state.assetModel.assetModelId,
    }),
  );
  console.log(
    `Asset Model ${state.assetModel.assetModelName} was deleted
successfully.`,
  );
} catch (caught) {
  if (caught.name === "ResourceNotFoundException") {
    console.log(
      `The Asset Model ${state.assetModel.assetModelName} was not found.`,
    );
  } else {
    console.log(`When deleting the asset model: ${caught.message}`);
  }
}
```

```
try {
  await state.cloudFormationClient.send(
    new DeleteStackCommand({
      StackName: stackName,
    }),
  );
  await waitUntilStackDeleteComplete(
    { client: state.cloudFormationClient },
    { StackName: stackName },
  );
  console.log("The stack was deleted successfully.");
} catch (caught) {
  console.log(
    `${caught.message}. The stack was NOT deleted. Please clean up the
resources manually.`
  );
}
},
{ skipWhen: (/** @type {} */ state) => !state.askToDeleteResources },
);

const goodbye = new ScenarioOutput(
  "goodbye",
  "This concludes the IoT SiteWise Basics scenario for the AWS Javascript SDK v3.
Thank you!",
);

const myScenario = new Scenario(
  "IoTSiteWise Basics",
  [
    greet,
    pressEnter,
    displayBuildCloudFormationStack,
    sdkBuildCloudFormationStack,
    pressEnter,
    displayCreateAWSSiteWiseAssetModel,
    sdkCreateAWSSiteWiseAssetModel,
    displayCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    waitThirtySeconds,
    sdkCreateAWSIoTSiteWiseAssetModel,
    pressEnter,
    displayRetrievePropertyId,
```

```

    sdkRetrievePropertyId,
    pressEnter,
    displaySendDataToIoTSiteWiseAsset,
    sdkSendDataToIoTSiteWiseAsset,
    pressEnter,
    displayRetrieveValueOfIoTSiteWiseAsset,
    sdkRetrieveValueOfIoTSiteWiseAsset,
    pressEnter,
    displayCreateIoTSiteWisePortal,
    sdkCreateIoTSiteWisePortal,
    pressEnter,
    displayDescribePortal,
    sdkDescribePortal,
    pressEnter,
    displayCreateIoTSiteWiseGateway,
    sdkCreateIoTSiteWiseGateway,
    pressEnter,
    displayDescribeIoTSiteWiseGateway,
    sdkDescribeIoTSiteWiseGateway,
    pressEnter,
    askToDeleteResources,
    displayConfirmDeleteResources,
    sdkDeleteResources,
    goodbye,
  ],
  {
    iotSiteWiseClient: new IoTSiteWiseClient({}),
    cloudFormationClient: new CloudFormationClient({}),
    asset: { assetName: "MyAsset1" },
    assetModel: { assetModelName: "MyAssetModel1" },
    portal: { portalName: "MyPortal1" },
    gateway: { gatewayName: "MyGateway1" },
    propertyIds: [],
    contactEmail: "user@mydomain.com",
    thing: "MyThing1",
    sampleData: { temperature: 23.5, humidity: 65.0 },
  },
);

/** @type {{ stepHandlerOptions: StepHandlerOptions }} */
export const main = async (stepHandlerOptions) => {
  await myScenario.run(stepHandlerOptions);
};

```



```
// Invoke main function if this file was run directly.
if (process.argv[1] === fileURLToPath(import.meta.url)) {
  const { values } = parseArgs({
    options: {
      yes: {
        type: "boolean",
        short: "y",
      },
    },
  });
  main({ confirmAll: values.yes });
}
```

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

Exécutez un scénario interactif à une invite de commande.

```
class IoTSiteWiseGettingStarted:
    """
    A scenario that demonstrates how to use Boto3 to manage IoT physical assets
    using
    the AWS IoT SiteWise.
    """

    def __init__(
        self,
        iot_siteWise_wrapper: IoTSiteWiseWrapper,
        cloud_formation_resource: ServiceResource,
    ):
        self.iot_siteWise_wrapper = iot_siteWise_wrapper
        self.cloud_formation_resource = cloud_formation_resource
        self.stack = None
```

```

        self.asset_model_id = None
        self.asset_id = None
        self.portal_id = None
        self.gateway_id = None

```

```

def run(self) -> None:
    """
    Runs the scenario.
    """
    print(
        """

```

AWS IoT SiteWise is a fully managed software-as-a-service (SaaS) that makes it easy to collect, store, organize, and monitor data from industrial equipment and processes.

It is designed to help industrial and manufacturing organizations collect data from their equipment and processes, and use that data to make informed decisions about their operations.

One of the key features of AWS IoT SiteWise is its ability to connect to a wide range of industrial equipment and systems, including programmable logic controllers (PLCs), sensors, and other industrial devices. It can collect data from these devices and organize it into a unified data model, making it easier to analyze and gain insights from the data. AWS IoT SiteWise also provides tools for visualizing the data, setting up alarms and alerts, and generating reports.

Another key feature of AWS IoT SiteWise is its ability to scale to handle large volumes of data.

It can collect and store data from thousands of devices and process millions of data points per second, making it suitable for large-scale industrial operations. Additionally, AWS IoT SiteWise is designed to be secure and compliant, with features like role-based access controls, data encryption, and integration with other AWS services for additional security and compliance features.

Let's get started...

```

    """
)
    press_enter_to_continue()
    print_dashes()

```

```

        print(f'')
        print(
            f'Use AWS CloudFormation to create an IAM role that is required for
this scenario.'
        )
        template_file = IoTSitewiseGettingStarted.get_template_as_string()

        self.stack = self.deploy_cloudformation_stack(
            "python-iot-sitewise-basics", template_file
        )
        outputs = self.stack.outputs
        iam_role = None

        for output in outputs:
            if output.get("OutputKey") == "SitewiseRoleArn":
                iam_role = output.get("OutputValue")

        if iam_role is None:
            error_string = f'Failed to retrieve iam_role from CloudFormation
stack.'
            logger.error(error_string)
            raise ValueError(error_string)

        print(f'The ARN of the IAM role is {iam_role}')
        print_dashes()
        print_dashes()
        print(f'1. Create an AWS SiteWise Asset Model')
        print(
            """

```

An AWS IoT SiteWise Asset Model is a way to represent the physical assets, such as equipment, processes, and systems, that exist in an industrial environment. This model provides a structured and hierarchical representation of these assets, allowing users to define the relationships and values of each asset.

This scenario creates two asset model values: temperature and humidity.

```

        """
    )
    press_enter_to_continue()
    asset_model_name = "MyAssetModel1"
    temperature_property_name = "temperature"
    humidity_property_name = "humidity"

```

```

try:
    properties = [
        {
            "name": temperature_property_name,
            "dataType": "DOUBLE",
            "type": {
                "measurement": {},
            },
        },
        {
            "name": humidity_property_name,
            "dataType": "DOUBLE",
            "type": {
                "measurement": {},
            },
        },
    ]
    self.asset_model_id = self.iot_sitewise_wrapper.create_asset_model(
        asset_model_name, properties
    )
    print(
        f"Asset Model successfully created. Asset Model ID:
{self.asset_model_id}. "
    )
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        self.asset_model_id =
self.get_model_id_for_model_name(asset_model_name)
        print(
            f"Asset Model {asset_model_name} already exists. Asset Model
ID: {self.asset_model_id}. "
        )
    else:
        raise

    press_enter_to_continue()
    print_dashes()
    print(f"2. Create an AWS IoT SiteWise Asset")
    print(
        """

```

The IoT SiteWise model that we just created defines the structure and metadata for your physical assets.

Now we create an asset from the asset model.

```

"""
)
press_enter_to_continue()

self.asset_id = self.iot_sitewise_wrapper.create_asset(
    "MyAsset1", self.asset_model_id
)

print(f"Asset created with ID: {self.asset_id}")
press_enter_to_continue()
print_dashes()
print_dashes()
print(f"3. Retrieve the property ID values")
print(
    """

```

To send data to an asset, we need to get the property ID values. In this scenario, we access the temperature and humidity property ID values.

```

"""
)
press_enter_to_continue()
property_ids = self.iot_sitewise_wrapper.list_asset_model_properties(
    self.asset_model_id
)
humidity_property_id = None
temperature_property_id = None
for property_id in property_ids:
    if property_id.get("name") == humidity_property_name:
        humidity_property_id = property_id.get("id")
    elif property_id.get("name") == temperature_property_name:
        temperature_property_id = property_id.get("id")
if humidity_property_id is None or temperature_property_id is None:
    error_string = f"Failed to retrieve property IDs from Asset Model."
    logger.error(error_string)
    raise ValueError(error_string)

print(f"The Humidity property Id is {humidity_property_id}")
print(f"The Temperature property Id is {temperature_property_id}")
press_enter_to_continue()
print_dashes()
print_dashes()

print(f"4. Send data to an AWS IoT SiteWise Asset")
print(

```

```
"""
```

By sending data to an IoT SiteWise Asset, you can aggregate data from multiple sources, normalize the data into a standard format, and store it in a centralized location. This makes it easier to analyze and gain insights from the data.

In this example, we generate sample temperature and humidity data and send it to the AWS IoT SiteWise asset.

```

"""
)
press_enter_to_continue()

values = [
    {
        "propertyId": humidity_property_id,
        "valueType": "doubleValue",
        "value": 65.0,
    },
    {
        "propertyId": temperature_property_id,
        "valueType": "doubleValue",
        "value": 23.5,
    },
]
self.iot_sitewise_wrapper.batch_put_asset_property_value(self.asset_id,
values)
print(f"Data sent successfully.")

press_enter_to_continue()
print_dashes()
print_dashes()

print(f"5. Retrieve the value of the IoT SiteWise Asset property")
print(
    """

```

IoT SiteWise is an AWS service that allows you to collect, process, and analyze industrial data from connected equipment and sensors. One of the key benefits of reading an IoT SiteWise property is the ability to gain valuable insights from your industrial data.

```
"""
```

```
)
```

```

        press_enter_to_continue()

        property_value = self.iot_sitewise_wrapper.get_asset_property_value(
            self.asset_id, temperature_property_id
        )
        print(f"The property name is '{temperature_property_name}'.")

        print(
            f"The value of this property is: {property_value['value']}
['doubleValue']}"
        )
        press_enter_to_continue()

        property_value = self.iot_sitewise_wrapper.get_asset_property_value(
            self.asset_id, humidity_property_id
        )
        print(f"The property name is '{humidity_property_name}'.")
        print(
            f"The value of this property is: {property_value['value']}
['doubleValue']}"
        )
        press_enter_to_continue()
        print_dashes()
        print_dashes()

        print(f"6. Create an IoT SiteWise Portal")
        print(
            """
An IoT SiteWise Portal allows you to aggregate data from multiple industrial
sources,
such as sensors, equipment, and control systems, into a centralized platform.
"""
        )

        press_enter_to_continue()
        contact_email = q.ask("Enter a contact email for the portal:",
q.non_empty)
        print("Creating the portal. The portal may take a while to become
active.")
        self.portal_id = self.iot_sitewise_wrapper.create_portal(
            "MyPortal1", iam_role, contact_email
        )
        print(f"Portal created successfully. Portal ID {self.portal_id}")
        press_enter_to_continue()

```

```

        print_dashes()
        print_dashes()

        print(f"7. Describe the Portal")
        print(
            """
In this step, we get a description of the portal and display the portal URL.
            """
        )
        press_enter_to_continue()
        portal_description =
self.iot_sitewise_wrapper.describe_portal(self.portal_id)
        print(f"Portal URL: {portal_description['portalStartUrl']}")
        press_enter_to_continue()
        print_dashes()
        print_dashes()

        print(f"8. Create an IoT SiteWise Gateway")
        press_enter_to_continue()
        self.gateway_id = self.iot_sitewise_wrapper.create_gateway(
            "MyGateway1", "MyThing1"
        )
        print(f"Gateway creation completed successfully. id is
{self.gateway_id}")
        print_dashes()
        print_dashes()
        print(f"9. Describe the IoT SiteWise Gateway")
        press_enter_to_continue()

        gateway_description = self.iot_sitewise_wrapper.describe_gateway(
            self.gateway_id
        )
        print(f"Gateway Name: {gateway_description['gatewayName']}")
        print(f"Gateway ARN: {gateway_description['gatewayArn']}")
        print(f"Gateway Platform:\n{gateway_description['gatewayPlatform']}")
        print(f"Gateway Creation Date: {gateway_description['gatewayArn']}")
        print_dashes()
        print_dashes()

        print(f"10. Delete the AWS IoT SiteWise Assets")
        if q.ask("Would you like to delete the IoT SiteWise Assets? (y/n)",
q.is_yesno):
            self.cleanup()
        else:

```



```

        print(f"The resources will not be deleted.")
    print_dashes()
    print_dashes()
    print(f"This concludes the AWS IoT SiteWise Scenario")

def cleanup(self) -> None:
    """
    Deletes the CloudFormation stack and the resources created for the demo.
    """

    if self.gateway_id is not None:
        self.iot_sitewise_wrapper.delete_gateway(self.gateway_id)
        print(f"Deleted gateway with id {self.gateway_id}.")
        self.gateway_id = None
    if self.portal_id is not None:
        self.iot_sitewise_wrapper.delete_portal(self.portal_id)
        print(f"Deleted portal with id {self.portal_id}.")
        self.portal_id = None
    if self.asset_id is not None:
        self.iot_sitewise_wrapper.delete_asset(self.asset_id)
        print(f"Deleted asset with id {self.asset_id}.")
        self.iot_sitewise_wrapper.wait_asset_deleted(self.asset_id)
        self.asset_id = None
    if self.asset_model_id is not None:
        self.iot_sitewise_wrapper.delete_asset_model(self.asset_model_id)
        print(f"Deleted asset model with id {self.asset_model_id}.")
        self.asset_model_id = None
    if self.stack is not None:
        stack = self.stack
        self.stack = None
        self.destroy_cloudformation_stack(stack)

def deploy_cloudformation_stack(
    self, stack_name: str, cfn_template: str
) -> ServiceResource:
    """
    Deploys prerequisite resources used by the scenario. The resources are
    defined in the associated `SitewiseRoles-template.yaml` AWS
    CloudFormation script and are deployed
    as a CloudFormation stack, so they can be easily managed and destroyed.

    :param stack_name: The name of the CloudFormation stack.
    :param cfn_template: The CloudFormation template as a string.
    :return: The CloudFormation stack resource.

```

```

    """
    print(f"Deploying CloudFormation stack: {stack_name}.")
    stack = self.cloud_formation_resource.create_stack(
        StackName=stack_name,
        TemplateBody=cfn_template,
        Capabilities=["CAPABILITY_NAMED_IAM"],
    )
    print(f"CloudFormation stack creation started: {stack_name}")
    print("Waiting for CloudFormation stack creation to complete...")
    waiter = self.cloud_formation_resource.meta.client.get_waiter(
        "stack_create_complete"
    )
    waiter.wait(StackName=stack.name)
    stack.load()
    print("CloudFormation stack creation complete.")

    return stack

def destroy_cloudformation_stack(self, stack: ServiceResource) -> None:
    """
    Destroys the resources managed by the CloudFormation stack, and the
    CloudFormation
    stack itself.

    :param stack: The CloudFormation stack that manages the example
    resources.
    """
    print(
        f"CloudFormation stack '{stack.name}' is being deleted. This may take
    a few minutes."
    )
    stack.delete()
    waiter = self.cloud_formation_resource.meta.client.get_waiter(
        "stack_delete_complete"
    )
    waiter.wait(StackName=stack.name)
    print(f"CloudFormation stack '{stack.name}' has been deleted.")

@staticmethod
def get_template_as_string() -> str:
    """
    Returns a string containing this scenario's CloudFormation template.
    """

```

```

        template_file_path = os.path.join(script_dir, "SitewiseRoles-
template.yaml")
        file = open(template_file_path, "r")
        return file.read()

def get_model_id_for_model_name(self, model_name: str) -> str:
    """
    Returns the model ID for the given model name.

    :param model_name: The name of the model.
    :return: The model ID.
    """
    model_id = None
    asset_models = self.iot_sitewise_wrapper.list_asset_models()
    for asset_model in asset_models:
        if asset_model["name"] == model_name:
            model_id = asset_model["id"]
            break
    return model_id

```

Classe IoTSitewise Wrapper qui enveloppe les actions. AWS IoT SiteWise

```

class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """

```

```

        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

def create_asset(self, asset_name: str, asset_model_id: str) -> str:

```

```

"""
Creates an AWS IoT SiteWise Asset.

:param asset_name: The name of the asset to create.
:param asset_model_id: The ID of the asset model to associate with the
asset.
:return: The ID of the created asset.
"""
try:
    response = self.iotsitewise_client.create_asset(
        assetName=asset_name, assetModelId=asset_model_id
    )
    asset_id = response["assetId"]
    waiter = self.iotsitewise_client.get_waiter("asset_active")
    waiter.wait(assetId=asset_id)
    return asset_id
except ClientError as err:
    if err.response["Error"] == "ResourceNotFoundException":
        logger.error("Asset model %s does not exist.", asset_model_id)
    else:
        logger.error(
            "Error creating asset %s. Here's why %s",
            asset_name,
            err.response["Error"]["Message"],
        )
    raise


def list_asset_models(self) -> List[Dict[str, Any]]:
    """
    Lists all AWS IoT SiteWise Asset Models.

    :return: A list of dictionaries containing information about each asset
    model.

    """
    try:
        asset_models = []
        paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
        pages = paginator.paginate()
        for page in pages:
            asset_models.extend(page["assetModelSummaries"])
        return asset_models

```

```

    except ClientError as err:
        logger.error(
            "Error listing asset models. Here's why %s",
            err.response["Error"]["Message"],
        )
        raise

    def list_asset_model_properties(self, asset_model_id: str) -> List[Dict[str,
Any]]:
        """
        Lists all AWS IoT SiteWise Asset Model Properties.

        :param asset_model_id: The ID of the asset model to list values for.
        :return: A list of dictionaries containing information about each asset
model property.
        """
        try:
            asset_model_properties = []
            paginator = self.iotsitewise_client.get_paginator(
                "list_asset_model_properties"
            )
            pages = paginator.paginate(assetModelId=asset_model_id)
            for page in pages:
                asset_model_properties.extend(page["assetModelPropertySummaries"])
            return asset_model_properties
        except ClientError as err:
            logger.error(
                "Error listing asset model values. Here's why %s",
                err.response["Error"]["Message"],
            )
            raise

    def batch_put_asset_property_value(
        self, asset_id: str, values: List[Dict[str, str]]
    ) -> None:
        """
        Sends data to an AWS IoT SiteWise Asset.

        :param asset_id: The asset ID.
        :param values: A list of dictionaries containing the values in the form
{propertyId : property_id,

```

```

        valueType : [stringValue|integerValue|doubleValue|
booleanValue],
        value : the_value}.

"""
try:
    entries = self.properties_to_values(asset_id, values)

self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Asset %s does not exist.", asset_id)
    else:
        logger.error(
            "Error sending data to asset. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form
        {propertyId : property_id,
        valueType : [stringValue|integerValue|doubleValue|
booleanValue],
        value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
    batch_put_asset_property_value.
    """
    entries = []
    for value in values:
        epoch_ns = time.time_ns()
        self.entry_id += 1
        if value["valueType"] == "stringValue":
            property_value = {"stringValue": value["value"]}
        elif value["valueType"] == "integerValue":
            property_value = {"integerValue": value["value"]}
        elif value["valueType"] == "booleanValue":
            property_value = {"booleanValue": value["value"]}

```

```

        elif value["valueType"] == "doubleValue":
            property_value = {"doubleValue": value["value"]}
        else:
            raise ValueError("Invalid valueType: %s", value["valueType"])
    entry = {
        "entryId": f"{self.entry_id}",
        "assetId": asset_id,
        "propertyId": value["propertyId"],
        "propertyValues": [
            {
                "value": property_value,
                "timestamp": {
                    "timeInSeconds": int(epoch_ns / 1000000000),
                    "offsetInNanos": epoch_ns % 1000000000,
                },
            }
        ],
    }
    entries.append(entry)
return entries

def get_asset_property_value(
    self, asset_id: str, property_id: str
) -> Dict[str, Any]:
    """
    Gets the value of an AWS IoT SiteWise Asset Property.

    :param asset_id: The ID of the asset.
    :param property_id: The ID of the property.
    :return: A dictionary containing the value of the property.
    """
    try:
        response = self.iotsitewise_client.get_asset_property_value(
            assetId=asset_id, propertyId=property_id
        )
        return response["propertyValue"]
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error(
                "Asset %s or property %s does not exist.", asset_id,
property_id
            )
        else:

```



```

        logger.error(
            "Error getting asset property value. Here's why %s",
            err.response["Error"]["Message"],
        )
    raise

def create_portal(
    self, portal_name: str, iam_role_arn: str, portal_contact_email: str
) -> str:
    """
    Creates an AWS IoT SiteWise Portal.

    :param portal_name: The name of the portal to create.
    :param iam_role_arn: The ARN of an IAM role.
    :param portal_contact_email: The contact email of the portal.
    :return: The ID of the created portal.
    """
    try:
        response = self.iotsitewise_client.create_portal(
            portalName=portal_name,
            roleArn=iam_role_arn,
            portalContactEmail=portal_contact_email,
        )
        portal_id = response["portalId"]
        waiter = self.iotsitewise_client.get_waiter("portal_active")
        waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
        return portal_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Portal %s already exists.", portal_name)
        else:
            logger.error(
                "Error creating portal %s. Here's why %s",
                portal_name,
                err.response["Error"]["Message"],
            )
    raise

def describe_portal(self, portal_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Portal.

```

```

        :param portal_id: The ID of the portal to describe.
        :return: A dictionary containing information about the portal.
        """
        try:
            response =
self.iotsitewise_client.describe_portal(portalId=portal_id)
            return response
        except ClientError as err:
            logger.error(
                "Error describing portal %s. Here's why %s",
                portal_id,
                err.response["Error"]["Message"],
            )
            raise

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Gateway %s already exists.", gateway_name)
        else:
            logger.error(
                "Error creating gateway %s. Here's why %s",
                gateway_name,
                err.response["Error"]["Message"],
            )
            raise

```

```
def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
    """
    Describes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to describe.
    :return: A dictionary containing information about the gateway.
    """
    try:
        response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
        return response
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error describing gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise
```

```
def delete_portal(self, portal_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Portal.

    :param portal_id: The ID of the portal to delete.
    """
    try:
        self.iotsitewise_client.delete_portal(portalId=portal_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Portal %s does not exist.", portal_id)
        else:
            logger.error(
                "Error deleting portal %s. Here's why %s",
                portal_id,
                err.response["Error"]["Message"],
            )
        raise

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

def delete_asset_model(self, asset_model_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset Model.

    :param asset_model_id: The ID of the asset model to delete.
    """
```

```

try:

self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
except ClientError as err:
    logger.error(
        "Error deleting asset model %s. Here's why %s",
        asset_model_id,
        err.response["Error"]["Message"],
    )
    raise

def wait_asset_deleted(self, asset_id: str) -> None:
    """
    Waits for an AWS IoT SiteWise Asset to be deleted.

    :param asset_id: The ID of the asset to wait for.
    """
    try:
        waiter = self.iotsitewise_client.get_waiter("asset_not_exists")
        waiter.wait(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error waiting for asset %s to be deleted. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

```

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Actions d' AWS IoT SiteWise utilisation AWS SDKs

Les exemples de code suivants montrent comment effectuer des AWS IoT SiteWise actions individuelles avec AWS SDKs. Chaque exemple inclut un lien vers GitHub, où vous pouvez trouver des instructions pour configurer et exécuter le code.

Les exemples suivants incluent uniquement les actions les plus couramment utilisées. Pour obtenir la liste complète, veuillez consulter la [AWS IoT SiteWise Référence d'API](#).

Exemples

- [Utilisation BatchPutAssetPropertyValue avec un AWS SDK ou une CLI](#)
- [Utilisation CreateAsset avec un AWS SDK ou une CLI](#)
- [Utilisation CreateAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation CreateGateway avec un AWS SDK ou une CLI](#)
- [Utilisation CreatePortal avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteAsset avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation DeleteGateway avec un AWS SDK ou une CLI](#)
- [Utilisation DeletePortal avec un AWS SDK ou une CLI](#)
- [Utilisation DescribeAssetModel avec un AWS SDK ou une CLI](#)
- [Utilisation DescribeGateway avec un AWS SDK ou une CLI](#)
- [Utilisation DescribePortal avec un AWS SDK ou une CLI](#)
- [Utilisation GetAssetPropertyValue avec un AWS SDK ou une CLI](#)
- [Utilisation ListAssetModels avec un AWS SDK ou une CLI](#)

Utilisation **BatchPutAssetPropertyValue** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser BatchPutAssetPropertyValue.

CLI

AWS CLI

Pour envoyer des données aux propriétés des actifs

L'batch-put-asset-property-valueexemple suivant envoie des données de puissance et de température aux propriétés des actifs identifiées par des alias de propriété.

```
aws iotsitewise batch-put-asset-property-value \  
  --cli-input-json file://batch-put-asset-property-value.json
```

Contenu de batch-put-asset-property-value.json :

```
{
  "entries": [
    {
      "entryId": "1575691200-company-windfarm-3-turbine-7-power",
      "propertyAlias": "company-windfarm-3-turbine-7-power",
      "propertyValues": [
        {
          "value": {
            "doubleValue": 4.92
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          },
          "quality": "GOOD"
        }
      ]
    },
    {
      "entryId": "1575691200-company-windfarm-3-turbine-7-temperature",
      "propertyAlias": "company-windfarm-3-turbine-7-temperature",
      "propertyValues": [
        {
          "value": {
            "integerValue": 38
          },
          "timestamp": {
            "timeInSeconds": 1575691200
          }
        }
      ]
    }
  ]
}
```

Sortie :

```
{
  "errorEntries": []
}
```

Pour plus d'informations, consultez la section [Ingestion de données à l'aide de l' SiteWise API AWS IoT](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [BatchPutAssetPropertyValue](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Sends data to the SiteWise service.
 *
 * @param assetId      the ID of the asset to which the data will be sent.
 * @param tempPropertyId the ID of the temperature property.
 * @param humidityPropId the ID of the humidity property.
 * @return a {@link CompletableFuture} that represents a {@link
BatchPutAssetPropertyValueResponse} result. The
 *      calling code can attach callbacks, then handle the result or
exception by calling
 *      {@link CompletableFuture#join()} or {@link
CompletableFuture#get()}.
 *      <p>
 *      If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *      available to the calling code as a {@link CompletionException}. By
calling
 *      {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<BatchPutAssetPropertyValueResponse>
sendDataToSiteWiseAsync(String assetId, String tempPropertyId, String
humidityPropId) {
    Map<String, Double> sampleData = generateSampleData();
    long timestamp = Instant.now().toEpochMilli();

    TimeInNanos time = TimeInNanos.builder()
        .timeInSeconds(timestamp / 1000)
```



```

        .offsetInNanos((int) ((timestamp % 1000) * 1000000))
        .build();

    BatchPutAssetPropertyValueRequest request =
    BatchPutAssetPropertyValueRequest.builder()
        .entries(Arrays.asList(
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-3")
                .assetId(assetId)
                .propertyId(tempPropertyId)
                .propertyValues(Arrays.asList(
                    AssetPropertyValue.builder()
                        .value(Variant.builder()
                            .doubleValue(sampleData.get("Temperature"))
                            .build())
                        .timestamp(time)
                        .build()
                ))
                .build(),
            PutAssetPropertyValueEntry.builder()
                .entryId("entry-4")
                .assetId(assetId)
                .propertyId(humidityPropId)
                .propertyValues(Arrays.asList(
                    AssetPropertyValue.builder()
                        .value(Variant.builder()
                            .doubleValue(sampleData.get("Humidity"))
                            .build())
                        .timestamp(time)
                        .build()
                ))
                .build()
        ))
        .build();

    return getAsyncClient().batchPutAssetPropertyValue(request)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An exception occurred: {}",
exception.getCause().getMessage());
            }
        });
}

```

- Pour plus de détails sur l'API, voir [BatchPutAssetPropertyValue](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  BatchPutAssetPropertyValueCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Batch put asset property values.
 * @param {{ entries : array }}
 */
export const main = async ({ entries }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new BatchPutAssetPropertyValueCommand({
        entries: entries,
      }),
    );
    console.log("Asset properties batch put successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(`${caught.message}. A resource could not be found.`);
    } else {
      throw caught;
    }
  }
}
```

```
}
};
```

- Pour plus de détails sur l'API, voir [BatchPutAssetPropertyValue](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
```

```

        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def batch_put_asset_property_value(
        self, asset_id: str, values: List[Dict[str, str]]
    ) -> None:
        """
        Sends data to an AWS IoT SiteWise Asset.

        :param asset_id: The asset ID.
        :param values: A list of dictionaries containing the values in the form
            {propertyId : property_id,
             valueType : [stringValue|integerValue|doubleValue|
booleanValue],
             value : the_value}.
        """
        try:
            entries = self.properties_to_values(asset_id, values)

        self.iotsitewise_client.batch_put_asset_property_value(entries=entries)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Asset %s does not exist.", asset_id)
            else:
                logger.error(
                    "Error sending data to asset. Here's why %s",
                    err.response["Error"]["Message"],
                )
            raise

```

Une fonction d'assistance pour générer le paramètre d'entrées à partir d'une liste de valeurs.

```

def properties_to_values(
    self, asset_id: str, values: list[dict[str, Any]]
) -> list[dict[str, Any]]:
    """
    Utility function to convert a values list to the entries parameter for
    batch_put_asset_property_value.
    :param asset_id : The asset ID.
    :param values : A list of dictionaries containing the values in the form

```

```

        {propertyId : property_id,
         valueType : [stringValue|integerValue|doubleValue|
booleanValue],
         value : the_value}.
    :return: An entries list to pass as the 'entries' parameter to
batch_put_asset_property_value.
"""
entries = []
for value in values:
    epoch_ns = time.time_ns()
    self.entry_id += 1
    if value["valueType"] == "stringValue":
        property_value = {"stringValue": value["value"]}
    elif value["valueType"] == "integerValue":
        property_value = {"integerValue": value["value"]}
    elif value["valueType"] == "booleanValue":
        property_value = {"booleanValue": value["value"]}
    elif value["valueType"] == "doubleValue":
        property_value = {"doubleValue": value["value"]}
    else:
        raise ValueError("Invalid valueType: %s", value["valueType"])
    entry = {
        "entryId": f"{self.entry_id}",
        "assetId": asset_id,
        "propertyId": value["propertyId"],
        "propertyValues": [
            {
                "value": property_value,
                "timestamp": {
                    "timeInSeconds": int(epoch_ns / 1000000000),
                    "offsetInNanos": epoch_ns % 1000000000,
                },
            },
        ],
    }
    entries.append(entry)
return entries

```

Voici un exemple de liste de valeurs à transmettre à la fonction d'assistance.

```
values = [
```

```
{
  "propertyId": humidity_property_id,
  "valueType": "doubleValue",
  "value": 65.0,
},
{
  "propertyId": temperature_property_id,
  "valueType": "doubleValue",
  "value": 23.5,
},
]
```

- Pour plus de détails sur l'API, consultez [BatchPutAssetPropertyValue](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateAsset** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser CreateAsset.

CLI

AWS CLI

Pour créer un actif

L'`create-asset` exemple suivant crée un actif d'éolienne à partir d'un modèle d'actif d'éolienne.

```
aws iotsitewise create-asset \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-1111EXAMPLE \
  --asset-name "Wind Turbine 1"
```

Sortie :

```
{
  "assetId": "a1b2c3d4-5678-90ab-cdef-3333EXAMPLE",
```

```

    "assetArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset/
a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
    "assetStatus": {
        "state": "CREATING"
    }
}

```

Pour plus d'informations, consultez la section [Création d'actifs](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [CreateAsset](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

/**
 * Creates an asset with the specified name and asset model Id.
 *
 * @param assetName    the name of the asset to create.
 * @param assetModelId the Id of the asset model to associate with the asset.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */

```

```

public CompletableFuture<CreateAssetResponse> createAssetAsync(String
assetName, String assetModelId) {
    CreateAssetRequest createAssetRequest = CreateAssetRequest.builder()
        .assetModelId(assetModelId)
        .assetDescription("Created using the AWS SDK for Java")
        .assetName(assetName)
        .build();

    return getAsyncClient().createAsset(createAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to create asset: {}",
exception.getCause().getMessage());
            }
        });
}

```

- Pour plus de détails sur l'API, voir [CreateAsset](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    CreateAssetCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset.
 * @param {{ assetName : string, assetModelId: string }}
 */

```



```
export const main = async ({ assetName, assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateAssetCommand({
        assetName: assetName, // The name to give the Asset.
        assetModelId: assetModelId, // The ID of the asset model from which to
        create the asset.
      })),
    );
    console.log("Asset created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The asset model could not be found. Please check the
        asset model id.`
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, voir [CreateAsset](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""
```

```

def __init__(self, iotsitewise_client: client) -> None:
    """
    Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
                                access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_asset(self, asset_name: str, asset_model_id: str) -> str:
        """
        Creates an AWS IoT SiteWise Asset.

        :param asset_name: The name of the asset to create.
        :param asset_model_id: The ID of the asset model to associate with the
        asset.

        :return: The ID of the created asset.
        """
        try:
            response = self.iotsitewise_client.create_asset(
                assetName=asset_name, assetModelId=asset_model_id
            )
            asset_id = response["assetId"]
            waiter = self.iotsitewise_client.get_waiter("asset_active")
            waiter.wait(assetId=asset_id)
            return asset_id
        except ClientError as err:
            if err.response["Error"] == "ResourceNotFoundException":

```

```

        logger.error("Asset model %s does not exist.", asset_model_id)
    else:
        logger.error(
            "Error creating asset %s. Here's why %s",
            asset_name,
            err.response["Error"]["Message"],
        )
    raise

```

- Pour plus de détails sur l'API, consultez [CreateAsset](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateAssetModel** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreateAssetModel`.

CLI

AWS CLI

Pour créer un modèle d'actifs

L'`create-asset-model` exemple suivant crée un modèle d'actif qui définit une éolienne avec les propriétés suivantes :

Numéro de série - Le numéro de série d'une éolienne
 L'énergie produite par une éolienne
 - Le flux de données de puissance généré par une éolienne
 Température C - Le flux de données de température d'une éolienne en Celsius
 Temperature F - Les points de données de température cartographiés de Celsius à Fahrenheit

```

aws iotsitewise create-asset-model \
  --cli-input-json file://create-wind-turbine-model.json

```

Contenu de `create-wind-turbine-model.json` :

```
{
  "assetModelName": "Wind Turbine Model",
  "assetModelDescription": "Represents a wind turbine",
  "assetModelProperties": [
    {
      "name": "Serial Number",
      "dataType": "STRING",
      "type": {
        "attribute": {}
      }
    },
    {
      "name": "Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature C",
      "dataType": "DOUBLE",
      "unit": "Celsius",
      "type": {
        "measurement": {}
      }
    },
    {
      "name": "Temperature F",
      "dataType": "DOUBLE",
      "unit": "Fahrenheit",
      "type": {
        "transform": {
          "expression": "temp_c * 9 / 5 + 32",
          "variables": [
            {
              "name": "temp_c",
              "value": {
                "propertyId": "Temperature C"
              }
            }
          ]
        }
      }
    }
  ]
}
```

```

    }
  },
  {
    "name": "Total Generated Power",
    "dataType": "DOUBLE",
    "unit": "kW",
    "type": {
      "metric": {
        "expression": "sum(power)",
        "variables": [
          {
            "name": "power",
            "value": {
              "propertyId": "Generated Power"
            }
          }
        ],
        "window": {
          "tumbling": {
            "interval": "1h"
          }
        }
      }
    }
  }
]
}

```

Sortie :

```

{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
  "assetModelStatus": {
    "state": "CREATING"
  }
}

```

Pour plus d'informations, consultez la section [Définition des modèles d'actifs](#) dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [CreateAssetModel](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Creates an asset model.
 *
 * @param name the name of the asset model to create.
 * @return a {@link CompletableFuture} that represents a {@link
CreateAssetModelResponse} result. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps it
 *         available to the calling code as a {@link CompletionException}. By
calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<CreateAssetModelResponse>
createAssetModelAsync(String name) {
    PropertyType humidity = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();

    PropertyType temperaturePropertyType = PropertyType.builder()
        .measurement(Measurement.builder().build())
        .build();
```

```
AssetModelPropertyDefinition temperatureProperty =
AssetModelPropertyDefinition.builder()
    .name("Temperature")
    .dataType(PropertyDataType.DOUBLE)
    .type(temperaturePropertyType)
    .build();

AssetModelPropertyDefinition humidityProperty =
AssetModelPropertyDefinition.builder()
    .name("Humidity")
    .dataType(PropertyDataType.DOUBLE)
    .type(humidity)
    .build();

CreateAssetModelRequest createAssetModelRequest =
CreateAssetModelRequest.builder()
    .assetModelName(name)
    .assetModelDescription("This is my asset model")
    .assetModelProperties(temperatureProperty, humidityProperty)
    .build();

return getAsyncClient().createAssetModel(createAssetModelRequest)
    .whenComplete((response, exception) -> {
        if (exception != null) {
            logger.error("Failed to create asset model: {} ",
exception.getCause().getMessage());
        }
    });
}
```

- Pour plus de détails sur l'API, voir [CreateAssetModel](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  CreateAssetModelCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an Asset Model.
 * @param {{ assetName : string, assetModelId: string }}
 */
export const main = async ({ assetModelName, assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateAssetModelCommand({
        assetModelName: assetModelName, // The name to give the Asset Model.
      }),
    );
    console.log("Asset model created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the asset model.`
      );
    } else {
      throw caught;
    }
  }
};
```


- Pour plus de détails sur l'API, voir [CreateAssetModel](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```

def create_asset_model(
    self, asset_model_name: str, properties: List[Dict[str, Any]]
) -> str:
    """
    Creates an AWS IoT SiteWise Asset Model.

    :param asset_model_name: The name of the asset model to create.
    :param properties: The property definitions of the asset model.
    :return: The ID of the created asset model.
    """
    try:
        response = self.iotsitewise_client.create_asset_model(
            assetModelName=asset_model_name,
            assetModelDescription="This is a sample asset model
description.",
            assetModelProperties=properties,
        )
        asset_model_id = response["assetModelId"]
        waiter = self.iotsitewise_client.get_waiter("asset_model_active")
        waiter.wait(assetModelId=asset_model_id)
        return asset_model_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Asset model %s already exists.", asset_model_name)
        else:
            logger.error(
                "Error creating asset model %s. Here's why %s",
                asset_model_name,
                err.response["Error"]["Message"],
            )
        raise

```

Voici un exemple de liste de propriétés à transmettre à la fonction.

```

properties = [
    {
        "name": temperature_property_name,
        "dataType": "DOUBLE",
        "type": {
            "measurement": {},
        },
    },

```

```

    },
    {
      "name": humidity_property_name,
      "dataType": "DOUBLE",
      "type": {
        "measurement": {},
      },
    },
  ],
]

```

- Pour plus de détails sur l'API, consultez [CreateAssetModelle](#) AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreateGateway** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser CreateGateway.

CLI

AWS CLI

Pour créer une passerelle

L'`create-gateway` exemple suivant crée une passerelle qui s'exécute sur AWS IoT Greengrass.

```

aws iotsitewise create-gateway \
  --gateway-name ExampleCorpGateway \
  --gateway-platform greengrass={groupArn=arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE}

```

Sortie :

```

{
  "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
  "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE"
}

```

```
}
```

Pour plus d'informations, consultez [la section Configuration d'une passerelle](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [CreateGateway](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Creates a new IoT Sitewise gateway.
 *
 * @param gatewayName The name of the gateway to create.
 * @param myThing      The name of the core device thing to associate with the
gateway.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the gateways ID. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createGatewayAsync(String gatewayName,
String myThing) {
    GreengrassV2 gg = GreengrassV2.builder()
```

```
        .coreDeviceThingName(myThing)
        .build();

GatewayPlatform platform = GatewayPlatform.builder()
    .greengrassV2(gg)
    .build();

Map<String, String> tag = new HashMap<>();
tag.put("Environment", "Production");

CreateGatewayRequest createGatewayRequest =
CreateGatewayRequest.builder()
    .gatewayName(gatewayName)
    .gatewayPlatform(platform)
    .tags(tag)
    .build();

return getAsyncClient().createGateway(createGatewayRequest)
    .handle((response, exception) -> {
        if (exception != null) {
            logger.error("Error creating the gateway.");
            throw (CompletionException) exception;
        }
        logger.info("The ARN of the gateway is {}" ,
response.gatewayArn());
        return response.gatewayId();
    });
}
```

- Pour plus de détails sur l'API, voir [CreateGateway](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  CreateGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Gateway.
 * @param {{ }}
 */
export const main = async ({ gatewayName }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new CreateGatewayCommand({
        gatewayName: gatewayName, // The name to give the created Gateway.
      }),
    );
    console.log("Gateway created successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem creating the Gateway.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, voir [CreateGateway](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_gateway(self, gateway_name: str, my_thing: str) -> str:
        """
        Creates an AWS IoT SiteWise Gateway.
```

```

:param gateway_name: The name of the gateway to create.
:param my_thing: The core device thing name.
:return: The ID of the created gateway.
"""
try:
    response = self.iotsitewise_client.create_gateway(
        gatewayName=gateway_name,
        gatewayPlatform={
            "greengrassV2": {"coreDeviceThingName": my_thing},
        },
        tags={"Environment": "Production"},
    )
    gateway_id = response["gatewayId"]
    return gateway_id
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
        logger.error("Gateway %s already exists.", gateway_name)
    else:
        logger.error(
            "Error creating gateway %s. Here's why %s",
            gateway_name,
            err.response["Error"]["Message"],
        )
    raise

```

- Pour plus de détails sur l'API, consultez [CreateGateway](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **CreatePortal** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `CreatePortal`.

CLI

AWS CLI

Pour créer un portail

L'`create-portal` suivant crée un portail Web pour une société de parc éolien. Vous ne pouvez créer des portails que dans la région où vous avez activé l'authentification AWS unique.

```
aws iotsitewise create-portal \
  --portal-name WindFarmPortal \
  --portal-description "A portal that contains wind farm projects for Example Corp." \
  --portal-contact-email support@example.com \
  --role-arn arn:aws:iam::123456789012:role/service-role/MySiteWiseMonitorServiceRole
```

Sortie :

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalStatus": {
    "state": "CREATING"
  },
  "ssoApplicationId": "ins-a1b2c3d4-EXAMPLE"
}
```

Pour plus d'informations, consultez [Getting started with AWS IoT SiteWise Monitor](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT et [Enabling AWS SSO](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [CreatePortal](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Creates a new IoT SiteWise portal.
 *
 * @param portalName the name of the portal to create.
 * @param iamRole the IAM role ARN to use for the portal.
 * @param contactEmail the email address of the portal contact.
 * @return a {@link CompletableFuture} that represents a {@link String}
result of the portal ID. The calling code
 * can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 * {@link CompletableFuture#get()}.
 *
 * <p>
 * If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 * it available to the calling code as a {@link CompletionException}.
By calling
 * {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<String> createPortalAsync(String portalName, String
iamRole, String contactEmail) {
    CreatePortalRequest createPortalRequest = CreatePortalRequest.builder()
        .portalName(portalName)
        .portalDescription("This is my custom IoT SiteWise portal.")
        .portalContactEmail(contactEmail)
        .roleArn(iamRole)
        .build();

    return getAsyncClient().createPortal(createPortalRequest)
        .handle((response, exception) -> {
            if (exception != null) {
```

```

        logger.error("Failed to create portal: {} ",
exception.getCause().getMessage());
        throw (CompletionException) exception;
    }
    return response.portalId();
});
}

```

- Pour plus de détails sur l'API, voir [CreatePortal](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    CreatePortalCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create a Portal.
 * @param {{ portalName: string, portalContactEmail: string, roleArn: string }}
 */
export const main = async ({ portalName, portalContactEmail, roleArn }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new CreatePortalCommand({
                portalName: portalName, // The name to give the created Portal.
                portalContactEmail: portalContactEmail, // A valid contact email.
                roleArn: roleArn, // The ARN of a service role that allows the portal's
                users to access the portal's resources.
            })
        );
    } catch (error) {
        console.error(error);
    }
};

```

```

    }},
    );
    console.log("Portal created successfully.");
    return result;
} catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
        console.warn(
            `${caught.message}. There was a problem creating the Portal.`
        );
    } else {
        throw caught;
    }
}
};

```

- Pour plus de détails sur l'API, voir [CreatePortal](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client

```

```

        self.entry_id = 0 # Incremented to generate unique entry IDs for
batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def create_portal(
        self, portal_name: str, iam_role_arn: str, portal_contact_email: str
    ) -> str:
        """
        Creates an AWS IoT SiteWise Portal.

        :param portal_name: The name of the portal to create.
        :param iam_role_arn: The ARN of an IAM role.
        :param portal_contact_email: The contact email of the portal.
        :return: The ID of the created portal.
        """
        try:
            response = self.iotsitewise_client.create_portal(
                portalName=portal_name,
                roleArn=iam_role_arn,
                portalContactEmail=portal_contact_email,
            )
            portal_id = response["portalId"]
            waiter = self.iotsitewise_client.get_waiter("portal_active")
            waiter.wait(portalId=portal_id, WaiterConfig={"MaxAttempts": 40})
            return portal_id
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
                logger.error("Portal %s already exists.", portal_name)
            else:
                logger.error(
                    "Error creating portal %s. Here's why %s",
                    portal_name,

```

```
err.response["Error"]["Message"],
    )
    raise
```

- Pour plus de détails sur l'API, consultez [CreatePortale](#) AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteAsset** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DeleteAsset.

CLI

AWS CLI

Pour supprimer un actif

L'`delete-asset`exemple suivant supprime un actif d'éolienne.

```
aws iotsitewise delete-asset \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE
```

Sortie :

```
{
  "assetStatus": {
    "state": "DELETING"
  }
}
```

Pour plus d'informations, consultez [la section Suppression d'actifs](#) dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [DeleteAsset](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Deletes an asset.
 *
 * @param assetId the ID of the asset to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetResponse} result. The calling code can
 *         attach callbacks, then handle the result or exception by calling
{@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteAssetResponse> deleteAssetAsync(String
assetId) {
    DeleteAssetRequest deleteAssetRequest = DeleteAssetRequest.builder()
        .assetId(assetId)
        .build();

    return getAsyncClient().deleteAsset(deleteAssetRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("An error occurred deleting asset with id: {}",
assetId);
            }
        });
}
```

- Pour plus de détails sur l'API, voir [DeleteAsset](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  DeleteAssetCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset.
 * @param {{ assetId : string }}
 */
export const main = async ({ assetId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetCommand({
        assetId: assetId, // The model id to delete.
      }),
    );
    console.log("Asset deleted successfully.");
    return { assetDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset.`
      );
    } else {
      throw caught;
    }
  }
}
```



```
}
};
```

- Pour plus de détails sur l'API, voir [DeleteAsset](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
            access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
```

```

iotsitewise_client = boto3.client("iotsitewise")
return cls(iotsitewise_client)

def delete_asset(self, asset_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Asset.

    :param asset_id: The ID of the asset to delete.
    """
    try:
        self.iotsitewise_client.delete_asset(assetId=asset_id)
    except ClientError as err:
        logger.error(
            "Error deleting asset %s. Here's why %s",
            asset_id,
            err.response["Error"]["Message"],
        )
        raise

```

- Pour plus de détails sur l'API, consultez [DeleteAsset](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteAssetModel** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DeleteAssetModel.

CLI

AWS CLI

Pour supprimer un modèle de ressource

L'`delete-asset-model` exemple suivant supprime un modèle d'actif d'éolienne.

```
aws iotsitewise delete-asset-model \
```

```
--asset-model-id a1b2c3d4-5678-90ab-cdef-11111EXAMPLE
```

Sortie :

```
{
  "assetModelStatus": {
    "state": "DELETING"
  }
}
```

Pour plus d'informations, consultez [la section Suppression de modèles d'actifs](#) dans le guide de SiteWise l'utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [DeleteAssetModel](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Deletes an Asset Model with the specified ID.
 *
 * @param assetModelId the ID of the Asset Model to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteAssetModelResponse} result. The calling code
 *       can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *       {@link CompletableFuture#get()}.
 *       <p>
 *       If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *       it available to the calling code as a {@link CompletionException}.
By calling
```

```

    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<DeleteAssetModelResponse>
deleteAssetModelAsync(String assetModelId) {
        DeleteAssetModelRequest deleteAssetModelRequest =
DeleteAssetModelRequest.builder()
            .assetModelId(assetModelId)
            .build();

        return getAsyncClient().deleteAssetModel(deleteAssetModelRequest)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("Failed to delete asset model with ID:{}",
exception.getMessage());
                }
            });
    }
}

```

- Pour plus de détails sur l'API, voir [DeleteAssetModel](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    DeleteAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Delete an asset model.

```

```

* @param {{ assetModelId : string }}
*/
export const main = async ({ assetModelId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteAssetModelCommand({
        assetModelId: assetModelId, // The model id to delete.
      }),
    );
    console.log("Asset model deleted successfully.");
    return { assetModelDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. There was a problem deleting the asset model.`
      );
    } else {
      throw caught;
    }
  }
};

```

- Pour plus de détails sur l'API, voir [DeleteAssetModel](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:

```

```

    """
    Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
                                access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def delete_asset_model(self, asset_model_id: str) -> None:
        """
        Deletes an AWS IoT SiteWise Asset Model.

        :param asset_model_id: The ID of the asset model to delete.
        """
        try:
            self.iotsitewise_client.delete_asset_model(assetModelId=asset_model_id)
        except ClientError as err:
            logger.error(
                "Error deleting asset model %s. Here's why %s",
                asset_model_id,
                err.response["Error"]["Message"],
            )
            raise

```

- Pour plus de détails sur l'API, consultez [DeleteAssetModelle](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeleteGateway** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DeleteGateway.

CLI

AWS CLI

Pour supprimer une passerelle

L'`delete-gateway`exemple suivant supprime une passerelle.

```
aws iotsitewise delete-gateway \  
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE
```

Cette commande ne produit aucun résultat.

Pour plus d'informations, consultez la section [Ingestion de données à l'aide d'une passerelle](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [DeleteGateway](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

/**
 * Deletes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to delete.
 * @return a {@link CompletableFuture} that represents a {@link
DeleteGatewayResponse} result.. The calling code
 *         can attach callbacks, then handle the result or exception by
calling {@link CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
public CompletableFuture<DeleteGatewayResponse> deleteGatewayAsync(String
gatewayId) {
    DeleteGatewayRequest deleteGatewayRequest =
DeleteGatewayRequest.builder()
        .gatewayId(gatewayId)
        .build();

    return getAsyncClient().deleteGateway(deleteGatewayRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete gateway: {}",
exception.getCause().getMessage());
            }
        });
}
}

```

- Pour plus de détails sur l'API, voir [DeleteGateway](#) la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  DeleteGatewayCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    await client.send(
      new DeleteGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway deleted successfully.");
    return { gatewayDeleted: true };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, voir [DeleteGateway](#) la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```
def delete_gateway(self, gateway_id: str) -> None:
    """
    Deletes an AWS IoT SiteWise Gateway.

    :param gateway_id: The ID of the gateway to delete.
    """
    try:
        self.iotsitewise_client.delete_gateway(gatewayId=gateway_id)
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceNotFoundException":
            logger.error("Gateway %s does not exist.", gateway_id)
        else:
            logger.error(
                "Error deleting gateway %s. Here's why %s",
                gateway_id,
                err.response["Error"]["Message"],
            )
        raise
```

- Pour plus de détails sur l'API, consultez [DeleteGateway](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DeletePortal** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DeletePortal.

CLI

AWS CLI

Pour supprimer un portail

L'`delete-portal` exemple suivant supprime le portail Web d'une société de parcs éoliens.

```
aws iotsitewise delete-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE
```

Sortie :

```
{
  "portalStatus": {
    "state": "DELETING"
  }
}
```

Pour plus d'informations, consultez [la section Suppression d'un portail](#) dans le guide de l'utilisateur de l'AWS IoT SiteWise.

- Pour plus de détails sur l'API, reportez-vous [DeletePortal](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Deletes a portal.
 *
 * @param portalId the ID of the portal to be deleted.
 * @return a {@link CompletableFuture} that represents a {@link
DeletePortalResponse}. The calling code can attach
 *         callbacks, then handle the result or exception by calling {@link
CompletableFuture#join()} or
 *         {@link CompletableFuture#get()}.
 *         <p>
 *         If any completion stage in this method throws an exception, the
method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
By calling
 *         {@link CompletionException#getCause()}, the calling code can
access the original exception.
 */
```

```

public CompletableFuture<DeletePortalResponse> deletePortalAsync(String
portalId) {
    DeletePortalRequest deletePortalRequest = DeletePortalRequest.builder()
        .portalId(portalId)
        .build();

    return getAsyncClient().deletePortal(deletePortalRequest)
        .whenComplete((response, exception) -> {
            if (exception != null) {
                logger.error("Failed to delete portal with ID: {}. Error:
{}", portalId, exception.getCause().getMessage());
            }
        });
}

```

- Pour plus de détails sur l'API, reportez-vous [DeletePortal](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    DeletePortalCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * List asset models.
 * @param {{ portalId : string }}
 */
export const main = async ({ portalId }) => {
    const client = new IoTSiteWiseClient({});

```

```

try {
  await client.send(
    new DeletePortalCommand({
      portalId: portalId, // The id of the portal.
    }),
  );
  console.log("Portal deleted successfully.");
  return { portalDeleted: true };
} catch (caught) {
  if (caught instanceof Error && caught.name === "ResourceNotFound") {
    console.warn(
      `${caught.message}. There was a problem deleting the portal. Please check
the portal id.`
    );
  } else {
    throw caught;
  }
}
};

```

- Pour plus de détails sur l'API, reportez-vous [DeletePortal](#) à la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

```

```

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.

        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def delete_portal(self, portal_id: str) -> None:
        """
        Deletes an AWS IoT SiteWise Portal.

        :param portal_id: The ID of the portal to delete.
        """
        try:
            self.iotsitewise_client.delete_portal(portalId=portal_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error("Portal %s does not exist.", portal_id)
            else:
                logger.error(
                    "Error deleting portal %s. Here's why %s",
                    portal_id,
                    err.response["Error"]["Message"],
                )
            raise

```

- Pour plus de détails sur l'API, consultez [DeletePortale](#) AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DescribeAssetModel** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribeAssetModel`.

CLI

AWS CLI

Pour décrire un modèle d'actifs

L'`describe-asset-model`exemple suivant décrit un modèle d'actifs de parc éolien.

```
aws iotsitewise describe-asset-model \
  --asset-model-id a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Sortie :

```
{
  "assetModelId": "a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelArn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
  "assetModelName": "Wind Farm Model",
  "assetModelDescription": "Represents a wind farm that comprises many wind turbines",
  "assetModelProperties": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-99999EXAMPLE",
      "name": "Total Generated Power",
      "dataType": "DOUBLE",
      "unit": "kW",
      "type": {
        "metric": {
          "expression": "sum(power)",
          "variables": [
            {
```



```

        "name": "power",
        "value": {
            "propertyId": "a1b2c3d4-5678-90ab-
cdef-66666EXAMPLE",
            "hierarchyId": "a1b2c3d4-5678-90ab-
cdef-77777EXAMPLE"
        }
    },
    "window": {
        "tumbling": {
            "interval": "1h"
        }
    }
},
{
    "id": "a1b2c3d4-5678-90ab-cdef-88888EXAMPLE",
    "name": "Region",
    "dataType": "STRING",
    "type": {
        "attribute": {
            "defaultValue": " "
        }
    }
},
],
"assetModelHierarchies": [
    {
        "id": "a1b2c3d4-5678-90ab-cdef-77777EXAMPLE",
        "name": "Wind Turbines",
        "childAssetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
    }
],
"assetModelCreationDate": 1575671284.0,
"assetModelLastUpdateDate": 1575671988.0,
"assetModelStatus": {
    "state": "ACTIVE"
}
}

```

Pour plus d'informations, consultez la section [Décrire un modèle d'actif spécifique](#) dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [DescribeAssetModel](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Retrieves the property IDs associated with a specific asset model.
 *
 * @param assetModelId the ID of the asset model that defines the properties.
 * @return a {@link CompletableFuture} that represents a {@link Map} result
 * that associates the property name to the
 *      *      property ID. The calling code can attach callbacks, then handle the
 * result or exception by calling
 *      *      {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.
 *      *      <p>
 *      *      If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *      *      it available to the calling code as a {@link CompletionException}.
 * By calling
 *      *      {@link CompletionException#getCause()}, the calling code can
 * access the original exception.
 */
public CompletableFuture<Map<String, String>> getPropertyIds(String
assetModelId) {
    ListAssetModelPropertiesRequest modelPropertiesRequest =
ListAssetModelPropertiesRequest.builder().assetModelId(assetModelId).build();
    return getAsyncClient().listAssetModelProperties(modelPropertiesRequest)
        .handle((response, throwable) -> {
            if (response != null) {
                return response.assetModelPropertySummaries().stream()
```

```

        .collect(Collectors
            .toMap(AssetModelPropertySummary::name,
AssetModelPropertySummary::id));
    } else {
        logger.error("Error occurred while fetching property IDs:
    {}. ", throwable.getCause().getMessage());
        throw (CompletionException) throwable;
    }
    });
}

```

- Pour plus de détails sur l'API, reportez-vous [DescribeAssetModel](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    DescribeAssetModelCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset model.
 * @param {{ assetModelId : string }}
 */
export const main = async ({ assetModelId }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const { assetModelDescription } = await client.send(
            new DescribeAssetModelCommand({
                assetModelId: assetModelId, // The ID of the Gateway to describe.
            })
        );
    } catch (error) {
        console.error(error);
    }
};

```

```

    }},
  );
  console.log("Asset model information retrieved successfully.");
  return { assetModelDescription: assetModelDescription };
} catch (caught) {
  if (caught instanceof Error && caught.name === "ResourceNotFound") {
    console.warn(
      `${caught.message}. The asset model could not be found. Please check the
      asset model id.`
    );
  } else {
    throw caught;
  }
}
};

```

- Pour plus de détails sur l'API, reportez-vous [DescribeAssetModel](#) à la section Référence des AWS SDK pour JavaScript API.

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DescribeGateway** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser DescribeGateway.

CLI

AWS CLI

Pour décrire une passerelle

L'`describe-gateway` exemple suivant décrit une passerelle.

```

aws iotsitewise describe-gateway \
  --gateway-id a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE

```

Sortie :

```
{
```

```

    "gatewayId": "a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayName": "ExampleCorpGateway",
    "gatewayArn": "arn:aws:iotsitewise:us-west-2:123456789012:gateway/a1b2c3d4-5678-90ab-cdef-1a1a1EXAMPLE",
    "gatewayPlatform": {
      "greengrass": {
        "groupArn": "arn:aws:greengrass:us-west-2:123456789012:/greengrass/groups/a1b2c3d4-5678-90ab-cdef-1b1b1EXAMPLE"
      }
    },
    "gatewayCapabilitySummaries": [
      {
        "capabilityNamespace": "iotsitewise:opcuacollector:1",
        "capabilitySyncStatus": "IN_SYNC"
      }
    ],
    "creationDate": 1588369971.457,
    "lastUpdateDate": 1588369971.457
  }

```

Pour plus d'informations, consultez la section [Ingestion de données à l'aide d'une passerelle](#) dans le guide de l' utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [DescribeGateway](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

/**
 * Describes the specified gateway.
 *
 * @param gatewayId the ID of the gateway to describe.
 * @return a {@link CompletableFuture} that represents a {@link
 * DescribeGatewayResponse} result. The calling code

```

```

    *      can attach callbacks, then handle the result or exception by
    calling {@link CompletableFuture#join()} or
    *      {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
    method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
    By calling
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<DescribeGatewayResponse> describeGatewayAsync(String
gatewayId) {
        DescribeGatewayRequest request = DescribeGatewayRequest.builder()
            .gatewayId(gatewayId)
            .build();

        return getAsyncClient().describeGateway(request)
            .whenComplete((response, exception) -> {
                if (exception != null) {
                    logger.error("An error occurred during the describeGateway
method: {}", exception.getCause().getMessage());
                }
            });
    }
}

```

- Pour plus de détails sur l'API, reportez-vous [DescribeGateway](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
```

```
DescribeGatewayCommand,
IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Create an SSM document.
 * @param {{ content: string, name: string, documentType?: DocumentType }}
 */
export const main = async ({ gatewayId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const { gatewayDescription } = await client.send(
      new DescribeGatewayCommand({
        gatewayId: gatewayId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Gateway information retrieved successfully.");
    return { gatewayDescription: gatewayDescription };
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Gateway could not be found. Please check the Gateway Id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, reportez-vous [DescribeGateway](#) à la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":
        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def describe_gateway(self, gateway_id: str) -> Dict[str, Any]:
        """
        Describes an AWS IoT SiteWise Gateway.
```



```

:param gateway_id: The ID of the gateway to describe.
:return: A dictionary containing information about the gateway.
"""
try:
    response =
self.iotsitewise_client.describe_gateway(gatewayId=gateway_id)
    return response
except ClientError as err:
    if err.response["Error"]["Code"] == "ResourceNotFoundException":
        logger.error("Gateway %s does not exist.", gateway_id)
    else:
        logger.error(
            "Error describing gateway %s. Here's why %s",
            gateway_id,
            err.response["Error"]["Message"],
        )
    raise

```

- Pour plus de détails sur l'API, consultez [DescribeGateway](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **DescribePortal** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `DescribePortal`.

CLI

AWS CLI

Pour décrire un portail

L'`describe-portal` exemple suivant décrit un portail Web pour une société de parc éolien.

```

aws iotsitewise describe-portal \
  --portal-id a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE

```

Sortie :

```
{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:us-west-2:123456789012:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalName": "WindFarmPortal",
  "portalDescription": "A portal that contains wind farm projects for Example Corp.",
  "portalClientId": "E-a1b2c3d4e5f6_a1b2c3d4e5f6EXAMPLE",
  "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
  "portalContactEmail": "support@example.com",
  "portalStatus": {
    "state": "ACTIVE"
  },
  "portalCreationDate": "2020-02-04T23:01:52.90248068Z",
  "portalLastUpdateDate": "2020-02-04T23:01:52.90248078Z",
  "roleArn": "arn:aws:iam::123456789012:role/MySiteWiseMonitorServiceRole"
}
```

Pour plus d'informations, consultez la section [Administration de vos portails](#) dans le Guide de l'utilisateur de l'AWS IoT SiteWise.

- Pour plus de détails sur l'API, reportez-vous [DescribePortal](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Retrieves a portal's description.
 *
 * @param portalId the ID of the portal to describe.
```

```

    * @return a {@link CompletableFuture} that represents a {@link String}
    result of the portal's start URL
    *      (see: {@link DescribePortalResponse#portalStartUrl()}). The
    calling code can attach callbacks, then handle the
    *      result or exception by calling {@link CompletableFuture#join()} or
    {@link CompletableFuture#get()}.
    *      <p>
    *      If any completion stage in this method throws an exception, the
    method logs the exception cause and keeps
    *      it available to the calling code as a {@link CompletionException}.
    By calling
    *      {@link CompletionException#getCause()}, the calling code can
    access the original exception.
    */
    public CompletableFuture<String> describePortalAsync(String portalId) {
        DescribePortalRequest request = DescribePortalRequest.builder()
            .portalId(portalId)
            .build();

        return getAsyncClient().describePortal(request)
            .handle((response, exception) -> {
                if (exception != null) {
                    logger.error("An exception occurred retrieving the portal
description: {}", exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                return response.portalStartUrl();
            });
    }
}

```

- Pour plus de détails sur l'API, reportez-vous [DescribePortal](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
import {
  DescribePortalCommand,
  IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe a portal.
 * @param {{ portalId: string }}
 */
export const main = async ({ portalId }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new DescribePortalCommand({
        portalId: portalId, // The ID of the Gateway to describe.
      }),
    );
    console.log("Portal information retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "ResourceNotFound") {
      console.warn(
        `${caught.message}. The Portal could not be found. Please check the Portal Id.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, reportez-vous [DescribePortal](#) à la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSitewiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)
```

```

def create_gateway(self, gateway_name: str, my_thing: str) -> str:
    """
    Creates an AWS IoT SiteWise Gateway.

    :param gateway_name: The name of the gateway to create.
    :param my_thing: The core device thing name.
    :return: The ID of the created gateway.
    """
    try:
        response = self.iotsitewise_client.create_gateway(
            gatewayName=gateway_name,
            gatewayPlatform={
                "greengrassV2": {"coreDeviceThingName": my_thing},
            },
            tags={"Environment": "Production"},
        )
        gateway_id = response["gatewayId"]
        return gateway_id
    except ClientError as err:
        if err.response["Error"]["Code"] == "ResourceAlreadyExistsException":
            logger.error("Gateway %s already exists.", gateway_name)
        else:
            logger.error(
                "Error creating gateway %s. Here's why %s",
                gateway_name,
                err.response["Error"]["Message"],
            )
        raise

```

- Pour plus de détails sur l'API, consultez [DescribePortale](#) AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **GetAssetPropertyValue** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser **GetAssetPropertyValue**.

CLI

AWS CLI

Pour récupérer la valeur actuelle d'une propriété d'actif

L'`get-asset-property-value` exemple suivant permet de récupérer la puissance totale actuelle d'une éolienne.

```
aws iotsitewise get-asset-property-value \
  --asset-id a1b2c3d4-5678-90ab-cdef-3333EXAMPLE \
  --property-id a1b2c3d4-5678-90ab-cdef-6666EXAMPLE
```

Sortie :

```
{
  "propertyValue": {
    "value": {
      "doubleValue": 6890.8677520453875
    },
    "timestamp": {
      "timeInSeconds": 1580853000,
      "offsetInNanos": 0
    },
    "quality": "GOOD"
  }
}
```

Pour plus d'informations, consultez la section [Interrogation des valeurs actuelles des propriétés des actifs](#) dans le Guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [GetAssetPropertyValue](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

 Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
/**
 * Fetches the value of an asset property.
 *
 * @param propId the ID of the asset property to fetch.
 * @param assetId the ID of the asset to fetch the property value for.
 * @return a {@link CompletableFuture} that represents a {@link Double}
 * result. The calling code can attach
 *     callbacks, then handle the result or exception by calling {@link
 * CompletableFuture#join()} or
 *     {@link CompletableFuture#get()}.
 *     <p>
 *     If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *     it available to the calling code as a {@link CompletionException}.
 * By calling
 *     {@link CompletionException#getCause()}, the calling code can
 * access the original exception.
 */
public CompletableFuture<Double> getAssetPropValueAsync(String propId, String
assetId) {
    GetAssetPropertyValueRequest assetPropertyValueRequest =
    GetAssetPropertyValueRequest.builder()
        .propertyId(propId)
        .assetId(assetId)
        .build();

    return getAsyncClient().getAssetPropertyValue(assetPropertyValueRequest)
        .handle((response, exception) -> {
            if (exception != null) {
                logger.error("Error occurred while fetching property
value: {}.", exception.getCause().getMessage());
                throw (CompletionException) exception;
            }
        });
}
```



```

        }
        return response.propertyValue().value().doubleValue();
    });
}

```

- Pour plus de détails sur l'API, reportez-vous [GetAssetPropertyValue](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    GetAssetPropertyValueCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

/**
 * Describe an asset property value.
 * @param {{ entryId : string }}
 */
export const main = async ({ entryId }) => {
    const client = new IoTSiteWiseClient({});
    try {
        const result = await client.send(
            new GetAssetPropertyValueCommand({
                entryId: entryId, // The ID of the Gateway to describe.
            }),
        );
        console.log("Asset property information retrieved successfully.");
        return result;
    } catch (caught) {
        if (caught instanceof Error && caught.name === "ResourceNotFound") {

```

```

        console.warn(
            `${caught.message}. The asset property entry could not be found. Please
            check the entry id.` ,
        );
    } else {
        throw caught;
    }
}
};

```

- Pour plus de détails sur l'API, reportez-vous [GetAssetPropertyValue](#) à la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

class IoTSiteWiseWrapper:
    """Encapsulates AWS IoT SiteWise actions using the client interface."""

    def __init__(self, iotsitewise_client: client) -> None:
        """
        Initializes the IoTSiteWiseWrapper with an AWS IoT SiteWise client.

        :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
        provides low-level
                                access to AWS IoT SiteWise services.
        """
        self.iotsitewise_client = iotsitewise_client
        self.entry_id = 0 # Incremented to generate unique entry IDs for
        batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSiteWiseWrapper":

```

```

        """
        Creates an IoTSiteWiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSiteWiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def get_asset_property_value(
        self, asset_id: str, property_id: str
    ) -> Dict[str, Any]:
        """
        Gets the value of an AWS IoT SiteWise Asset Property.

        :param asset_id: The ID of the asset.
        :param property_id: The ID of the property.
        :return: A dictionary containing the value of the property.
        """
        try:
            response = self.iotsitewise_client.get_asset_property_value(
                assetId=asset_id, propertyId=property_id
            )
            return response["propertyValue"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "ResourceNotFoundException":
                logger.error(
                    "Asset %s or property %s does not exist.", asset_id,
property_id
                )
            else:
                logger.error(
                    "Error getting asset property value. Here's why %s",
                    err.response["Error"]["Message"],
                )
            raise

```

- Pour plus de détails sur l'API, consultez [GetAssetPropertyValue](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Utilisation **ListAssetModels** avec un AWS SDK ou une CLI

Les exemples de code suivants illustrent comment utiliser `ListAssetModels`.

CLI

AWS CLI

Pour répertorier tous les modèles d'actifs

L'`list-asset-models`exemple suivant répertorie tous les modèles d'actifs définis dans votre AWS compte dans la région actuelle.

```
aws iotsitewise list-asset-models
```

Sortie :

```
{
  "assetModelSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
      "name": "Wind Farm Model",
      "description": "Represents a wind farm that comprises many wind turbines",
      "creationDate": 1575671284.0,
      "lastUpdateDate": 1575671988.0,
      "status": {
        "state": "ACTIVE"
      }
    },
    {
      "id": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "arn": "arn:aws:iotsitewise:us-west-2:123456789012:asset-model/a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
      "name": "Wind Turbine Model",
      "description": "Represents a wind turbine manufactured by Example Corp",

```

```

        "creationDate": 1575671207.0,
        "lastUpdateDate": 1575686273.0,
        "status": {
            "state": "ACTIVE"
        }
    }
]
}

```

Pour plus d'informations, consultez la section [Liste de tous les modèles d'actifs](#) dans le guide de l' SiteWise utilisateur de l'AWS IoT.

- Pour plus de détails sur l'API, reportez-vous [ListAssetModels](#) à la section Référence des AWS CLI commandes.

Java

SDK pour Java 2.x

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

/**
 * Retrieves the asset model ID for the given asset model name.
 *
 * @param assetModelName the name of the asset model for the ID.
 * @return a {@link CompletableFuture} that represents a {@link String}
 * result of the asset model ID or null if the
 *         asset model cannot be found. The calling code can attach
 * callbacks, then handle the result or exception
 *         by calling {@link CompletableFuture#join()} or {@link
 * CompletableFuture#get()}.
 *
 * <p>
 *         If any completion stage in this method throws an exception, the
 * method logs the exception cause and keeps
 *         it available to the calling code as a {@link CompletionException}.
 * By calling
 *         {@link CompletionException#getCause()}, the calling code can
 * access the original exception.

```

```

    */
    public CompletableFuture<String> getAssetModelIdAsync(String assetModelName)
    {
        ListAssetModelsRequest listAssetModelsRequest =
        ListAssetModelsRequest.builder().build();
        return getAsyncClient().listAssetModels(listAssetModelsRequest)
            .handle((listAssetModelsResponse, exception) -> {
                if (exception != null) {
                    logger.error("Failed to retrieve Asset Model ID: {}",
                    exception.getCause().getMessage());
                    throw (CompletionException) exception;
                }
                for (AssetModelSummary assetModelSummary :
                listAssetModelsResponse.assetModelSummaries()) {
                    if (assetModelSummary.name().equals(assetModelName)) {
                        return assetModelSummary.id();
                    }
                }
                return null;
            });
    }
}

```

- Pour plus de détails sur l'API, reportez-vous [ListAssetModels](#) à la section Référence des AWS SDK for Java 2.x API.

JavaScript

SDK pour JavaScript (v3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```

import {
    ListAssetModelsCommand,
    IoTSiteWiseClient,
} from "@aws-sdk/client-iotsitewise";
import { parseArgs } from "node:util";

```

```
/**
 * List asset models.
 * @param {{ assetModelTypes : array }}
 */
export const main = async ({ assetModelTypes = [] }) => {
  const client = new IoTSiteWiseClient({});
  try {
    const result = await client.send(
      new ListAssetModelsCommand({
        assetModelTypes: assetModelTypes, // The model types to list
      }),
    );
    console.log("Asset model types retrieved successfully.");
    return result;
  } catch (caught) {
    if (caught instanceof Error && caught.name === "IoTSiteWiseError") {
      console.warn(
        `${caught.message}. There was a problem listing the asset model types.`,
      );
    } else {
      throw caught;
    }
  }
};
```

- Pour plus de détails sur l'API, reportez-vous [ListAssetModels](#) à la section Référence des AWS SDK pour JavaScript API.

Python

SDK pour Python (Boto3)

Note

Il y en a plus à ce sujet GitHub. Trouvez l'exemple complet et découvrez comment le configurer et l'exécuter dans le [référentiel d'exemples de code AWS](#).

```
class IoTSiteWiseWrapper:
```

```

"""Encapsulates AWS IoT SiteWise actions using the client interface."""

def __init__(self, iotsitewise_client: client) -> None:
    """
    Initializes the IoTSitewiseWrapper with an AWS IoT SiteWise client.

    :param iotsitewise_client: A Boto3 AWS IoT SiteWise client. This client
    provides low-level
                                access to AWS IoT SiteWise services.
    """
    self.iotsitewise_client = iotsitewise_client
    self.entry_id = 0 # Incremented to generate unique entry IDs for
    batch_put_asset_property_value.

    @classmethod
    def from_client(cls) -> "IoTSitewiseWrapper":
        """
        Creates an IoTSitewiseWrapper instance with a default AWS IoT SiteWise
        client.

        :return: An instance of IoTSitewiseWrapper initialized with the default
        AWS IoT SiteWise client.
        """
        iotsitewise_client = boto3.client("iotsitewise")
        return cls(iotsitewise_client)

    def list_asset_models(self) -> List[Dict[str, Any]]:
        """
        Lists all AWS IoT SiteWise Asset Models.

        :return: A list of dictionaries containing information about each asset
        model.

        """
        try:
            asset_models = []
            paginator =
self.iotsitewise_client.get_paginator("list_asset_models")
            pages = paginator.paginate()
            for page in pages:
                asset_models.extend(page["assetModelSummaries"])
            return asset_models
        except ClientError as err:

```



```
logger.error(  
    "Error listing asset models. Here's why %s",  
    err.response["Error"]["Message"],  
)  
raise
```

- Pour plus de détails sur l'API, consultez [ListAssetModels](#) le AWS manuel de référence de l'API SDK for Python (Boto3).

Pour obtenir la liste complète des guides de développement du AWS SDK et des exemples de code, consultez [Utilisation de ce service avec un AWS SDK](#). Cette rubrique comprend également des informations sur le démarrage et sur les versions précédentes de SDK.

Sécurité dans AWS IoT SiteWise

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture 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 cela 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 AWS cadre des [programmes](#) de de). Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS IoT SiteWise, consultez la section [AWS services concernés par programme de conformité](#) et .
- 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 SiteWise. Les rubriques suivantes expliquent comment procéder à la configuration AWS IoT SiteWise 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 SiteWise ressources.

Rubriques

- [Protection des données dans AWS IoT SiteWise](#)
- [Chiffrement des données dans AWS IoT SiteWise](#)
- [Gestion des identités et des accès pour AWS IoT SiteWise](#)
- [Validation de conformité pour AWS IoT SiteWise](#)
- [Résilience dans AWS IoT SiteWise](#)
- [Sécurité de l'infrastructure dans AWS IoT SiteWise](#)
- [Analyse de configuration et de vulnérabilité dans AWS IoT SiteWise](#)
- [Points de terminaison VPC pour AWS IoT SiteWise](#)
- [Bonnes pratiques de sécurité pour AWS IoT SiteWise](#)

Protection des données dans AWS IoT SiteWise

Le [modèle de responsabilité AWS partagée](#) de s'applique à la protection des données dans AWS IoT SiteWise. 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 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 AWS Identity and Access Management (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 SiteWise 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.

Rubriques

- [Confidentialité du trafic interrégion pour AWS IoT SiteWise](#)
- [AWS IoT SiteWise Amélioration du service commercial adjoint](#)

Confidentialité du trafic interrégion pour AWS IoT SiteWise

Les connexions entre les applications locales AWS IoT SiteWise et entre elles, telles que les passerelles SiteWise Edge, sont sécurisées par le biais de connexions TLS (Transport Layer Security). Pour de plus amples informations, veuillez consulter [Chiffrement des données en transit pour AWS IoT SiteWise](#).

AWS IoT SiteWise ne prend pas en charge les connexions entre les zones de disponibilité d'une AWS région ni les connexions entre AWS comptes.

Vous ne pouvez configurer IAM Identity Center que dans une seule région à la fois. SiteWise Monitor se connecte à la région que vous avez configurée pour IAM Identity Center. Cela signifie que vous utilisez une région pour accéder à l'IAM Identity Center, mais que vous pouvez créer des portails dans n'importe quelle région.

AWS IoT SiteWise Amélioration du service commercial adjoint

AWS IoT SiteWise L'Assistant n'utilise pas les données des clients pour améliorer le service ou le sous-jacent LLMs.

Chiffrement des données dans AWS IoT SiteWise

Le chiffrement des données fait référence à la protection des données en transit (lorsqu'elles voyagent vers et depuis AWS IoT SiteWise, et entre les passerelles SiteWise Edge et les serveurs) et au repos (lorsqu'elles sont stockées sur des appareils locaux ou dans AWS des services). Vous pouvez protéger les données en transit à l'aide du protocole TLS (Transport Layer Security) ou au repos à l'aide du chiffrement côté client.

 Note

AWS IoT SiteWise les expositions de traitement Edge APIs hébergées sur des passerelles SiteWise Edge et accessibles via le réseau local. Ils APIs sont exposés via une connexion TLS soutenue par un certificat de serveur appartenant au AWS IoT SiteWise connecteur Edge. Pour l'authentification du client, ceux-ci APIs utilisent un mot de passe de contrôle d'accès. La clé privée du certificat de serveur et le mot de passe de contrôle d'accès sont tous deux stockés sur disque. AWS IoT SiteWise le traitement périphérique repose sur le chiffrement du système de fichiers pour la sécurité de ces informations d'identification au repos.

Pour plus d'informations sur le chiffrement côté serveur et le chiffrement côté client, consultez les rubriques ci-dessous.

Rubriques

- [Chiffrement au repos AWS IoT SiteWise](#)
- [Chiffrement des données en transit pour AWS IoT SiteWise](#)
- [Gestion des clés dans AWS IoT SiteWise](#)

Chiffrement au repos AWS IoT SiteWise

AWS IoT SiteWise stocke vos données dans le AWS cloud et sur des passerelles AWS IoT SiteWise Edge.

Données au repos dans le AWS cloud

AWS IoT SiteWise stocke les données dans d'autres AWS services qui chiffrent les données au repos par défaut. Encryption at rest s'intègre à AWS Key Management Service (AWS KMS) pour gérer la clé de chiffrement utilisée pour chiffrer les valeurs des propriétés de vos actifs et les valeurs agrégées dans AWS IoT SiteWise. Vous pouvez choisir d'utiliser une clé gérée par le client pour chiffrer les valeurs des propriétés des actifs et les valeurs agrégées dans AWS IoT SiteWise. Vous pouvez créer, gérer et consulter votre clé de chiffrement via AWS KMS.

Vous pouvez choisir une clé Clé détenue par AWS pour chiffrer vos données ou choisir une clé gérée par le client pour chiffrer les valeurs des propriétés et les valeurs agrégées de vos actifs :

Comment ça marche

Le chiffrement au repos s'intègre AWS KMS à la gestion de la clé de chiffrement utilisée pour chiffrer vos données.

- Clé détenue par AWS — Clé de chiffrement par défaut. AWS IoT SiteWise possède cette clé. Vous ne pouvez pas voir cette clé dans votre AWS compte. Vous ne pouvez pas non plus voir les opérations sur la clé dans AWS CloudTrail les journaux. Vous pouvez utiliser cette clé sans frais supplémentaires.
- Clé gérée par le client — La clé est stockée dans votre compte, que vous créez, détenez et gérez. Vous avez le contrôle total de la clé KMS. Des AWS KMS frais supplémentaires s'appliquent.

Clés détenues par AWS

Clés détenues par AWS ne sont pas enregistrés dans votre compte. Elles font partie d'un ensemble de clés KMS que AWS possède et gère pour une utilisation dans plusieurs AWS comptes. AWS services que vous pouvez Clés détenues par AWS utiliser pour protéger vos données.

Vous ne pouvez pas afficher, gérer Clés détenues par AWS, utiliser ou auditer leur utilisation. Cependant, vous n'avez pas besoin de travailler ou de modifier de programme pour protéger les clés qui chiffrent vos données.

Aucuns frais mensuels ni frais d'utilisation ne vous sont facturés si vous en utilisez Clés détenues par AWS, et ils ne sont pas pris en compte dans les AWS KMS quotas de votre compte.

Clés gérées par le client

Les clés gérées par le client sont des clés KMS de votre compte que vous créez, possédez et gérez. Vous avez le contrôle total de ces clés KMS, telles que les suivantes :

- Établir et maintenir leurs politiques clés, leurs politiques IAM et leurs subventions
- Les activer et les désactiver
- Rotation de leur matériel cryptographique
- Ajout de balises
- Création d'alias qui y font référence
- Planifier leur suppression

Vous pouvez également utiliser CloudTrail Amazon CloudWatch Logs pour suivre les demandes AWS IoT SiteWise envoyées AWS KMS en votre nom.

Si vous utilisez des clés gérées par le client, vous devez autoriser l' AWS IoT SiteWise accès à la clé KMS enregistrée dans votre compte. AWS IoT SiteWise utilise le chiffrement des enveloppes et la hiérarchie des clés pour chiffrer les données. Votre clé de AWS KMS chiffrement est utilisée pour chiffrer la clé racine de cette hiérarchie de clés. Pour plus d'informations, consultez [Chiffrement d'enveloppe](#) dans le Guide du développeur AWS Key Management Service .

L'exemple de politique suivant accorde des AWS IoT SiteWise autorisations pour créer une clé gérée par le client en votre nom. Lorsque vous créez votre clé, vous devez autoriser les `kms:DescribeKey` actions `kms:CreateGrant` et.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:CreateGrant",
        "kms:DescribeKey"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Le contexte de chiffrement de la subvention que vous avez créée utilise votre identifiant de compte `aws:iotsitewise:subscriberId` et.

Données inactives sur les passerelles SiteWise Edge

AWS IoT SiteWise les passerelles stockent les données suivantes sur le système de fichiers local :

- Informations de configuration de la source OPC UA
- L'ensemble des chemins de flux de données OPC UA provenant de sources OPC UA connectées
- Données industrielles mises en cache lorsque la passerelle SiteWise Edge perd la connexion à Internet

SiteWise Les passerelles Edge fonctionnent. AWS IoT Greengrass s'appuie sur les autorisations de fichiers Unix et le chiffrement complet du disque (s'il est activé) pour protéger les données au repos sur le cœur. Il est de votre responsabilité de sécuriser le système de fichiers et l'appareil.

Toutefois, AWS IoT Greengrass copie les copies locales des secrets de votre serveur OPC UA récupérés depuis Secrets Manager. Pour plus d'informations, consultez la section [Chiffrement des secrets](#) dans le guide du AWS IoT Greengrass Version 1 développeur.

Pour plus d'informations sur le chiffrement au repos sur les AWS IoT Greengrass cœurs, consultez la section [Chiffrement au repos](#) dans le manuel du AWS IoT Greengrass Version 1 développeur.

Chiffrement des données en transit pour AWS IoT SiteWise

AWS IoT SiteWise utilise le chiffrement en transit pour sécuriser les données transmises entre vos appareils, vos passerelles et le AWS Cloud. Les communications avec AWS IoT SiteWise sont cryptées à l'aide des protocoles HTTPS et TLS 1.2, ce qui garantit la confidentialité de vos données et leur protection contre tout accès non autorisé ou toute interception.

Les données transitent par trois modes de communication :

- [Sur Internet : les](#) communications entre les appareils locaux (y compris les passerelles SiteWise Edge) AWS IoT SiteWise sont cryptées.
- [Sur le réseau local](#) : les communications entre SiteWise l'application et OpsHub les passerelles SiteWise Edge sont toujours cryptées. La communication entre l'application de SiteWise surveillance exécutée dans votre navigateur et les passerelles SiteWise Edge est toujours cryptée. Les communications entre les passerelles SiteWise Edge et les sources OPC UA peuvent être cryptées.
- [Entre les composants des passerelles SiteWise Edge](#) : les communications entre les AWS IoT Greengrass composants des passerelles SiteWise Edge ne sont pas chiffrées.

Rubriques

- [Données en transit sur Internet](#)
- [Données en transit sur le réseau local](#)
- [Données en transit entre les composants locaux sur SiteWise Edge](#)

Données en transit sur Internet

AWS IoT SiteWise utilise le protocole TLS (Transport Layer Security) pour chiffrer toutes les communications sur Internet. Toutes les données envoyées au AWS Cloud sont envoyées via une connexion TLS à l'aide des protocoles MQTT ou HTTPS. Elles sont donc sécurisées par défaut. SiteWise Les passerelles Edge, qui s'exécutent sur AWS IoT Greengrass, et les notifications relatives à la valeur des propriétés utilisent le modèle de sécurité du AWS IoT transport. Pour de plus amples informations, veuillez consulter [Sécurité du transport](#) dans le Manuel du développeur AWS IoT .

Données en transit sur le réseau local

SiteWise Les passerelles Edge suivent les spécifications OPC UA pour la communication avec les sources OPC UA locales. Il est de votre responsabilité de configurer vos sources pour utiliser un mode de sécurité des messages qui chiffre les données en transit.

Si vous choisissez un mode de sécurité des messages de signature, les données en transit entre les passerelles SiteWise Edge et les sources sont signées mais pas chiffrées. Si vous choisissez un mode de sécurité des messages par signature et chiffrement, les données en transit entre les passerelles SiteWise Edge et les sources sont signées et chiffrées. Pour de plus amples informations sur les sources de configuration, veuillez consulter [Ajoutez des sources de données à votre passerelle AWS IoT SiteWise Edge](#).

La communication entre l'application de console Edge et les passerelles SiteWise Edge est toujours cryptée par TLS. Le connecteur SiteWise Edge de la passerelle SiteWise Edge génère et stocke un certificat auto-signé afin de pouvoir établir une connexion TLS avec la console Edge pour AWS IoT SiteWise l'application. Vous devez copier ce certificat de votre passerelle SiteWise Edge vers la console Edge pour AWS IoT SiteWise l'application avant de connecter l'application à la passerelle SiteWise Edge. Cela garantit que la console Edge pour AWS IoT SiteWise l'application est en mesure de vérifier qu'elle est connectée à votre passerelle SiteWise Edge approuvée.

Outre le protocole TLS pour garantir la confidentialité et l'authenticité du serveur, SiteWise Edge utilise le protocole SigV4 pour établir l'authenticité de l'application de console Edge. Le connecteur SiteWise Edge de la passerelle SiteWise Edge accepte et stocke un mot de passe afin de vérifier les connexions entrantes provenant de l'application de console Edge, de l'application de SiteWise surveillance exécutée dans les navigateurs et d'autres clients basés sur le AWS IoT SiteWise SDK.

Pour plus d'informations sur la génération du mot de passe et du certificat de serveur, consultez [the section called "Gérez les passerelles"](#).

Données en transit entre les composants locaux sur SiteWise Edge

SiteWise Les passerelles Edge s'exécutent AWS IoT Greengrass, ce qui ne chiffre pas les données échangées localement sur le AWS IoT Greengrass cœur, car elles ne quittent pas l'appareil. Cela inclut la communication entre AWS IoT Greengrass les composants tels que le AWS IoT SiteWise connecteur. Pour plus d'informations, consultez la section [Données relatives à l'appareil principal](#) dans le Guide du AWS IoT Greengrass Version 1 développeur.

Gestion des clés dans AWS IoT SiteWise

AWS IoT SiteWise gestion des clés dans le cloud

Par défaut, AWS IoT SiteWise utilise Clés gérées par AWS pour protéger vos données dans le AWS cloud. Vous pouvez mettre à jour vos paramètres afin d'utiliser une clé gérée par le client pour chiffrer certaines données. AWS IoT SiteWise Vous pouvez créer, gérer et consulter votre clé de chiffrement via AWS Key Management Service (AWS KMS).

AWS IoT SiteWise prend en charge le chiffrement côté serveur en stockant les clés gérées par le client AWS KMS pour chiffrer les données suivantes :

- Valeurs des propriétés des actifs
- Valeurs agrégées

Note

Les autres données et ressources sont cryptées à l'aide du chiffrement par défaut avec des clés gérées par AWS IoT SiteWise. Cette clé est enregistrée dans le AWS IoT SiteWise compte.

Pour plus d'informations, voir [Qu'est-ce que c'est AWS Key Management Service ?](#) dans le Guide AWS Key Management Service du développeur.

Activez le chiffrement à l'aide de clés gérées par le client

Pour utiliser des clés gérées par le client avec AWS IoT SiteWise, vous devez mettre à jour vos AWS IoT SiteWise paramètres.

Pour activer le chiffrement à l'aide de clés KMS

1. Accédez à la [console AWS IoT SiteWise](#).
2. Choisissez Paramètres du compte, puis Modifier pour ouvrir la page Modifier les paramètres du compte.
3. Pour le type de clé de chiffrement, choisissez Choisir une autre AWS KMS clé. Cela permet le chiffrement avec les clés gérées par le client stockées dans AWS KMS.

Note

Actuellement, vous ne pouvez utiliser le chiffrement par clé géré par le client que pour les valeurs des propriétés des actifs et les valeurs agrégées.

4. Choisissez votre clé KMS avec l'une des options suivantes :
 - Pour utiliser une clé KMS existante : choisissez l'alias de votre clé KMS dans la liste.
 - Pour créer une nouvelle clé KMS, choisissez Créer une AWS KMS clé.

Note

Cela ouvre le tableau de bord AWS KMS . Pour plus d'informations sur la création d'une clé KMS, consultez la section [Création de clés](#) dans le Guide du AWS Key Management Service développeur.

5. Choisissez Enregistrer pour mettre à jour vos paramètres.

SiteWise Gestion des clés de la passerelle Edge

SiteWise Les passerelles Edge fonctionnent AWS IoT Greengrass, et les appareils AWS IoT Greengrass principaux utilisent des clés publiques et privées pour s'authentifier auprès du AWS cloud et chiffrer les secrets locaux, tels que les secrets d'authentification OPC UA. Pour plus d'informations, consultez la section [Gestion des clés](#) dans le Guide du AWS IoT Greengrass Version 1 développeur.

Gestion des identités et des accès pour AWS IoT SiteWise

AWS Identity and Access Management (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 SiteWise les ressources. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Audience dédiée à AWS IoT SiteWise la sécurité](#)
- [Authentifiez-vous avec des identités dans AWS IoT SiteWise](#)
- [Comment AWS IoT SiteWise fonctionne avec IAM](#)
- [AWS politiques gérées pour AWS IoT SiteWise](#)
- [Utilisez des rôles liés à un service pour AWS IoT SiteWise](#)
- [Configurez les autorisations pour les alarmes liées aux événements dans AWS IoT SiteWise](#)
- [Prévention interservices confuse des adjoints dans AWS IoT SiteWise](#)
- [Résoudre les problèmes d' AWS IoT SiteWise identité et d'accès](#)

Audience dédiée à AWS IoT SiteWise la sécurité

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez. AWS IoT SiteWise

Utilisateur du service : si vous utilisez le AWS IoT SiteWise service pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de nouvelles AWS IoT SiteWise fonctionnalités pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS IoT SiteWise, consultez [Résoudre les problèmes d' AWS IoT SiteWise identité et d'accès](#).

Administrateur du service — Si vous êtes responsable des AWS IoT SiteWise ressources de votre entreprise, vous avez probablement un accès complet à AWS IoT SiteWise. C'est à vous de déterminer les AWS IoT SiteWise fonctionnalités et les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM avec AWS IoT SiteWise, voir [Comment AWS IoT SiteWise fonctionne avec IAM](#).

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaitez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à AWS IoT SiteWise. Pour consulter des exemples de politiques AWS IoT SiteWise basées sur l'identité que vous pouvez utiliser dans IAM, consultez [AWS IoT SiteWise exemples de politiques basées sur l'identité](#)

Authentifiez-vous avec des identités dans AWS IoT SiteWise

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section [Comment vous connecter à votre compte Compte AWS dans](#) le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez [AWS Signature Version 4 pour les demandes d'API](#) dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez [Authentification multifactorielle](#) dans le Guide de l'utilisateur AWS IAM Identity Center et [Authentification multifactorielle AWS dans IAM](#) dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez [Tâches nécessitant les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. 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 au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez [passer d'un rôle d'utilisateur à un rôle IAM \(console\)](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Méthodes pour endosser un rôle](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- **Accès utilisateur fédéré** : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center .
- **Autorisations d'utilisateur IAM temporaires** : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- **Accès intercompte** : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez [Accès intercompte aux ressources dans IAM](#) dans le Guide de l'utilisateur IAM.
- **Accès multiservices** — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
 - **Sessions d'accès direct (FAS)** : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains

services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez [Transmission des sessions d'accès](#).

- Rôle de service : il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser 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.
- Rôle lié à un service — 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.
- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon](#) dans le guide de l'utilisateur IAM.

Comment AWS IoT SiteWise fonctionne avec IAM

Avant d'utiliser AWS Identity and Access Management (IAM) pour gérer l'accès à AWS IoT SiteWise, vous devez connaître les fonctionnalités IAM disponibles. AWS IoT SiteWise

Fonctionnalité IAM	Souten par AWS IoT SiteWis
Stratégies basées sur l'identité avec autorisations au niveau des ressources	Oui
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
Politiques basées sur les ressources	Non
Listes de contrôle d'accès (ACLs)	Non
Autorisation basée sur des balises (ABAC)	Oui
Informations d'identification temporaires	Oui
Sessions d'accès transféré (FAS)	Oui
Rôles liés à un service	Oui
Rôles de service	Oui

Pour obtenir une vue d'ensemble de la façon dont AWS IoT SiteWise les autres AWS services fonctionnent avec IAM, consultez la section [AWS Services compatibles avec IAM](#) dans le Guide de l'utilisateur d'IAM.

Table des matières

- [AWS IoT SiteWise Rôles IAM](#)
 - [Utilisez des informations d'identification temporaires avec AWS IoT SiteWise](#)
 - [Sessions d'accès direct \(FAS\) pour AWS IoT SiteWise](#)
 - [Rôles liés à un service](#)
 - [Rôles de service](#)

- [Choisissez un rôle IAM dans AWS IoT SiteWise](#)
- [Autorisation basée sur les balises AWS IoT SiteWise](#)
- [AWS IoT SiteWise politiques basées sur l'identité](#)
 - [Actions de politique](#)
 - [BatchPutAssetPropertyValue autorisation](#)
 - [Ressources de politique](#)
 - [Clés de condition de politique](#)
 - [Exemples](#)
- [AWS IoT SiteWise exemples de politiques basées sur l'identité](#)
 - [Bonnes pratiques en matière de politiques](#)
 - [Utiliser la AWS IoT SiteWise console](#)
 - [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)
 - [Permettre aux utilisateurs d'ingérer des données dans des actifs d'une seule hiérarchie](#)
 - [Afficher les AWS IoT SiteWise ressources en fonction des balises](#)
- [Gérez l'accès à l'aide de politiques dans AWS IoT SiteWise](#)
 - [Politiques basées sur l'identité](#)
 - [Politiques basées sur les ressources](#)
 - [Listes de contrôle d'accès \(ACLs\)](#)
 - [Autres types de politique](#)
 - [Plusieurs types de politique](#)

AWS IoT SiteWise Rôles IAM

Un [rôle IAM](#) est une entité au sein de votre compte AWS qui dispose d'autorisations spécifiques.

Utilisez des informations d'identification temporaires avec AWS IoT SiteWise

Vous pouvez utiliser des informations d'identification temporaires pour vous connecter à l'aide de la fédération, endosser un rôle IAM ou encore pour endosser un rôle intercompte. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d' AWS STS API telles que [AssumeRole](#) ou [GetFederationToken](#).

[AWS IoT SiteWise prend en charge l'utilisation d'informations d'identification temporaires.](#)

SiteWise Monitor aide les utilisateurs fédérés à accéder aux portails. Les utilisateurs du portail s'authentifient à l'aide de leurs informations d'identification IAM Identity Center ou IAM.

 Important

Les utilisateurs ou les rôles doivent être `iotsitewise:DescribePortal` autorisés à se connecter au portail.

Lorsqu'un utilisateur se connecte à un portail, SiteWise Monitor génère une politique de session qui fournit les autorisations suivantes :

- Accès en lecture seule aux actifs et aux données des actifs de AWS IoT SiteWise votre compte auxquels le rôle de ce portail permet d'accéder.
- Accès aux projets de ce portail auxquels l'utilisateur dispose d'un accès administrateur (propriétaire du projet) ou en lecture seule (visualiseur de projet).

Pour de plus amples informations sur les autorisations utilisateur du portail fédéré, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).

Sessions d'accès direct (FAS) pour AWS IoT SiteWise

Prend en charge les sessions d'accès direct (FAS) : oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez [Transmission des sessions d'accès](#).

Rôles liés à un service

[Les rôles liés à un service](#) permettent aux AWS services d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre AWS

compte et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

AWS IoT SiteWise prend en charge les rôles liés aux services. Pour plus d'informations sur la création ou la gestion des rôles liés à un service AWS IoT SiteWise, consultez [Utilisez des rôles liés à un service pour AWS IoT SiteWise](#).

Rôles de service

Cette fonction permet à un service d'endosser une [fonction du service](#) en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les rôles de service apparaissent dans votre AWS compte et sont détenus par le compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

AWS IoT SiteWise utilise un rôle de service pour permettre aux utilisateurs du portail SiteWise Monitor d'accéder à certaines de vos AWS IoT SiteWise ressources en votre nom. Pour de plus amples informations, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).

Vous devez disposer des autorisations requises avant de pouvoir créer des modèles AWS IoT Events d'alarme dans AWS IoT SiteWise. Pour de plus amples informations, veuillez consulter [Configurez les autorisations pour les alarmes liées aux événements dans AWS IoT SiteWise](#).

Choisissez un rôle IAM dans AWS IoT SiteWise

Lorsque vous créez une portail ressource dans AWS IoT SiteWise, vous devez choisir un rôle pour permettre aux utilisateurs fédérés de votre portail SiteWise Monitor d'y accéder en votre AWS IoT SiteWise nom. Si vous avez déjà créé un rôle de service, il vous AWS IoT SiteWise fournit une liste de rôles parmi lesquels choisir. Sinon, vous pouvez créer un rôle avec les autorisations requises lorsque vous créez un portail. Il est important de choisir un rôle qui permet d'accéder à vos actifs et à vos données d'actifs. Pour de plus amples informations, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).

Autorisation basée sur les balises AWS IoT SiteWise

Vous pouvez associer des balises aux AWS IoT SiteWise ressources ou transmettre des balises dans une demande à AWS IoT SiteWise. 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`. Pour

plus d'informations sur le balisage des ressources AWS IoT SiteWise , consultez [Marquez vos AWS IoT SiteWise ressources](#).

Pour visualiser un exemple de politique basée sur l'identité permettant de limiter l'accès à une ressource en fonction des balises de cette ressource, consultez [Afficher les AWS IoT SiteWise ressources en fonction des balises](#).

AWS IoT SiteWise politiques basées sur l'identité

Les politiques IAM vous permettent de contrôler qui peut faire quoi. AWS IoT SiteWise Vous pouvez décider quelles actions sont autorisées ou non et définir des conditions spécifiques pour ces actions. Par exemple, vous pouvez définir des règles concernant les personnes autorisées à consulter ou à modifier des informations dans AWS IoT SiteWise. AWS IoT SiteWise prend en charge des actions, des ressources et des clés de condition spécifiques. Pour en savoir plus sur 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.

Actions de politique

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. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions de politique en AWS IoT SiteWise cours utilisent le préfixe suivant avant l'action : `iotsitewise:`. Par exemple, pour autoriser une personne à télécharger les données relatives AWS IoT SiteWise aux propriétés des actifs dans le cadre de l'opération d'BatchPutAssetPropertyValueAPI, vous devez inclure l'`iotsitewise:BatchPutAssetPropertyValue` action dans sa politique. Les déclarations de politique doivent inclure un NotAction élément Action ou. AWS IoT SiteWise définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule instruction, séparez-les par des virgules, comme suit :

```
"Action": [  
  "iotsitewise:action1",  
  "iotsitewise:action2"  
]
```

Vous pouvez aussi préciser plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions qui commencent par le mot `Describe`, incluez l'action suivante.

```
"Action": "iotsitewise:Describe*"
```

Pour consulter la liste des AWS IoT SiteWise actions, reportez-vous à la section [Actions définies par AWS IoT SiteWise](#) dans le guide de l'utilisateur IAM.

BatchPutAssetPropertyValue autorisation

AWS IoT SiteWise autorise l'accès à l'[BatchPutAssetPropertyValue](#) action de manière inhabituelle. Pour la plupart des actions, lorsque vous autorisez ou refusez l'accès, cette action renvoie une erreur si les autorisations ne sont pas accordées. Avec `BatchPutAssetPropertyValue`, vous pouvez envoyer plusieurs entrées de données à différents actifs et propriétés d'actifs dans une seule demande d'API. AWS IoT SiteWise autorise chaque saisie de données de manière indépendante. Pour toute entrée individuelle dont l'autorisation échoue dans la demande, AWS IoT SiteWise inclut une erreur `AccessDeniedException` dans la liste d'erreurs renvoyée. AWS IoT SiteWise reçoit les données pour toute entrée autorisée et réussie, même si une autre entrée dans la même demande échoue.

Important

Avant d'ingérer des données dans un flux de données, procédez comme suit :

- Autorisez la `time-series` ressource si vous utilisez un alias de propriété pour identifier le flux de données.
- Autorisez la `asset` ressource si vous utilisez un ID d'actif pour identifier l'actif qui contient la propriété d'actif associée.

Ressources de politique

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. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son [Amazon Resource Name \(ARN\)](#). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*" 
```

Chaque déclaration de politique IAM s'applique aux ressources que vous spécifiez à l'aide de leur ARNs. Un ARN a la syntaxe générale suivante :

```
arn:${Partition}:${Service}:${Region}:${Account}:${ResourceType}/${ResourcePath}
```

Pour plus d'informations sur le format de ARNs, consultez [Identifier les AWS ressources avec Amazon Resource Names \(ARNs\)](#).

Par exemple, pour spécifier l'actif avec l'ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` dans votre instruction, utilisez l'ARN suivant :

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE" 
```

Pour spécifier tous les flux de données appartenant à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:iotsitewise:region:123456789012:time-series/*" 
```

Pour spécifier tous les actifs appartenant à un compte spécifique, utilisez le caractère générique (*) :

```
"Resource": "arn:aws:iotsitewise:region:123456789012:asset/*"
```

Certaines AWS IoT SiteWise actions, telles que celles relatives à la création de ressources, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

```
"Resource": ""
```

Pour spécifier plusieurs ressources dans une seule instruction, séparez-les ARNs par des virgules.

```
"Resource": [  
  "resource1",  
  "resource2"  
]
```

Pour consulter la liste des types de AWS IoT SiteWise ressources et leurs caractéristiques ARNs, consultez la section [Types de ressources définis par AWS IoT SiteWise](#) dans le guide de l'utilisateur IAM. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par AWS IoT SiteWise](#).

Clés de condition de politique

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 (ou le bloc Condition) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément Condition est facultatif. 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.

Si vous spécifiez plusieurs éléments Condition dans une instruction, ou plusieurs clés dans un seul élément Condition, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une OR opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource

uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez [Éléments d'une politique IAM : variables et identifications](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. 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.

Important

Plusieurs clés de condition sont propres à une ressource et certaines actions d'API utilisent plusieurs ressources. Si vous écrivez une déclaration de stratégie avec une clé de condition, utilisez l'élément `Resource` de la déclaration pour spécifier la ressource à laquelle la clé de condition s'applique. Dans le cas contraire, la stratégie peut empêcher totalement les utilisateurs d'exécuter l'action, car le contrôle de la condition échoue pour les ressources auxquelles la clé de condition ne s'applique pas. Si vous ne voulez pas spécifier de ressource ou si vous avez écrit l'élément `Action` de votre stratégie pour inclure plusieurs actions d'API, vous devez utiliser le type de condition `...IfExists` pour garantir que la clé de condition est ignorée pour les ressources qui ne l'utilisent pas. Pour plus d'informations, voir... [IfExists conditions énoncées](#) dans le guide de l'utilisateur IAM.

AWS IoT SiteWise définit son propre ensemble de clés de condition et prend également en charge l'utilisation de certaines clés de condition globales. 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.

AWS IoT SiteWise clés de condition

Clé de condition	Description	Types
<code>iotsitewise:isAssociatedWithAssetProperty</code>	Si les flux de données sont associés à une propriété d'actif. Utilisez cette clé de condition pour définir les autorisations en fonction de l'existence d'une propriété d'actif associée pour les flux de données. Exemple de valeur : <code>true</code>	Chaîne

Clé de condition	Description	Types
<code>iotsitewise:assetHierarchyPath</code>	Le chemin hiérarchique de la ressource, qui est une chaîne d'actifs séparés IDs chacun par une barre oblique. Utilisez cette clé de condition pour définir des autorisations en fonction d'un sous-ensemble de votre hiérarchie de tous les actifs de votre compte. Exemple de valeur : /a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/ a1b2c3d4-5678-90ab-cdef-6666EXAMPLE	Chaîne
<code>iotsitewise:propertyId</code>	ID d'une propriété d'actif. Utilisez cette clé de condition pour définir des autorisations basées sur une propriété spécifiée d'un modèle de ressource. Cette clé de condition s'applique à tous les actifs de ce modèle. Exemple de valeur : a1b2c3d4-5678-90ab-cdef-3333EXAMPLE	Chaîne

Clé de condition	Description	Types
<code>iotsitewise:childAssetId</code>	ID d'une ressource associée en tant qu'enfant à une autre ressource. Utilisez cette clé de condition pour définir les autorisations en fonction des ressources enfants. Pour définir des autorisations en fonction des ressources parent, utilisez la section ressource d'une instruction de stratégie. Exemple de valeur : a1b2c3d4-5678-90ab-cdef-66666EXAMPLE	Chaîne
<code>iotsitewise:iam</code>	L'ARN d'une identité IAM lors de la liste des politiques d'accès. Utilisez cette clé de condition pour définir les autorisations de politique d'accès pour une identité IAM. Exemple de valeur : arn:aws:iam::123456789012:user/JohnDoe	Chaîne, null
<code>iotsitewise:propertyAlias</code>	Alias qui identifie une propriété d'actif ou un flux de données. Utilisez cette clé de condition pour définir les autorisations en fonction de l'alias.	Chaîne

Clé de condition	Description	Types
<code>iotsitewise:user</code>	ID d'un utilisateur du IAM Identity Center lors de la liste des politiques d'accès. Utilisez cette clé de condition pour définir les autorisations de politique d'accès pour un utilisateur d'IAM Identity Center. Exemple de valeur : a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-aaaaEXAMPLE	Chaîne, null
<code>iotsitewise:group</code>	ID d'un groupe IAM Identity Center lors de la liste des politiques d'accès. Utilisez cette clé de condition pour définir les autorisations de politique d'accès pour un groupe IAM Identity Center. Exemple de valeur : a1b2c3d4e5-a1b2c3d4-5678-90ab-cdef-bbbbbEXAMPLE	Chaîne, null
<code>iotsitewise:portal</code>	ID d'un portail dans une stratégie d'accès. Utilisez cette clé de condition pour définir les autorisations de stratégie d'accès basées sur un portail. Exemple de valeur : a1b2c3d4-5678-90ab-cdef-77777EXAMPLE	Chaîne, null

Clé de condition	Description	Types
<code>iotsitewise:project</code>	ID d'un projet dans une stratégie d'accès ou ID d'un projet pour un tableau de bord. Utilisez cette clé de condition pour définir des autorisations de stratégie d'accès ou de tableau de bord en fonction d'un projet. Exemple de valeur : a1b2c3d4-5678-90ab-cdef-88888EXAMPLE	Chaîne, null

Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section [Actions définies par AWS IoT SiteWise](#).

Exemples

Pour consulter des exemples de politiques AWS IoT SiteWise basées sur l'identité, consultez. [AWS IoT SiteWise exemples de politiques basées sur l'identité](#)

AWS IoT SiteWise exemples de politiques basées sur l'identité

Par défaut, les entités (utilisateurs et rôles) ne sont pas autorisées à créer ou à modifier AWS IoT SiteWise des ressources. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour ajuster les autorisations, un administrateur AWS Identity and Access Management (IAM) doit effectuer les opérations suivantes :

1. Créez des politiques IAM qui accordent aux utilisateurs et aux rôles l'autorisation d'effectuer des opérations d'API spécifiques sur les ressources dont ils ont besoin.
2. Associez ces politiques aux utilisateurs ou aux groupes qui ont besoin de ces autorisations.

Pour apprendre à créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de politique JSON, veuillez consulter [Création de politiques dans l'onglet JSON](#) dans le Guide de l'utilisateur IAM.

Rubriques

- [Bonnes pratiques en matière de politiques](#)
- [Utiliser la AWS IoT SiteWise console](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations](#)
- [Permettre aux utilisateurs d'ingérer des données dans des actifs d'une seule hiérarchie](#)
- [Afficher les AWS IoT SiteWise ressources en fonction des balises](#)

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer AWS IoT SiteWise 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 AWS 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.

Utiliser la AWS IoT SiteWise console

Pour accéder à la AWS IoT SiteWise console, vous avez besoin d'un ensemble d'autorisations de base. Ces autorisations vous permettent de consulter et de gérer les informations relatives AWS IoT SiteWise aux ressources de votre AWS compte.

Si vous définissez une politique trop restrictive, la console risque de ne pas fonctionner comme prévu pour les utilisateurs ou les rôles (entités) concernés par cette politique. Pour garantir que ces entités peuvent toujours utiliser la AWS IoT SiteWise console, associez-leur la politique [AWSIoTSiteWiseConsoleFullAccess](#) gérée ou définissez des autorisations équivalentes pour ces entités. Pour plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

Si les entités utilisent uniquement la AWS Command Line Interface (CLI) ou l' AWS IoT SiteWise API, et non la console, elles n'ont pas besoin de ces autorisations minimales. Dans ce cas, donnez-leur simplement accès aux actions spécifiques dont ils ont besoin pour leurs tâches d'API.

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:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

Permettre aux utilisateurs d'ingérer des données dans des actifs d'une seule hiérarchie

Dans cet exemple, vous souhaitez autoriser un utilisateur de votre AWS compte à écrire des données sur toutes les propriétés des actifs dans une hiérarchie d'actifs spécifique, en commençant par l'actif racinea1b2c3d4-5678-90ab-cdef-2222EXAMPLE. La stratégie accorde l'autorisation `iotsitewise:BatchPutAssetPropertyValue` à l'utilisateur. Cette stratégie utilise la clé de

condition `iotsitewise:assetHierarchyPath` pour restreindre l'accès aux ressources dont le chemin hiérarchique correspond à l'actif ou à ses descendants.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PutAssetPropertyValuesForHierarchy",
      "Effect": "Allow",
      "Action": "iotsitewise:BatchPutAssetPropertyValue",
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",
      "Condition": {
        "StringLike": {
          "iotsitewise:assetHierarchyPath": [
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE",
            "/a1b2c3d4-5678-90ab-cdef-2222EXAMPLE/*"
          ]
        }
      }
    }
  ]
}
```

Afficher les AWS IoT SiteWise ressources en fonction des balises

Utilisez les conditions de votre politique basée sur l'identité pour contrôler l'accès aux AWS IoT SiteWise ressources en fonction des balises. Cet exemple montre comment créer une politique permettant de visualiser les actifs. Toutefois, l'autorisation est accordée uniquement si la balise de ressource `Owner` a pour valeur le nom d'utilisateur de cet utilisateur. Cette politique accorde également l'autorisation d'effectuer cette action sur la console.

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

```
    },  
    {  
      "Sid": "DescribeAssetIfOwner",  
      "Effect": "Allow",  
      "Action": "iotsitewise:DescribeAsset",  
      "Resource": "arn:aws:iotsitewise:*:*:asset/*",  
      "Condition": {  
        "StringEquals": {  
          "aws:ResourceTag/Owner": "${aws:username}"  
        }  
      }  
    }  
  ]  
}
```

Associez cette politique aux utilisateurs de votre compte. Si un utilisateur nommé `richard-roe` tente de consulter une AWS IoT SiteWise ressource, celle-ci doit être étiquetée `Owner=richard-roe` ou `owner=richard-roe`. Dans le cas contraire, Richard se voit refuser l'accès. Les noms des clés des balises de condition ne distinguent pas les majuscules et minuscules. Donc, `Owner` correspond aux deux `Owner` et `owner`. Pour plus d'informations, consultez [Éléments de politique JSON IAM : Condition](#) dans le Guide de l'utilisateur IAM.

Gérez l'accès à l'aide de politiques dans AWS IoT SiteWise

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

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.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. 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. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

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.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique 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. 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

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.

Listes de contrôle d'accès (ACLs)

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.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la [présentation de la liste de contrôle d'accès \(ACL\)](#) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- **Politiques de contrôle des services (SCPs)** : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations et consultez SCPs les [politiques de contrôle des services](#) dans le Guide de AWS Organizations l'utilisateur.
- **Politiques de contrôle des ressources (RCPs)** : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les

autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, consultez la section [Resource control policies \(RCPs\)](#) dans le guide de AWS Organizations l'utilisateur.

- **Politiques de séance** : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. 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.

AWS politiques gérées pour AWS IoT SiteWise

Simplifiez l'ajout d'autorisations aux utilisateurs, aux groupes et aux rôles à l'aide de politiques AWS gérées plutôt que de rédiger vous-même des politiques. Il faut du temps et de l'expertise pour [créer des politiques IAM gérées par les clients](#) qui fournissent des autorisations précises à votre équipe. Pour une configuration plus rapide, pensez à utiliser nos politiques AWS gérées pour les cas d'utilisation courants. Trouvez les politiques AWS gérées dans votre AWS compte. Pour plus d'informations sur les politiques gérées AWS, consultez [Politiques gérées AWS](#) dans le Guide de l'utilisateur IAM.

AWS les services se chargent de mettre à jour et de maintenir les politiques AWS gérées, ce qui signifie que vous ne pouvez pas modifier les autorisations de ces politiques. Occasionnellement, des autorisations AWS IoT SiteWise peuvent être ajoutées pour s'adapter aux nouvelles fonctionnalités, ce qui a un impact sur toutes les identités associées à la politique. Ces mises à jour sont courantes lors de l'introduction de nouveaux services ou fonctionnalités. Cependant, les autorisations ne sont jamais supprimées, ce qui garantit que vos configurations restent intactes.

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 une liste avec des descriptions des politiques relatives aux fonctions de travail, voir les [politiques AWS gérées pour les fonctions de travail](#) dans le guide de l'utilisateur d'IAM.

AWS politique gérée : AWSIoT Site WiseReadOnlyAccess

Utilisez la politique AWSIoT Site WiseReadOnlyAccess AWS gérée pour autoriser l'accès en lecture seule à. AWS IoT SiteWise

Vous pouvez associer la politique AWSIoT Site WiseReadOnlyAccess à vos identités IAM.

Autorisations au niveau du service

Cette politique fournit un accès en lecture seule à. AWS IoT SiteWise Cette politique n'inclut aucune autre autorisation de service.

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

AWS politique gérée : AWSService RoleForlo TSite Wise

Le AWSServiceRoleForIoTSiteWise rôle utilise la AWSServiceRoleForIoTSiteWise politique avec les autorisations suivantes. Cette politique :

- Permet AWS IoT SiteWise de déployer des passerelles SiteWise Edge (qui s'exécutent surAWS IoT Greengrass).
- Permet d' AWS IoT SiteWise effectuer une journalisation.

- Permet AWS IoT SiteWise d'exécuter une requête de recherche de métadonnées sur la AWS IoT TwinMaker base de données.

Si vous utilisez AWS IoT SiteWise un compte utilisateur unique, le `AWSServiceRoleForIoTSiteWise` rôle crée la `AWSServiceRoleForIoTSiteWise` politique dans votre compte IAM et l'associe aux rôles `AWSServiceRoleForIoTSiteWise` [liés au service](#) pour. AWS IoT SiteWise

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSiteWiseAccessLogGroup",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
    },
    {
      "Sid": "AllowSiteWiseAccessLog",
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
    }
  ]
}
```

```

},
{
  "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
  "Effect": "Allow",
  "Action": [
    "iottwinmaker:GetWorkspace",
    "iottwinmaker:ExecuteQuery"
  ],
  "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "iottwinmaker:linkedServices": [
        "IOTSITWISE"
      ]
    }
  }
}
]
}

```

AWS IoT SiteWise mises à jour des politiques AWS gérées

Vous pouvez consulter les informations relatives aux mises à jour des politiques AWS gérées pour AWS IoT SiteWise, à partir de la date à laquelle ce service a commencé à suivre les modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page Historique du AWS IoT SiteWise document.

Modification	Description	Date
AWSServiceRoleForIoTSiteWise — Mise à jour d'une politique existante	AWS IoT SiteWise peut désormais exécuter une requête de recherche de métadonnées sur la AWS IoT TwinMaker base de données.	6 novembre 2023
AWSIoTSiteWiseReadOnlyAccess – Mise à jour d'une politique existante	AWS IoT SiteWise a ajouté un nouveau préfixe de politique <code>BatchGet*</code> , qui vous permet d'effectuer des opérations de lecture par lots.	16 septembre 2022

Modification	Description	Date
AWSIoTSiteWiseReadOnlyAccess : nouvelle politique	AWS IoT SiteWise a ajouté une nouvelle politique pour accorder un accès en lecture seule à AWS IoT SiteWise	24 novembre 2021
AWS IoT SiteWise a commencé à suivre les modifications	AWS IoT SiteWise a commencé à suivre les modifications apportées AWS à ses politiques gérées.	24 novembre 2021

Utilisez des rôles liés à un service pour AWS IoT SiteWise

AWS IoT SiteWise utilise des AWS Identity and Access Management rôles liés à un [service](#) (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié AWS IoT SiteWise. Les rôles liés à un service sont prédéfinis par AWS IoT SiteWise et incluent toutes les autorisations dont le service a besoin pour appeler d'autres AWS services en votre nom.

Les rôles liés aux services simplifient la configuration de AWS IoT SiteWise en incluant automatiquement toutes les autorisations nécessaires. AWS IoT SiteWise définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul AWS IoT SiteWise peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisations. Et cette politique d'autorisation ne peut être attachée à aucune 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 SiteWise 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 Rôle lié au service. Sélectionnez un Oui ayant un lien pour consulter la documentation du rôle lié à un service, pour ce service.

Rubriques

- [Autorisations des rôles liés à un service pour AWS IoT SiteWise](#)
- [Créer un rôle lié à un service pour AWS IoT SiteWise](#)

- [Mettre à jour un rôle lié à un service pour AWS IoT SiteWise](#)
- [Supprimer un rôle lié à un service pour AWS IoT SiteWise](#)
- [Régions prises en charge pour les rôles AWS IoT SiteWise liés à un service](#)
- [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#)

Autorisations des rôles liés à un service pour AWS IoT SiteWise

AWS IoT SiteWise utilise le rôle lié au service nommé `AWSServiceRoleForIoTSiteWise`. AWS IoT SiteWise utilise ce rôle lié au service pour déployer des passerelles SiteWise Edge (qui s'exécutent sur AWS IoT Greengrass) et effectuer la journalisation.

Le rôle `AWSServiceRoleForIoTSiteWise` lié au service utilise la `AWSServiceRoleForIoTSiteWise` politique avec les autorisations suivantes. Cette politique :

- Permet AWS IoT SiteWise de déployer des passerelles SiteWise Edge (qui s'exécutent sur AWS IoT Greengrass).
- Permet d' AWS IoT SiteWise effectuer une journalisation.
- Permet AWS IoT SiteWise d'exécuter une requête de recherche de métadonnées sur la AWS IoT TwinMaker base de données.

Pour plus d'informations sur les actions autorisées dans `AWSServiceRoleForIoTSiteWise`, consultez [les politiques AWS gérées pour AWS IoT SiteWise](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowSiteWiseReadGreenGrass",
      "Effect": "Allow",
      "Action": [
        "greengrass:GetAssociatedRole",
        "greengrass:GetCoreDefinition",
        "greengrass:GetCoreDefinitionVersion",
        "greengrass:GetGroup",
        "greengrass:GetGroupVersion"
      ],
      "Resource": "*"
    },
    {
```

```

    "Sid": "AllowSiteWiseAccessLogGroup",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogGroup",
        "logs:DescribeLogGroups"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*"
},
{
    "Sid": "AllowSiteWiseAccessLog",
    "Effect": "Allow",
    "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
    ],
    "Resource": "arn:aws:logs:*:*:log-group:/aws/iotsitewise*:log-stream:*"
},
{
    "Sid": "AllowSiteWiseAccessSiteWiseManagedWorkspaceInTwinMaker",
    "Effect": "Allow",
    "Action": [
        "iottwinmaker:GetWorkspace",
        "iottwinmaker:ExecuteQuery"
    ],
    "Resource": "arn:aws:iottwinmaker:*:*:workspace/*",
    "Condition": {
        "ForAnyValue:StringEquals": {
            "iottwinmaker:linkedServices": [
                "IOTSITewise"
            ]
        }
    }
}
]
}

```

Vous pouvez utiliser les journaux pour surveiller et dépanner vos passerelles SiteWise Edge. Pour de plus amples informations, veuillez consulter [Surveiller les journaux de la passerelle SiteWise Edge](#).

Pour autoriser une entité IAM (telle qu'un utilisateur, un groupe ou un rôle) à créer, modifier ou supprimer un rôle lié à un service, configurez d'abord les autorisations. Pour plus d'informations, consultez [Autorisations de rôles liés à un service](#) dans le Guide de l'utilisateur IAM.

Créer un rôle lié à un service pour AWS IoT SiteWise

AWS IoT SiteWise nécessite un rôle lié au service pour effectuer certaines actions et accéder aux ressources en votre nom. Un rôle lié à un service est un type unique de rôle AWS Identity and Access Management (IAM) directement lié à. AWS IoT SiteWise En créant ce rôle, vous accordez AWS IoT SiteWise les autorisations nécessaires pour accéder à d'autres AWS services et ressources nécessaires à son fonctionnement, tels qu'Amazon S3 pour le stockage de données ou AWS IoT pour la communication entre appareils.

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous effectuez les opérations suivantes dans la AWS IoT SiteWise console, AWS IoT SiteWise crée le rôle lié au service pour vous.

- Créez une passerelle Greengrass V1.
- Configurez l'option de journalisation.
- Choisir le bouton d'inscription dans le bandeau d'exécution de la requête.

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 effectuez une opération dans la AWS IoT SiteWise console, AWS IoT SiteWise crée à nouveau le rôle lié au service pour vous.

Vous pouvez également utiliser la console ou l'API IAM pour créer un rôle lié à un service pour. AWS IoT SiteWise

- Pour ce faire, dans la console IAM, créez un rôle avec la politique `AWSServiceRoleForIoTSiteWise` et une relation de confiance avec `eciotsitewise.amazonaws.com`.
- Pour ce faire, utilisez l'API AWS CLI ou IAM, créez un rôle avec la `arn:aws:iam::aws:policy/aws-service-role/AWSServiceRoleForIoTSiteWise` politique et une relation de confiance avec `eciotsitewise.amazonaws.com`.

Pour plus d'informations, consultez la section [Créer 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.

Mettre à jour un rôle lié à un service pour AWS IoT SiteWise

AWS IoT SiteWise ne vous permet pas de modifier le rôle lié au service `AWSServiceRoleForIoTSiteWise`. Une fois que vous avez créé un rôle lié à un service, vous ne pouvez pas changer le nom du rôle, car plusieurs entités peuvent faire référence au rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, voir [Mettre à jour un rôle lié à un service](#) dans le Guide de l'utilisateur IAM.

Supprimer un rôle lié à un service pour AWS IoT SiteWise

Si une fonctionnalité ou un service nécessitant un rôle lié à un service n'est plus utilisé, il est conseillé de supprimer le rôle associé. Cela permet d'éviter d'avoir une entité inactive qui n'est ni surveillée ni maintenue. Cependant, vous devez nettoyer les ressources de votre rôle lié à un service avant de pouvoir les supprimer manuellement.

Note

Si le AWS IoT SiteWise service utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, attendez quelques minutes et réessayez.

Pour supprimer les AWS IoT SiteWise ressources utilisées par les `AWSServiceRoleForIoTSiteWise`

1. Désactivez la journalisation pour AWS IoT SiteWise. Pour de plus amples informations, consultez [Modifier votre niveau de journalisation](#).
2. Supprimez toutes les passerelles SiteWise Edge actives.

Pour supprimer manuellement le rôle lié à un service à l'aide d'IAM

Utilisez la console IAM, le AWS CLI, ou l' AWS API pour supprimer le rôle lié au service `AWSServiceRoleForIoTSiteWise`. Pour plus d'informations, consultez la section [Supprimer des rôles ou des profils d'instance](#) dans le guide de l'utilisateur IAM.

Régions prises en charge pour les rôles AWS IoT SiteWise liés à un service

AWS IoT SiteWise prend en charge l'utilisation de rôles liés au service dans toutes les régions où le service est disponible. Pour plus d'informations, consultez [Points de terminaison et quotas AWS IoT SiteWise](#).

Utiliser les rôles de service pour AWS IoT SiteWise Monitor

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.

Pour permettre aux utilisateurs du portail SiteWise Monitor fédéré d'accéder à vos AWS IAM Identity Center ressources AWS IoT SiteWise et à vos ressources, vous devez attribuer un rôle de service à chaque portail que vous créez. Le rôle de service doit spécifier SiteWise Monitor en tant qu'entité de confiance et inclure la politique [AWSIoTSiteWiseMonitorPortalAccess](#) gérée ou définir [des autorisations équivalentes](#). Cette politique est gérée par AWS et définit l'ensemble d'autorisations que SiteWise Monitor utilise pour accéder à vos ressources AWS IoT SiteWise et à celles d'IAM Identity Center.

Lorsque vous créez un portail SiteWise Monitor, vous devez choisir un rôle qui permet aux utilisateurs de ce portail d'accéder à vos ressources AWS IoT SiteWise et à celles d'IAM Identity Center. La AWS IoT SiteWise console peut créer et configurer le rôle pour vous. Vous pourrez modifier le rôle dans IAM ultérieurement. Les utilisateurs de votre portail rencontreront des problèmes lors de l'utilisation de leurs portails SiteWise Monitor si vous supprimez les autorisations requises pour le rôle ou si vous supprimez le rôle.

Note

Les portails créés avant le 29 avril 2020 ne nécessitaient pas de rôles de service. Si vous avez créé des portails avant cette date, vous devez joindre des rôles de service pour continuer à les utiliser. Pour ce faire, accédez à la page Portails de la [AWS IoT SiteWise console](#), puis choisissez Migrer tous les portails pour utiliser les rôles IAM.

Les sections suivantes décrivent comment créer et gérer le rôle de service SiteWise Monitor dans le AWS Management Console ou le AWS Command Line Interface.

Table des matières

- [Autorisations de rôle de service pour SiteWise Monitor \(Classic\)](#)
- [Autorisations de rôle de service pour SiteWise Monitor \(compatible avec l'IA\)](#)
- [Gérer le rôle de service SiteWise Monitor \(console\)](#)
 - [Rechercher le rôle de service d'un portail \(console\)](#)

- [Création d'un rôle de service de SiteWise surveillance \(AWS IoT SiteWise console\)](#)
- [Création d'un rôle de service de SiteWise surveillance \(console IAM\)](#)
- [Modifier le rôle de service d'un portail \(console\)](#)
- [Gérer le rôle de service SiteWise Monitor \(CLI\)](#)
- [Trouver le rôle de service d'un portail \(CLI\)](#)
- [Création du rôle de service SiteWise Monitor \(CLI\)](#)
- [SiteWise Surveillez les mises à jour de AWSIoTSiteWiseMonitorServiceRole](#)

Autorisations de rôle de service pour SiteWise Monitor (Classic)

Lorsque vous créez un portail, vous AWS IoT SiteWise permet de créer un rôle dont le nom commence par `AWSIoTSiteWiseMonitorServiceRole`. Ce rôle permet aux utilisateurs fédérés de SiteWise Monitor d'accéder à la configuration de votre portail, aux actifs, aux données des actifs, ainsi qu'à la configuration d'IAM Identity Center.

Le rôle approuve le fait que le service suivant endosse le rôle :

- `monitor.iotsitewise.amazonaws.com`

Le rôle utilise la politique d'autorisation suivante, qui commence par `AWSIoTSiteWiseMonitorServicePortalPolicy`, pour permettre aux utilisateurs de SiteWise Monitor d'effectuer des actions sur les ressources de votre compte. La stratégie [AWSIoTSiteWiseMonitorPortalAccess](#) gérée définit des autorisations équivalentes.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal",
        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",

```

```

        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",
        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
}

```



```

    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:CreateAlarmModel",
      "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:RequestTag/iotsitewisemonitor": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iotevents:UpdateAlarmModel",
      "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "aws:ResourceTag/iotsitewisemonitor": "false"
      }
    }
  },
  {
    "Effect": "Allow",
    "Action": [
      "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "iam:PassedToService": [
          "iotevents.amazonaws.com"
        ]
      }
    }
  }
]

```

```
}
```

Pour plus d'informations sur les autorisations requises pour les alarmes, consultez [Configurez les autorisations pour les alarmes liées aux événements dans AWS IoT SiteWise](#).

Lorsqu'un utilisateur du portail se connecte, SiteWise Monitor crée une [politique de session](#) basée sur l'intersection entre le rôle de service et les politiques d'accès de cet utilisateur. Les stratégies d'accès définissent le niveau d'accès des identités à vos portails et projets. Pour plus d'informations sur les autorisations du portail et les politiques d'accès, consultez [Administrez vos portails SiteWise Monitor](#) et [CreateAccessPolicy](#).

Autorisations de rôle de service pour SiteWise Monitor (compatible avec l'IA)

Lorsque vous créez un portail, vous AWS IoT SiteWise permet de créer un rôle dont le nom commence par `lo TSite WisePortalRole`. Ce rôle permet aux utilisateurs fédérés de SiteWise Monitor d'accéder à la configuration de votre portail, aux actifs, aux données des actifs, ainsi qu'à la configuration d'IAM Identity Center.

 Warning

Les rôles de propriétaire de projet et de visionneur de projet ne sont pas pris en charge pour SiteWise Monitor (compatible avec l'IA).

Le rôle approuve le fait que le service suivant endosse le rôle :

- `monitor.iotsitewise.amazonaws.com`

Le rôle utilise la politique d'autorisation suivante, qui commence par `lo TSite Wise AIPortal AccessPolicy`, pour permettre aux utilisateurs de SiteWise Monitor d'effectuer des actions sur les ressources de votre compte.

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

```

        "iotsitewise:ListProjects",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise>CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:ListAssets",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:ListAssetProperties",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:GetInterpolatedAssetPropertyValues",
        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",
        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]

```

```
}
```

Lorsqu'un utilisateur du portail se connecte, SiteWise Monitor crée une [politique de session](#) basée sur l'intersection entre le rôle de service et les politiques d'accès de cet utilisateur.

Gérer le rôle de service SiteWise Monitor (console)

Console AWS IoT SiteWise Facilite la gestion du rôle de service SiteWise Monitor pour les portails. Lors de la création d'un portail, la console vérifie les rôles existants susceptibles d'être rattachés. Si aucun n'est disponible, la console peut créer et configurer un rôle de service pour vous. Pour de plus amples informations, veuillez consulter [Création d'un portail dans SiteWise Monitor](#).

Rubriques

- [Rechercher le rôle de service d'un portail \(console\)](#)
- [Création d'un rôle de service de SiteWise surveillance \(AWS IoT SiteWise console\)](#)
- [Création d'un rôle de service de SiteWise surveillance \(console IAM\)](#)
- [Modifier le rôle de service d'un portail \(console\)](#)

Rechercher le rôle de service d'un portail (console)

Suivez les étapes ci-dessous pour trouver le rôle de service associé à un portail SiteWise Monitor.

Pour rechercher le rôle de service d'un portail

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation de gauche, choisissez Portals (Portails).
3. Choisissez le portail pour lequel vous souhaitez rechercher le rôle de service.

Le rôle associé au portail apparaît sous Autorisations, rôle de service.

Création d'un rôle de service de SiteWise surveillance (AWS IoT SiteWise console)

Lorsque vous créez un portail SiteWise Monitor, vous pouvez créer un rôle de service pour votre portail. Pour de plus amples informations, veuillez consulter [Création d'un portail dans SiteWise Monitor](#).

Vous pouvez également créer un rôle de service pour un portail existant dans la AWS IoT SiteWise console. Cela remplace le rôle de service existant du portail.

Pour créer un rôle de service pour un portail existant

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Portails.
3. Choisissez le portail pour lequel vous souhaitez créer un rôle de service.
4. Sous Détails du portail, choisissez Modifier.
5. Sous Autorisations, choisissez Créer et utiliser un nouveau rôle de service dans la liste.
6. Saisissez un nom pour votre nouveau rôle.
7. Choisissez Enregistrer.

Création d'un rôle de service de SiteWise surveillance (console IAM)

Vous pouvez créer un rôle de service à partir du modèle de rôle de service de la console IAM. Ce modèle de rôle inclut la politique [AWSIoTSiteWiseMonitorPortalAccess](#) gérée et spécifie SiteWise Monitor comme une entité de confiance.

Pour créer un rôle de service à partir du modèle de rôle de service du portail

1. Accédez à la [Console IAM](#).
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Sélectionnez Create role (Créer un rôle).
4. Dans Choisir un cas d'utilisation, sélectionnez IoT SiteWise.
5. Dans Sélectionnez votre cas d'utilisation, choisissez IoT SiteWise Monitor - Portal.
6. Choisissez Suivant : Autorisations.
7. Choisissez Suivant : Balises.
8. Choisissez Suivant : Vérification.
9. Entrez un nom de rôle pour le nouveau rôle de service.
10. Choisissez Créer un rôle.

Modifier le rôle de service d'un portail (console)

Utilisez la procédure suivante pour choisir un autre rôle de service de SiteWise surveillance pour un portail.

Pour modifier le rôle de service d'un portail

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation, choisissez Portails.
3. Choisissez le portail pour lequel vous souhaitez modifier le rôle de service.
4. Sous Détails du portail, choisissez Modifier.
5. Sous Autorisations, choisissez Utiliser un rôle existant.
6. Choisissez un rôle existant à attacher à ce portail.
7. Choisissez Enregistrer.

Gérer le rôle de service SiteWise Monitor (CLI)

Vous pouvez utiliser le AWS CLI pour les tâches de gestion des rôles de service de portail suivantes :

Rubriques

- [Trouver le rôle de service d'un portail \(CLI\)](#)
- [Création du rôle de service SiteWise Monitor \(CLI\)](#)

Trouver le rôle de service d'un portail (CLI)

Pour trouver le rôle de service associé à un portail de SiteWise surveillance, exécutez la commande suivante pour répertorier tous vos portails dans la région actuelle.

```
aws iotsitewise list-portals
```

L'opération renvoie une réponse qui contient les résumés de vos portails au format suivant.

```
{
  "portalSummaries": [
    {
      "id": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
      "name": "WindFarmPortal",
      "description": "A portal that contains wind farm projects for Example Corp.",
      "roleArn": "arn:aws:iam::123456789012:role/service-role/role-name",
      "startUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
      "creationDate": "2020-02-04T23:01:52.90248068Z",
      "lastUpdateDate": "2020-02-04T23:01:52.90248078Z"
    }
  ]
}
```

```
]
}
```

Vous pouvez également utiliser cette [DescribePortal](#) opération pour trouver le rôle de votre portail si vous connaissez l'ID de votre portail.

Création du rôle de service SiteWise Monitor (CLI)

Suivez les étapes ci-dessous pour créer un nouveau rôle de service de SiteWise surveillance.

Pour créer un rôle SiteWise de service de surveillance

1. Créez un rôle avec une politique de confiance qui permet à SiteWise Monitor d'assumer ce rôle. Cet exemple crée un rôle nommé **MySiteWiseMonitorPortalRole** à partir d'une stratégie d'approbation stockée dans une chaîne JSON.

Linux, macOS, or Unix

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document '{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "monitor.iotsitewise.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

Windows command prompt

```
aws iam create-role --role-name MySiteWiseMonitorPortalRole --assume-role-
policy-document "{\"Version\":\"2012-10-17\",\"Statement\": [{\"Effect\":\"Allow\", \"Principal\": {\"Service\": \"monitor.iotsitewise.amazonaws.com\"}, \"Action\": \"sts:AssumeRole\"}]}"
```

2. Copiez l'ARN de rôle du rôle des métadonnées dans la sortie. Lorsque vous créez un portail, vous utilisez cet ARN pour associer le rôle à votre portail. Pour plus d'informations sur la création d'un portail, consultez [CreatePortall](#) la référence des AWS IoT SiteWise API.

3. a. Pour le SiteWise moniteur (classique) : associez la `AWSIoTSiteWiseMonitorPortalAccess` politique au rôle ou associez une politique définissant des autorisations équivalentes.

```
aws iam attach-role-policy --role-name MySiteWiseMonitorPortalRole --policy-arn
arn:aws:iam::aws:policy/service-role/AWSIoTSiteWiseMonitorPortalAccess
```

- b. Pour le SiteWise moniteur (compatible avec l'IA) : attachez la `IoTSiteWiseAIPortalAccessPolicy` politique au rôle ou attachez une politique qui définit des autorisations équivalentes. Par exemple, créez une politique avec des autorisations d'accès au portail. L'exemple suivant crée une politique nommée `MySiteWiseMonitorPortalAccess`.

```
aws iam create-policy \
  --policy-name MySiteWiseMonitorPortalAccess \
  --policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iotsitewise:CreateProject",
          "iotsitewise:DescribePortal",
          "iotsitewise:ListProjects",
          "iotsitewise:DescribeProject",
          "iotsitewise:UpdateProject",
          "iotsitewise>DeleteProject",
          "iotsitewise:CreateDashboard",
          "iotsitewise:DescribeDashboard",
          "iotsitewise:UpdateDashboard",
          "iotsitewise>DeleteDashboard",
          "iotsitewise:ListDashboards",
          "iotsitewise:ListAssets",
          "iotsitewise:DescribeAsset",
          "iotsitewise:ListAssociatedAssets",
          "iotsitewise:ListAssetProperties",
          "iotsitewise:DescribeAssetProperty",
          "iotsitewise:GetAssetPropertyValue",
          "iotsitewise:GetAssetPropertyValueHistory",
          "iotsitewise:GetAssetPropertyAggregates",
          "iotsitewise:GetInterpolatedAssetPropertyValues",
```



```

        "iotsitewise:BatchGetAssetPropertyAggregates",
        "iotsitewise:BatchGetAssetPropertyValue",
        "iotsitewise:BatchGetAssetPropertyValueHistory",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:DescribeAssetCompositeModel",
        "iotsitewise:DescribeAssetModelCompositeModel",
        "iotsitewise:ListAssetModelProperties",
        "iotsitewise:ExecuteQuery",
        "iotsitewise:ListTimeSeries",
        "iotsitewise:DescribeTimeSeries",
        "iotsitewise:InvokeAssistant",
        "iotsitewise:DescribeDataset",
        "iotsitewise:ListDatasets",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource",
        "iottwinmaker:ListWorkspaces",
        "iottwinmaker:ExecuteQuery",
        "iottwinmaker:GetWorkspace",
        "identitystore:DescribeUser"
    ],
    "Resource": "*"
}
]
}'

```

Pour attacher un rôle de service à un portail existant

1. Pour récupérer les détails existants du portail, exécutez la commande suivante. Remplacez *portal-id* par l'ID du portail.

```
aws iotsitewise describe-portal --portal-id portal-id
```

L'opération renvoie une réponse qui contient les détails du portail dans le format suivant.

```

{
  "portalId": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "portalArn": "arn:aws:iotsitewise:region:account-id:portal/a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",

```

```

    "portalName": "WindFarmPortal",
    "portalDescription": "A portal that contains wind farm projects for Example Corp.",
    "portalClientId": "E-1a2b3c4d5e6f_sn6tbqHVzLWVEXAMPLE",
    "portalStartUrl": "https://a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE.app.iotsitewise.aws",
    "portalContactEmail": "support@example.com",
    "portalStatus": {
        "state": "ACTIVE"
    },
    "portalCreationDate": "2020-04-29T23:01:52.90248068Z",
    "portalLastUpdateDate": "2020-04-29T00:28:26.103548287Z",
    "roleArn": "arn:aws:iam::123456789012:role/service-role/AWSIoTSiteWiseMonitorServiceRole_1aEXAMPLE"
}

```

2. Pour attacher un rôle de service à un portail, exécutez la commande suivante. Remplacez *role-arn* par l'ARN du rôle de service et remplacez les paramètres restants par les valeurs existantes du portail.

```

aws iotsitewise update-portal \
  --portal-id portal-id \
  --role-arn role-arn \
  --portal-name portal-name \
  --portal-description portal-description \
  --portal-contact-email portal-contact-email

```

SiteWise Surveillez les mises à jour de AWSIoT SiteWiseMonitorServiceRole

Vous pouvez consulter les détails des mises à jour de AWSIoT SiteWiseMonitorServiceRole for SiteWise Monitor, à partir du moment où ce service a commencé à suivre les modifications. Pour recevoir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page Historique du AWS IoT SiteWise document.

Modification	Description	Date
AWSIoT SiteWiseMonitorPortal Access — Politique mise à jour	AWS IoT SiteWise a mis à jour la politique AWSIoT SiteWiseMonitorPortal	27 mai 2021

Modification	Description	Date
	Access gérée pour la fonctionnalité d'alarmes.	
AWS IoT SiteWise a commencé à suivre les modifications	AWS IoT SiteWise a commencé à suivre les modifications apportées à son rôle de service.	15 décembre 2020

Configurez les autorisations pour les alarmes liées aux événements dans AWS IoT SiteWise

Lorsque vous utilisez un modèle AWS IoT Events d'alarme pour surveiller la propriété AWS IoT SiteWise d'un actif, vous devez disposer des autorisations IAM suivantes :

- Rôle de AWS IoT Events service qui permet d' AWS IoT Events envoyer des données à AWS IoT SiteWise. Pour plus d'informations, consultez la section [Gestion des identités et des accès AWS IoT Events](#) dans le Guide du AWS IoT Events développeur.
- Vous devez disposer des autorisations AWS IoT SiteWise d'action suivantes :
`iotsitewise:DescribeAssetModel`
`etiotssitewise:UpdateAssetModelPropertyRouting`. Ces autorisations permettent d' AWS IoT SiteWise envoyer les valeurs des propriétés des actifs aux modèles AWS IoT Events d'alarme.

Pour plus d'informations, consultez la section [Politiques basées sur les ressources](#) dans le guide de l'utilisateur IAM.

Autorisations d'action requises

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.

Avant de définir un modèle AWS IoT Events d'alarme, vous devez accorder les autorisations suivantes qui permettent d' AWS IoT SiteWise envoyer les valeurs des propriétés des actifs au modèle d'alarme.

- `iotsitewise:DescribeAssetModel`, `iotsitewise:ListAssetModels` — Permet AWS IoT Events de vérifier si une propriété d'actif existe.
- `iotsitewise:UpdateAssetModelPropertyRouting`— Permet AWS IoT SiteWise de créer automatiquement des abonnements permettant AWS IoT SiteWise d'envoyer des données à AWS IoT Events.

Pour plus d'informations sur les actions AWS IoT SiteWise prises en charge, consultez la section [Actions définies par AWS IoT SiteWise](#) dans la référence d'autorisation de service.

Exemple Exemple de politique d'autorisation 1

La politique suivante permet d' AWS IoT SiteWise envoyer les valeurs des propriétés des actifs à n'importe quel modèle AWS IoT Events d'alarme.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Exemple Exemple de politique d'autorisation 2

La politique suivante permet d' AWS IoT SiteWise envoyer les valeurs d'une propriété d'actif spécifiée à un modèle AWS IoT Events d'alarme spécifié.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:UpdateAlarmModel"
      ],
      "Resource": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModelPropertyRouting"
      ],
      "Resource": [
        "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/12345678-90ab-cdef-1234-567890abcdef"
      ],
      "Condition": {
        "StringLike": {
          "iotsitewise:propertyId": "abcdef12-3456-7890-abcd-ef1234567890",
          "iotevents:alarmModelArn": "arn:aws:iotevents:us-east-1:123456789012:alarmModel/MyAlarmModel"
        }
      }
    }
  ]
}
```

ListInputRoutings Autorisation (facultative)

Lorsque vous mettez à jour ou supprimez un modèle d'actif, vous AWS IoT SiteWise pouvez vérifier si un modèle d'alarme AWS IoT Events surveille une propriété d'actif associée à ce modèle

d'actif. Cela vous empêche de supprimer une propriété de ressource actuellement AWS IoT Events utilisée par une alarme. Pour activer cette fonctionnalité dans AWS IoT SiteWise, vous devez avoir l'`iotevents:ListInputRoutings` autorisation. Cette autorisation permet AWS IoT SiteWise d'appeler l'opération d'[ListInputRoutings](#) API prise en charge par AWS IoT Events.

Note

Nous vous recommandons vivement d'ajouter cette `ListInputRoutings` autorisation.

Exemple Exemple de politique d'autorisation

La politique suivante vous permet de mettre à jour et de supprimer des modèles d'actifs, ainsi que d'utiliser l'`ListInputRoutings` API dans AWS IoT SiteWise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:UpdateAssetModel",
        "iotsitewise>DeleteAssetModel",
        "iotevents:ListInputRoutings"
      ],
      "Resource": "arn:aws:iotsitewise:us-east-1:123456789012:asset-model/*"
    }
  ]
}
```

Autorisations requises pour SiteWise Monitor

Si vous souhaitez utiliser la fonctionnalité d'alarmes dans les portails de SiteWise surveillance, vous devez mettre à jour le [rôle de service de SiteWise surveillance](#) avec la politique suivante :

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

```

```

        "iotsitewise:CreateProject",
        "iotsitewise:DescribeProject",
        "iotsitewise:UpdateProject",
        "iotsitewise>DeleteProject",
        "iotsitewise:ListProjects",
        "iotsitewise:BatchAssociateProjectAssets",
        "iotsitewise:BatchDisassociateProjectAssets",
        "iotsitewise:ListProjectAssets",
        "iotsitewise:CreateDashboard",
        "iotsitewise:DescribeDashboard",
        "iotsitewise:UpdateDashboard",
        "iotsitewise>DeleteDashboard",
        "iotsitewise:ListDashboards",
        "iotsitewise:CreateAccessPolicy",
        "iotsitewise:DescribeAccessPolicy",
        "iotsitewise:UpdateAccessPolicy",
        "iotsitewise>DeleteAccessPolicy",
        "iotsitewise:ListAccessPolicies",
        "iotsitewise:DescribeAsset",
        "iotsitewise:ListAssets",
        "iotsitewise:ListAssociatedAssets",
        "iotsitewise:DescribeAssetProperty",
        "iotsitewise:GetAssetPropertyValue",
        "iotsitewise:GetAssetPropertyValueHistory",
        "iotsitewise:GetAssetPropertyAggregates",
        "iotsitewise:BatchPutAssetPropertyValue",
        "iotsitewise:ListAssetRelationships",
        "iotsitewise:DescribeAssetModel",
        "iotsitewise:ListAssetModels",
        "iotsitewise:UpdateAssetModel",
        "iotsitewise:UpdateAssetModelPropertyRouting",
        "sso-directory:DescribeUsers",
        "sso-directory:DescribeUser",
        "iotevents:DescribeAlarmModel",
        "iotevents:ListTagsForResource"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:BatchAcknowledgeAlarm",
        "iotevents:BatchSnoozeAlarm",
        "iotevents:BatchEnableAlarm",

```

```

        "iotevents:BatchDisableAlarm"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "iotevents:keyValue": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:CreateAlarmModel",
        "iotevents:TagResource"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:RequestTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iotevents:UpdateAlarmModel",
        "iotevents>DeleteAlarmModel"
    ],
    "Resource": "*",
    "Condition": {
        "Null": {
            "aws:ResourceTag/iotsitewisemonitor": "false"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "iam:PassRole"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "iam:PassedToService": [

```



```
    "iotevents.amazonaws.com"
  ]
}
}
```

Prévention interservices confuse des adjoints dans AWS IoT SiteWise

Le problème de délégué confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. En AWS, l'usurpation d'identité interservices peut entraîner la confusion des adjoints. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous recommandons d'utiliser les clés de contexte de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) et les clés contextuelles dans les politiques de ressources afin de limiter les autorisations qui AWS IoT SiteWise accordent un autre service à la ressource. Si la valeur `aws:SourceArn` ne contient pas l'ID de compte, tel que l'Amazon Resource Name (ARN) d'un compartiment Amazon S3, vous devez utiliser les deux clés de contexte de condition globale pour limiter les autorisations. Si vous utilisez les deux clés de contexte de condition globale et que la valeur `aws:SourceArn` contient l'ID de compte, la valeur `aws:SourceAccount` et le compte dans la valeur `aws:SourceArn` doivent utiliser le même ID de compte lorsqu'ils sont utilisés dans la même instruction de politique.

- Utilisez `aws:SourceArn` si vous souhaitez qu'une seule ressource soit associée à l'accès entre services.
- Utilisez `aws:SourceAccount` si vous souhaitez autoriser l'association d'une ressource de ce compte à l'utilisation interservices.

La valeur de `aws:SourceArn` doit être la ressource AWS IoT SiteWise client associée à la `sts:AssumeRole` demande.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename:*:123456789012:*`.

Exemple — Prévention adjointe confuse

L'exemple suivant montre comment vous pouvez utiliser les touches de contexte de condition `aws:SourceAccount` globale `aws:SourceArn` et globale AWS IoT SiteWise pour éviter le problème de confusion des adjoints.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotsitewise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Resource": [
      "arn:aws:iotsitewise::ResourceName/*"
    ],
    "Condition": {
      "ArnLike": {
        "aws:SourceArn": "arn:aws:iotsitewise:*:123456789012:*"
      },
      "StringEquals": {
        "aws:SourceAccount": "123456789012"
      }
    }
  }
}
```

Résoudre les problèmes d' AWS IoT SiteWise 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 SiteWise et AWS Identity and Access Management (IAM).

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans AWS IoT SiteWise](#)
- [Je ne suis pas autorisé à effectuer iam:PassRole](#)
- [Je souhaite autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS IoT SiteWise ressources](#)

Je ne suis pas autorisé à effectuer une action dans AWS IoT SiteWise

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni votre nom d'utilisateur et votre mot de passe.

L'exemple d'erreur suivant se produit lorsque l'utilisateur mateojackson IAM essaie d'utiliser la console pour afficher les détails d'un actif mais ne dispose pas des `iotsitewise:DescribeAsset` autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
    iotsitewise:DescribeAsset on resource: a1b2c3d4-5678-90ab-cdef-2222EXAMPLE
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à la ressource d'actif avec l'ID `a1b2c3d4-5678-90ab-cdef-2222EXAMPLE` à l'aide de l'action `iotsitewise:DescribeAsset`.

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 SiteWise.

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, un utilisateur doit 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 SiteWise. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary ne dispose pas des autorisations nécessaires pour transférer 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 autoriser des personnes extérieures à mon AWS compte à accéder à mes AWS IoT SiteWise 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 SiteWise en charge, consultez [Comment AWS IoT SiteWise 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.

Validation de conformité pour AWS IoT SiteWise

AWS IoT SiteWise ne fait l'objet d'aucun programme de AWS conformité.

Pour une liste des AWS services concernés par des programmes de conformité spécifiques, voir [AWS Services concernés par programme de conformité AWS](#) . 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 AWS IoT SiteWise est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides [de démarrage rapide sur la sécurité et la conformité](#) Guides sur la sécurité et la conformité — Ces guides de déploiement abordent les considérations architecturales et fournissent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- Livre blanc [sur l'architecture pour la sécurité et la conformité HIPAA](#) — Ce livre blanc décrit comment les entreprises peuvent créer des applications conformes à la loi HIPAA. AWS
- AWS Ressources de <https://aws.amazon.com/compliance/resources/> de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [Évaluation des ressources à l'aide des règles](#) énoncées dans le guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#) — Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.
- [Dix règles d'or en matière de sécurité pour les solutions IoT industrielles](#) — Ce billet de blog présente dix règles d'or qui vous aident à sécuriser vos systèmes de contrôle industriel (ICS), l'Internet des objets (IIoT) industriel et vos environnements cloud.
- [Meilleures pratiques de sécurité pour l'OT dans le secteur manufacturier](#) : ce livre blanc décrit les meilleures pratiques de sécurité pour concevoir, déployer et structurer ces charges de travail de fabrication hybrides sur site pour le cloud. AWS

Résilience dans AWS IoT SiteWise

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. Les régions 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.

AWS IoT SiteWise est entièrement géré et utilise des AWS services hautement disponibles et durables, tels qu'Amazon S3 et Amazon EC2. Pour garantir la disponibilité en cas d'interruption de la zone de disponibilité, AWS IoT SiteWise fonctionne sur plusieurs zones de disponibilité.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section [Infrastructure AWS mondiale](#).

Outre l'infrastructure AWS mondiale, AWS IoT SiteWise propose plusieurs fonctionnalités pour répondre à vos besoins en matière de résilience et de sauvegarde des données :

- Vous pouvez publier des mises à jour de la valeur des propriétés par le AWS IoT Core biais de messages MQTT, puis configurer des règles pour agir sur ces données. Grâce à cette fonctionnalité, vous pouvez sauvegarder des données dans d'autres AWS services tels qu'Amazon S3 et Amazon DynamoDB. Pour plus d'informations, consultez [Interagissez avec d'autres AWS services](#) et [Exportez des données vers Amazon S3 avec des notifications relatives aux propriétés des actifs](#).
- Vous pouvez utiliser le AWS IoT SiteWise Get * APIs pour récupérer et sauvegarder les données historiques des propriétés des actifs. Pour de plus amples informations, veuillez consulter [Rechercher les valeurs historiques des propriétés des actifs dans AWS IoT SiteWise](#).
- Vous pouvez utiliser le AWS IoT SiteWise Describe * APIs pour récupérer les définitions de vos ressources, telles que les actifs et les modèles. Vous pouvez sauvegarder ces définitions et les utiliser ultérieurement pour recréer vos ressources. Pour plus d'informations, consultez la page [Référence de l'API AWS IoT SiteWise](#).

Sécurité de l'infrastructure dans AWS IoT SiteWise

En tant que service géré, AWS IoT SiteWise il est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section [Sécurité du AWS cloud](#). Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section [Protection de l'infrastructure](#) dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder AWS IoT SiteWise via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). 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.

SiteWise Les passerelles Edge, qui s'exécutent sur AWS IoT Greengrass, utilisent des certificats X.509 et des clés cryptographiques pour se connecter et s'authentifier dans le cloud. AWS Pour plus d'informations, consultez la section [Authentification et autorisation de l'appareil AWS IoT Greengrass](#) dans le guide du AWS IoT Greengrass Version 1 développeur.

Analyse de configuration et de vulnérabilité dans AWS IoT SiteWise

Les flottes IoT sont composées d'un grand nombre d'appareils disposant de capacités diverses, d'une durée de vie longue et qui sont répartis géographiquement. Ces caractéristiques peuvent rendre la configuration des flottes complexe et source d'erreurs. Étant donné que les appareils ont généralement une puissance de traitement, une mémoire et un stockage limités, ils ne peuvent pas toujours prendre en charge le chiffrement et les autres mesures de sécurité. En outre, les appareils utilisent souvent des logiciels aux vulnérabilités connues. Ces facteurs font des parcs IoT une cible attractive pour les pirates informatiques et rend difficile la sécurisation de votre parc sur une base permanente.

AWS IoT Device Defender répond à ces défis en fournissant des outils permettant d'identifier les problèmes de sécurité et les écarts par rapport aux meilleures pratiques. Utilisez-le AWS IoT Device Defender pour analyser, auditer et surveiller les appareils connectés afin de détecter les comportements anormaux et d'atténuer les risques de sécurité. AWS IoT Device Defender peut auditer les flottes d'appareils pour s'assurer qu'elles respectent les meilleures pratiques de sécurité et détecter les comportements anormaux sur les appareils. Cela permet d'appliquer des politiques de sécurité cohérentes à l'ensemble de votre parc d' AWS IoT appareils et de réagir rapidement lorsque des appareils sont compromis. Pour plus d'informations, veuillez consulter la rubrique [Présentation de AWS IoT Device Defender](#) dans le Guide du développeur AWS IoT Device Defender .

Si vous utilisez des passerelles SiteWise Edge pour ingérer des données vers le service, il est de votre responsabilité de configurer et de gérer l'environnement de votre passerelle SiteWise Edge. Cette responsabilité inclut la mise à niveau vers les dernières versions du logiciel système, AWS IoT Greengrass du logiciel et du AWS IoT SiteWise connecteur de la passerelle SiteWise Edge. Pour plus d'informations, voir [Configurer le AWS IoT Greengrass noyau](#) dans le guide du AWS IoT Greengrass Version 1 développeur et [Gérer les passerelles SiteWise Edge](#).

Points de terminaison VPC pour AWS IoT SiteWise

Un point de terminaison VPC d'interface établit une connexion privée entre votre cloud privé virtuel (VPC) et. AWS IoT SiteWise [AWS PrivateLink](#) alimente les points de terminaison de l'interface, permettant un accès privé aux opérations de AWS IoT SiteWise l'API. Vous pouvez éviter d'avoir besoin d'une passerelle Internet, d'un appareil NAT, d'une connexion VPN ou AWS Direct Connect. Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec les opérations d' AWS IoT SiteWise API. Le trafic entre votre VPC et celui qui AWS IoT SiteWise ne quitte pas le AWS réseau.

Chaque point de terminaison d'interface est représenté par une ou plusieurs [interfaces réseau Elastic](#) dans vos sous-réseaux.

Avant de configurer un point de terminaison VPC d'interface pour AWS IoT SiteWise, consultez la section [Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface](#) dans le Guide.AWS PrivateLink

Opérations d'API pour les points de terminaison VPC dans AWS IoT SiteWise

AWS IoT SiteWise permet d'appeler les opérations d' AWS IoT SiteWise API suivantes depuis votre VPC :

- Pour toutes les opérations de l'API du plan de données, utilisez le point de terminaison suivant : Remplacez *region* par votre Région AWS

```
data.iotsitewise.region.amazonaws.com
```

Les opérations de l'API du plan de données sont les suivantes :

- [BatchGetAssetPropertyValue](#)
- [BatchGetAssetPropertyValueHistory](#)
- [BatchPutAssetPropertyValue](#)
- [GetAssetPropertyAggregates](#)
- [GetAssetPropertyValue](#)
- [GetAssetPropertyValueHistory](#)
- [GetInterpolatedAssetPropertyValues](#)
- Pour les opérations d'API du plan de contrôle que vous utilisez pour gérer les modèles d'actifs, les actifs, les passerelles SiteWise Edge, les balises et les configurations de compte, utilisez le point de terminaison suivant. *region* Remplacez-le par votre Région AWS.

```
api.iotsitewise.region.amazonaws.com
```

Les opérations d'API du plan de contrôle prises en charge sont les suivantes :

- [AssociateAssets](#)
- [CreateAsset](#)
- [CreateAssetModel](#)
- [DeleteAsset](#)
- [DeleteAssetModel](#)
- [DeleteDashboard](#)

- [DescribeAsset](#)

- [DescribeAssetModel](#)
- [DescribeAssetProperty](#)
- [DescribeDashboard](#)
- [DescribeLoggingOptions](#)
- [DisassociateAssets](#)
- [ListAssetModels](#)
- [ListAssetRelationships](#)
- [ListAssets](#)
- [ListAssociatedAssets](#)
- [PutLoggingOptions](#)
- [UpdateAsset](#)
- [UpdateAssetModel](#)
- [UpdateAssetProperty](#)
- [CreateGateway](#)
- [DeleteGateway](#)
- [DescribeDefaultEncryptionConfiguration](#)
- [DescribeGateway](#)
- [DescribeGatewayCapabilityConfiguration](#)
- [DescribeStorageConfiguration](#)
- [ListGateways](#)
- [ListTagsForResource](#)
- [UpdateGateway](#)
- [UpdateGatewayCapabilityConfiguration](#)
- [PutDefaultEncryptionConfiguration](#)
- [PutStorageConfiguration](#)
- [TagResource](#)
- [UntagResource](#)

 Note

Le point de terminaison VPC de l'interface pour les opérations d'API du plan de contrôle ne prend actuellement pas en charge les appels aux opérations de l'API SiteWise Monitor suivantes :

- [BatchAssociateProjectAssets](#)
- [BatchDisassociateProjectAssets](#)
- [CreateAccessPolicy](#)
- [CreateDashboard](#)
- [CreatePortal](#)
- [CreateProject](#)
- [DeleteAccessPolicy](#)
- [DeletePortal](#)
- [DeleteProject](#)
- [DescribeAccessPolicy](#)
- [DescribePortal](#)
- [DescribeProject](#)
- [ListAccessPolicies](#)
- [ListDashboards](#)
- [ListPortals](#)
- [ListProjects](#)
- [ListProjectAssets](#)
- [UpdateAccessPolicy](#)
- [UpdateDashboard](#)
- [UpdatePortal](#)
- [UpdateProject](#)

Créer un point de terminaison de VPC d'interface pour AWS IoT SiteWise

Pour créer un point de terminaison VPC pour le AWS IoT SiteWise service, utilisez la console

[Amazon VPC](#) ou le [\(\) AWS Command Line Interface AWS CLI](#) Pour plus d'informations, consultez

la section [Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface](#) dans le AWS PrivateLink Guide.

Créez un point de terminaison VPC pour AWS IoT SiteWise en utilisant l'un des noms de service suivants :

- Pour les opérations de l'API du plan de données, utilisez le nom de service suivant :

```
com.amazonaws.region.iotsitewise.data
```

- Pour les opérations de l'API du plan de contrôle, utilisez le nom de service suivant :

```
com.amazonaws.region.iotsitewise.api
```

Accès AWS IoT SiteWise via un point de terminaison VPC d'interface

Lorsque vous créez un point de terminaison d'interface, nous générons des noms d'hôte DNS spécifiques au point de terminaison avec lesquels vous pouvez communiquer. AWS IoT SiteWise 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 SiteWise 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 Région AWS.

```
data.iotsitewise.region.amazonaws.com
```

- Pour les opérations de l'API du plan de contrôle, utilisez le point de terminaison suivant : Remplacez *region* par votre Région AWS.

```
api.iotsitewise.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 SiteWise via le point de terminaison :

1. 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.iotsitewise.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.iotsitewise.region.vpce.amazonaws.com
```

2. Désactive l'injection de préfixe d'hôte. Le AWS CLI et AWS SDKs ajoutez différents préfixes d'hôte au point de terminaison du service lorsque vous appelez chaque opération d'API. Cette fonctionnalité entraîne la production de AWS CLI données et AWS SDKs URLs qui ne sont pas valides AWS IoT SiteWise lorsque vous spécifiez un point de terminaison VPC.

Important

Vous ne pouvez pas désactiver l'injection de préfixe d'hôte dans le AWS CLI ou le Outils AWS pour PowerShell. Cela signifie que si vous désactivez le DNS privé, vous ne pouvez pas utiliser ces outils pour accéder AWS IoT SiteWise via le point de terminaison VPC. Activez le DNS privé pour utiliser AWS CLI ou Outils AWS pour PowerShell pour accéder AWS IoT SiteWise via le point de terminaison.

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 la section [Accès à un AWS service à l'aide d'un point de terminaison VPC d'interface](#) dans le AWS PrivateLink Guide.

Créez une politique de point de terminaison VPC pour AWS IoT SiteWise

Vous pouvez attacher une stratégie de point de terminaison à votre point de terminaison d'un VPC qui contrôle l'accès à AWS IoT SiteWise. La politique spécifie les informations suivantes :

- Le principal qui peut effectuer des opérations.
- Les opérations qui peuvent être effectuées.
- Les ressources sur lesquelles les opérations peuvent être effectuées.

Pour plus d'informations, consultez [Contrôle de l'accès aux points de terminaison d'un VPC à l'aide de politiques de point de terminaison](#) dans le Guide de l'utilisateur Amazon VPC.

Exemple : politique de point de terminaison VPC pour les actions AWS IoT SiteWise

Voici un exemple de politique de point de terminaison pour AWS IoT SiteWise. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux AWS IoT SiteWise actions répertoriées *iotsitewiseadmin* à AWS l'utilisateur *123456789012* responsable de l'actif spécifié.

```
{
  "Statement": [
    {
      "Action": [
        "iotsitewise:CreateAsset",
        "iotsitewise:ListGateways",
        "iotsitewise:ListTagsForResource"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iotsitewise:us-west-2:123456789012:asset/a1b2c3d4-5678-90ab-cdef-33333EXAMPLE",
      "Principal": {
        "AWS": [
          "123456789012:user/iotsitewiseadmin"
        ]
      }
    }
  ]
}
```

```
}  
  }  
] }  
}
```

Bonnes pratiques de sécurité pour AWS IoT SiteWise

Cette rubrique présente les meilleures pratiques de sécurité pour AWS IoT SiteWise.

Utilisez les informations d'authentification sur vos serveurs OPC UA

Exigez des informations d'authentification pour vous connecter à vos serveurs OPC UA. Consultez la documentation de vos serveurs pour le faire. Ensuite, pour permettre à votre passerelle SiteWise Edge de se connecter à vos serveurs OPC UA, ajoutez des secrets d'authentification du serveur à votre passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Configurer l'authentification des sources de données pour SiteWise Edge](#).

Utilisez des modes de communication cryptés pour vos serveurs OPC UA

Choisissez un mode de sécurité des messages chiffrés et non obsolète lorsque vous configurez vos sources OPC UA pour votre passerelle Edge. SiteWise Cela permet de sécuriser vos données industrielles lorsqu'elles sont transférées de vos serveurs OPC UA vers la passerelle SiteWise Edge. Pour plus d'informations, consultez [Données en transit sur le réseau local](#) et [Configurer une source OPC UA dans Edge SiteWise](#).

Maintenez vos composants à jour

Si vous utilisez des passerelles SiteWise Edge pour ingérer des données vers le service, il est de votre responsabilité de configurer et de gérer l'environnement de votre passerelle SiteWise Edge. Cette responsabilité inclut la mise à niveau vers les dernières versions du logiciel système, AWS IoT Greengrass des logiciels et des connecteurs de la passerelle.

Note

Le connecteur AWS IoT SiteWise Edge stocke les secrets sur votre système de fichiers. Ces secrets contrôlent qui peut consulter les données mises en cache dans votre passerelle SiteWise Edge. Il est vivement recommandé d'activer le chiffrement du disque ou du système de fichiers pour le système exécutant votre passerelle SiteWise Edge.

Pour plus d'informations sur la mise à niveau des composants de la AWS IoT SiteWise console, consultez [Modifier la version des packs de composants de la passerelle SiteWise Edge](#).

Chiffrez le système de fichiers de votre passerelle SiteWise Edge

Chiffrez et sécurisez votre passerelle SiteWise Edge afin que vos données industrielles soient sécurisées lorsqu'elles passent par la passerelle SiteWise Edge. Si votre passerelle SiteWise Edge possède un module de sécurité matériel, vous pouvez le configurer AWS IoT Greengrass pour sécuriser votre passerelle SiteWise Edge. Pour plus d'informations, consultez la section [Intégration de la sécurité matérielle](#) dans le Guide du AWS IoT Greengrass Version 1 développeur. Sinon, consultez la documentation de votre système d'exploitation pour savoir comment chiffrer et sécuriser votre système de fichiers.

Accès sécurisé à votre configuration Edge

Ne partagez pas le mot de passe de l'application Edge Console ou celui de l'application SiteWise Monitor. Ne mettez pas ce mot de passe dans un endroit où tout le monde peut le voir. Mettez en œuvre une politique saine de rotation des mots de passe en configurant une date d'expiration appropriée pour votre mot de passe.

Sécurisation des données sur Siemens Industrial Edge Management

Les données de l'appareil que vous choisissez de partager avec AWS IoT SiteWise Edge sont définies dans vos rubriques Siemens IEM Databus de configuration. En sélectionnant des sujets à partager avec SiteWise Edge, vous partagez des données thématiques avec. AWS IoT SiteWiseSiemens Industrial Edge Marketplace s'agit d'un marché indépendant, distinct de AWS. Pour protéger vos données partagées, l'application SiteWise Edge ne s'exécutera que si vous l'utilisez Siemens Secured Storage. Pour plus d'informations, voir [Secure Storage](#) dans Siemens la documentation.

Accorder aux utilisateurs de SiteWise Monitor les autorisations minimales possibles

Respectez le principe du moindre privilège en utilisant l'ensemble minimal d'autorisations de politique d'accès pour les utilisateurs de votre portail.

- Lorsque vous créez un portail, définissez un rôle qui autorise l'ensemble minimal d'actifs requis pour ce portail. Pour de plus amples informations, veuillez consulter [Utiliser les rôles de service pour AWS IoT SiteWise Monitor](#).

- Lorsque vous et les administrateurs de votre portail créez et partagez des projets, utilisez l'ensemble minimal d'actifs requis pour ce projet.
- Lorsqu'une identité n'a plus besoin d'accéder à un portail ou à un projet, supprimez-la de cette ressource. Si cette identité n'est plus applicable à votre organisation, supprimez-la de votre banque d'identités.

La meilleure pratique du moindre principe s'applique également aux rôles IAM. Pour de plus amples informations, veuillez consulter [Bonnes pratiques en matière de politiques](#).

Ne pas exposer d'informations sensibles

Vous devez empêcher l'enregistrement des informations d'identification et d'autres informations sensibles, telles que les informations personnellement identifiables (PII). Nous vous recommandons de mettre en œuvre les mesures de protection suivantes, même si l'accès aux journaux locaux sur une passerelle SiteWise Edge nécessite des privilèges root et que l'accès aux CloudWatch journaux nécessite des autorisations IAM.

- N'utilisez pas d'informations sensibles dans les noms, descriptions ou propriétés de vos ressources ou modèles.
- N'utilisez pas d'informations sensibles dans les noms de source ou de passerelle SiteWise Edge.
- N'utilisez pas d'informations sensibles dans les noms ou descriptions de vos portails, projets ou tableaux de bord.

Suivez les meilleures pratiques en matière de AWS IoT Greengrass sécurité

Suivez les meilleures pratiques de AWS IoT Greengrass sécurité pour votre passerelle SiteWise Edge. Pour plus d'informations, consultez [la section Bonnes pratiques en matière de sécurité](#) dans le guide du AWS IoT Greengrass Version 1 développeur.

Consultez aussi

- [Bonnes pratiques de sécurité](#) décrites dans le guide du AWS IoT développeur
- [Dix règles d'or en matière de sécurité pour les solutions IoT industrielles](#)

Connectez-vous et surveillez AWS IoT SiteWise

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité AWS IoT SiteWise et des performances de vos autres AWS solutions. AWS IoT SiteWise 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 vos AWS ressources et les applications que vous utilisez AWS en temps réel. Collectez et suivez les métriques, créez des tableaux de bord personnalisés et définissez des alarmes qui vous avertissent ou prennent des mesures lorsqu'une métrique spécifiée atteint un certain seuil. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos EC2 instances Amazon 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 accède à vos fichiers journaux à partir de passerelles SiteWise Edge et d'autres sources. CloudTrail 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 connexes effectués par ou au nom de votre AWS compte. CloudTrail Transmet ensuite 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 [AWS CloudTrail Guide de l'utilisateur](#).

Rubriques

- [Surveillez avec Amazon CloudWatch Logs](#)
- [Surveiller les journaux de la passerelle SiteWise Edge](#)
- [Surveillez AWS IoT SiteWise avec Amazon CloudWatch Metrics](#)
- [Enregistrez les appels AWS IoT SiteWise d'API avec AWS CloudTrail](#)

Surveillez avec Amazon CloudWatch Logs

Configurez AWS IoT SiteWise pour consigner les informations dans CloudWatch les journaux afin de surveiller et de dépanner le service.

Lorsque vous utilisez la AWS IoT SiteWise console, AWS IoT SiteWise crée un rôle lié au service qui permet au service de consigner des informations en votre nom. Si vous n'utilisez pas la AWS IoT SiteWise console, vous devez créer manuellement un rôle lié à un service pour recevoir les journaux. Pour de plus amples informations, veuillez consulter [Créer un rôle lié à un service pour AWS IoT SiteWise](#).

Vous devez disposer d'une politique de ressources permettant de AWS IoT SiteWise placer les événements du journal dans des CloudWatch flux. Pour créer et mettre à jour une politique de ressources pour CloudWatch les journaux, exécutez la commande suivante. Remplacez *logging-policy-name* par le nom de la politique à créer.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\" } ] }"
```

CloudWatch Logs prend également en charge les clés contextuelles de SourceAccount condition [aws : SourceArn](#) et [aws :](#). Ces clés de contexte de condition sont facultatives.

Pour créer ou mettre à jour une politique de ressources qui AWS IoT SiteWise permet de placer uniquement les journaux associés à la AWS IoT SiteWise ressource spécifiée dans des CloudWatch flux, exécutez la commande et procédez comme suit :

- Remplacez *logging-policy-name* par le nom de la politique à créer.
- *source-ARN* Remplacez-le par l'ARN de votre AWS IoT SiteWise ressource, tel qu'un modèle d'actif ou un actif. Pour trouver l'ARN de chaque type de AWS IoT SiteWise ressource, consultez la section [Types de ressources définis par AWS IoT SiteWise](#) dans la référence d'autorisation de service.
- Remplacez *account-ID* par l'ID de AWS compte associé à la AWS IoT SiteWise ressource spécifiée.

```
aws logs put-resource-policy --policy-name logging-policy-name --policy-document "{ \"Version\": \"2012-10-17\", \"Statement\": [ { \"Sid\": \"IoTSiteWiseToCloudWatchLogs\", \"Effect\": \"Allow\", \"Principal\": { \"Service\": [ \"iotsitewise.amazonaws.com\" ] }, \"Action\": \"logs:PutLogEvents\", \"Resource\": \"*\", \"Condition\": { \"StringLike\": { \"aws:SourceArn\": [ \"source-ARN\" ], \"aws:SourceAccount\": [ \"account-ID\" ] } } } ] }
```

Par défaut, AWS IoT SiteWise n'enregistre pas les informations dans CloudWatch Logs. Pour activer la journalisation, choisissez un niveau de journalisation autre que Disabled (OFF). AWS IoT SiteWise prend en charge les niveaux de journalisation suivants :

- OFF— La journalisation est désactivée.
- ERROR— Les erreurs sont enregistrées.
- INFO— Les erreurs et les messages d'information sont enregistrés.

Vous pouvez configurer les passerelles SiteWise Edge pour enregistrer les informations dans CloudWatch Logs through AWS IoT Greengrass. Pour de plus amples informations, veuillez consulter [Surveiller les journaux de la passerelle SiteWise Edge](#).

Vous pouvez également configurer AWS IoT Core pour consigner les informations dans les CloudWatch journaux si vous dépannez une action de AWS IoT SiteWise règle. Pour de plus amples informations, veuillez consulter [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#).

Table des matières

- [Gérer la connexion AWS IoT SiteWise](#)
 - [Trouvez votre niveau de journalisation](#)
 - [Modifier votre niveau de journalisation](#)
- [Exemple : entrées de fichier AWS IoT SiteWise journal](#)

Gérer la connexion AWS IoT SiteWise

Utilisez la AWS IoT SiteWise console ou AWS CLI pour les tâches de configuration de journalisation suivantes.

Trouvez votre niveau de journalisation

Console

Suivez la procédure suivante pour trouver votre niveau de journalisation actuel dans la console AWS IoT SiteWise .

Pour trouver votre niveau de AWS IoT SiteWise journalisation actuel

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le panneau de navigation de gauche, choisissez Options de journalisation.

L'état de journalisation actuel apparaît sous Statut de journalisation. Si la journalisation est activée, le niveau de journalisation actuel apparaît sous Niveau de verbosité.

AWS CLI

Exécutez la commande suivante pour trouver votre niveau de AWS IoT SiteWise journalisation actuel avec le AWS CLI.

```
aws iotsitewise describe-logging-options
```

L'opération renvoie une réponse qui contient votre niveau de journalisation au format suivant.

```
{
  "loggingOptions": {
    "level": "String"
  }
}
```

Modifier votre niveau de journalisation

Utilisez la procédure suivante pour modifier votre niveau de journalisation dans la AWS IoT SiteWise console ou en utilisant AWS CLI.

Console

Pour modifier votre niveau de AWS IoT SiteWise journalisation

1. Accédez à la [console AWS IoT SiteWise](#).

2. Dans le panneau de navigation de gauche, choisissez Options de journalisation.
3. Choisissez Modifier.
4. Choisissez le niveau de verbosité à activer.
5. Choisissez Save (Enregistrer).

AWS CLI

Exécutez la AWS CLI commande suivante pour modifier votre niveau de AWS IoT SiteWise journalisation. *logging-level* Remplacez-le par le niveau de journalisation que vous souhaitez.

```
aws iotsitewise put-logging-options --logging-options level=logging-level
```

Exemple : entrées de fichier AWS IoT SiteWise journal

Chaque entrée de AWS IoT SiteWise journal inclut des informations sur l'événement et des ressources pertinentes pour cet événement, afin que vous puissiez comprendre et analyser les données du journal.

L'exemple suivant montre une entrée CloudWatch Logs qui AWS IoT SiteWise se connecte lorsque vous créez avec succès un modèle d'actif.

```
{
  "eventTime": "2020-05-05T00:10:22.902Z",
  "logLevel": "INFO",
  "eventType": "AssetModelCreationSuccess",
  "message": "Successfully created asset model.",
  "resources": {
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE"
  }
}
```

Surveiller les journaux de la passerelle SiteWise Edge

Vous pouvez configurer votre passerelle AWS IoT SiteWise Edge pour enregistrer les informations dans Amazon CloudWatch Logs ou dans le système de fichiers local.

Rubriques

- [Utiliser Amazon CloudWatch Logs](#)
- [Utiliser les connexions au service AWS IoT SiteWise](#)

Utiliser Amazon CloudWatch Logs

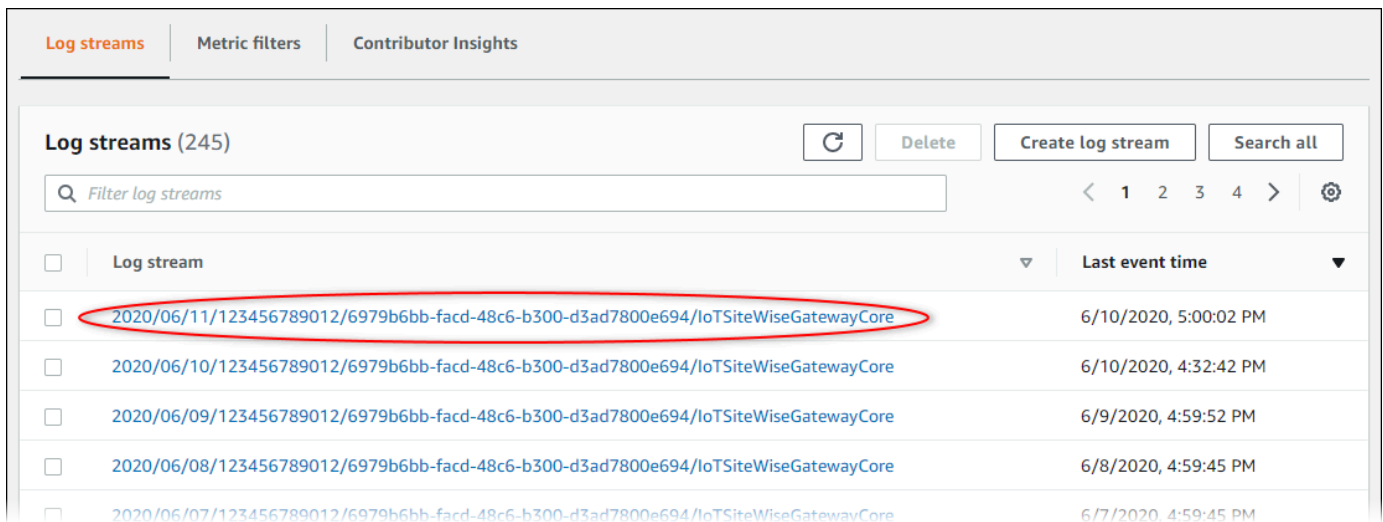
Vous pouvez configurer votre passerelle SiteWise Edge pour envoyer des CloudWatch journaux à Logs. Pour plus d'informations, voir [Activer la journalisation des CloudWatch journaux](#) dans le guide du AWS IoT Greengrass Version 2 développeur.

Pour configurer les CloudWatch journaux et y accéder (console)

1. Accédez à la [console CloudWatch](#).
2. Dans le panneau de navigation, choisissez Groupes de journaux.
3. Vous pouvez trouver les journaux des AWS IoT SiteWise composants dans les groupes de journaux suivants :
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgeCollectorOpcua`— Les journaux du composant de la passerelle SiteWise Edge qui collecte les données à partir des sources OPC UA de la passerelle SiteWise Edge.
 - `/aws/greengrass/UserComponent/region/aws.iot.SiteWiseEdgePublisher`— Les journaux du composant de la passerelle SiteWise Edge qui publie les flux de données OPC UA vers AWS IoT SiteWise.

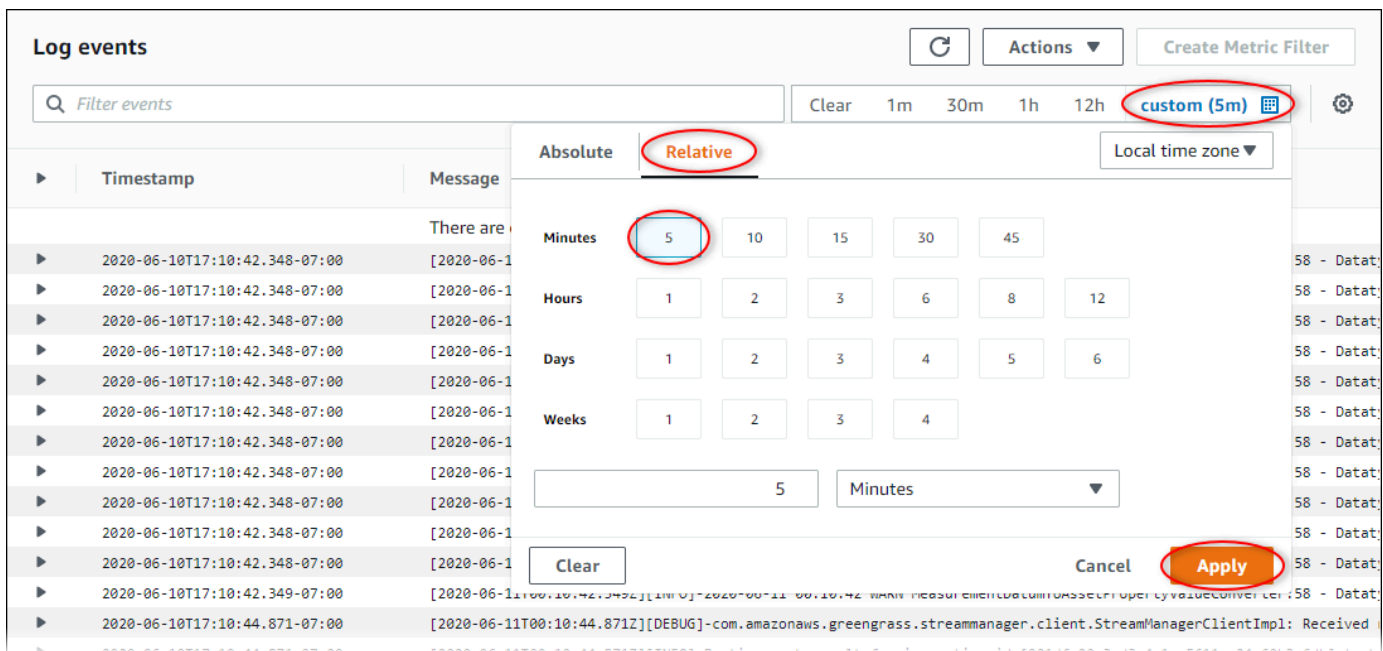
Choisissez le groupe de journaux pour la fonction à déboguer.

4. Choisissez un flux de journal dont le nom se termine par le nom de votre AWS IoT Greengrass groupe. Par défaut, CloudWatch affiche le flux de journal le plus récent en premier.



5. Pour afficher les journaux des 5 dernières minutes, procédez comme suit :

- Choisissez personnalisé dans le coin supérieur droit.
- Choisissez Relatif.
- Choisissez 5 minutes.
- Choisissez Appliquer.



- (Facultatif) Pour afficher moins de journaux, vous pouvez choisir 1 minute dans le coin supérieur droit.
- Faites défiler jusqu'au bas des entrées du journal pour afficher les journaux les plus récents.

Utiliser les connexions au service AWS IoT SiteWise

SiteWise Les périphériques Edge Gateway incluent des fichiers journaux de service pour aider à résoudre les problèmes. Les sections suivantes vous aideront à trouver et à utiliser les fichiers journaux de service pour les composants AWS IoT SiteWise OPC UA Collector et AWS IoT SiteWise Publisher.

AWS IoT SiteWise Fichier journal du service OPC UA Collector

Le composant AWS IoT SiteWise OPC UA Collector utilise le fichier journal suivant.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Pour consulter les journaux de ce composant

- Exécutez la commande suivante sur le périphérique principal pour afficher le fichier journal de ce composant en temps réel. Remplacez */greengrass/v2* ou *C:\greengrass\v2* par le chemin d'accès au dossier AWS IoT Greengrass racine.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgeCollectorOpcua.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgeCollectorOpcua.log -Tail  
10 -Wait
```

AWS IoT SiteWise Fichier journal du service Publisher

Le composant AWS IoT SiteWise Publisher utilise le fichier journal suivant.

Linux

```
/greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows

```
C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log
```

Pour consulter les journaux de ce composant

- Exécutez la commande suivante sur le périphérique principal pour afficher le fichier journal de ce composant en temps réel. Remplacez */greengrass/v2* ou *C:\greengrass\v2* par le chemin d'accès au dossier AWS IoT Greengrass racine.

Linux

```
sudo tail -f /greengrass/v2/logs/aws.iot.SiteWiseEdgePublisher.log
```

Windows (PowerShell)

```
Get-Content C:\greengrass\v2\logs\aws.iot.SiteWiseEdgePublisher.log -Tail 10 -Wait
```

Surveillez AWS IoT SiteWise avec Amazon CloudWatch Metrics

Vous pouvez surveiller AWS IoT SiteWise l'utilisation 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 SiteWise publie les métriques et les dimensions répertoriées dans les sections ci-dessous dans l'espace de AWS/IoTSiteWise noms.

 Tip

AWS IoT SiteWise publie des statistiques à une minute d'intervalle. Lorsque vous visualisez ces statistiques sous forme de graphiques dans la CloudWatch console, nous vous recommandons de choisir une période d'une minute. Cela vous permet de voir la résolution la plus élevée disponible de vos données de métrique.

Rubriques

- [AWS IoT Greengrass Version 2 métriques de passerelle](#)

AWS IoT Greengrass Version 2 métriques de passerelle

AWS IoT SiteWise publie des métriques de passerelle pour les flux classiques, les passerelles V2 et les passerelles V3 compatibles MQTT. Sauf indication contraire, chaque métrique s'applique aux deux versions de passerelle auto-hébergées. Toutes les métriques de la passerelle SiteWise Edge sont publiées à une minute d'intervalle.

SiteWise Mesures relatives à la passerelle Edge

Métrique	Description
Gateway.AvailableMemory	Mémoire disponible d'une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
Gateway.AvailableDiskSpace	L'espace disque disponible d'une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
Gateway.CloudConnectivity	État de connectivité cloud d'une passerelle SiteWise Edge. Unité : aucune

Métrique	Description
	Dimension : GatewayId
Gateway.CpuUsage	L'utilisation du processeur d'une passerelle SiteWise Edge. Unité : pourcentage Dimension : Aucune
Gateway.TotalDiskSpace	Espace disque total d'une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
Gateway.TotalMemory	Mémoire totale d'une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
Gateway.UsedDiskSpace	L'espace disque utilisé par une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
Gateway.UsedMemory	Mémoire utilisée d'une passerelle SiteWise Edge. Unité : octets Dimension : Aucune

Métrique	Description
<code>Gateway.UsedPercentageDiskSpace</code>	Pourcentage d'espace disque utilisé par une passerelle SiteWise Edge. Unité : octets Dimension : Aucune
<code>Gateway.UsedPercentageMemory</code>	Pourcentage de mémoire utilisé par une passerelle SiteWise Edge. Unité : octets Dimension : Aucune

AWS IoT SiteWise statistiques relatives aux éditeurs

Métrique	Description
<code>IoTSiteWisePublisher.ComponentBuildVersion</code>	Cette métrique indique la version de compilation du composant SiteWise éditeur IoT exécuté sur la passerelle. Une valeur de 1 signifie que la passerelle exécute une version de l'éditeur correspondant à la <code>ComponentBuildVersion</code> dimension. Unité : 1 Dimensions : <code>GatewayId</code>, <code>ComponentBuildVersion</code>
<code>IoTSiteWisePublisher.DroppedCount</code>	Nombre de points de données déposés par une passerelle SiteWise Edge (<code>GatewayId</code>) et non publiés dans le cloud, générés chaque minute. Unité : nombre

Métrique	Description
	Dimensions : GatewayId
<code>IoTSiteWisePublisher.Heartbeat</code>	Généré chaque minute par le Publisher dans la passerelle SiteWise Edge. Unité : 1 (1 représentant l'éditeur est en cours d'exécution et il manque le point de données indiquant que l'éditeur n'est pas en cours d'exécution.) Dimensions : GatewayId
<code>IoTSiteWisePublisher.IsConnectedToMqttBroker</code>	Généré chaque minute par le Publisher dans la passerelle SiteWise Edge. Unité : 1 (1 représentant l'éditeur est connecté à un broker MQTT.) Dimensions : GatewayId
<code>IoTSiteWisePublisher.MessageCheckpointPersistenceErrorCount</code>	La métrique indique que la passerelle a détecté un problème avec le fichier de point de contrôle utilisé pour suivre les données traitées par l'éditeur. La valeur de 1 signifie qu'une défaillance s'est produite. Unité : aucune Dimensions : AccountId, GatewayId
<code>IoTSiteWisePublisher.MqttMessageReceivedSuccessCount</code>	Le nombre de messages reçus avec succès par l'éditeur en provenance du broker MQTT, générés chaque minute. Unité : nombre Dimensions : GatewayId

Métrique	Description
<code>IoTSiteWisePublisher.MqttReceivedSuccessBytes</code>	Le nombre d'octets de données de message reçus avec succès par l'éditeur en provenance du broker MQTT, générés chaque minute. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.NumberOfSubscriptionsToMqttBroker</code>	Le nombre de sujets souscrits au broker MQTT par l'éditeur, généré chaque minute. Un sujet joker à plusieurs niveaux est compté pour 1. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.NumberOfUniqueMqttTopicsReceived</code>	Le nombre de sujets uniques reçus par l'éditeur par le courtier MQTT, générés chaque minute. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.PublisherFailureCount</code>	Le nombre de points de données qu'une passerelle SiteWise Edge (GatewayId) n'a pas réussi à publier, généré chaque minute. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.PublisherRejectedCount</code>	Le nombre de points de données rejetés par une passerelle SiteWise Edge (GatewayId) depuis le cloud, générés chaque minute. Unité : nombre Dimensions : GatewayId

Métrique	Description
<code>IoTSiteWisePublisher.PublisherSuccessCount</code>	Nombre de points de données qu'une passerelle SiteWise Edge (GatewayId) a publiés avec succès dans le cloud, générés chaque minute. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.PublisherToS3FailureCount</code>	Nombre de points de données qu'une passerelle (GatewayId) n'a pas réussi à publier dans un compartiment Amazon S3. Unité : nombre Dimensions : GatewayId
<code>IoTSiteWisePublisher.PublisherToS3SuccessCount</code>	Le nombre de points de données qu'une passerelle (GatewayId) a publiés avec succès dans un compartiment Amazon S3. Unité : nombre Dimensions : GatewayId

Métriques du collecteur OPC UA

Métrique	Description
<code>OpcUaCollector.ActiveDataStreamCount</code>	Nombre de flux de données auxquels une passerelle SiteWise Edge (gatewayId) s'est abonnée pour une source OPC UA (sourceName). Unité : nombre GatewayIdDimension1 : SourceName, PropertyGroup

Métrique	Description
<code>OpcUaCollector.ComponentBuildVersion</code> (non disponible sur les streams classiques, les passerelles V2)	Cette métrique indique la version de construction du composant collecteur IoT SiteWise OPC UA exécuté sur la passerelle. Une valeur de 1 signifie que la passerelle exécute une version du collecteur correspondant à la <code>ComponentBuildVersion</code> dimension. Unité : 1 Dimensions : <code>GatewayId</code>, <code>ComponentBuildVersion</code>
<code>OpcUaCollector.ConversionErrors</code>	Nombre de points de données reçus par une passerelle SiteWise Edge (<code>gatewayId</code>) pour une source OPC UA (<code>sourceName</code>) qui ont entraîné des erreurs de conversion lors de l'envoi des données. AWS IoT SiteWise Ces points de données ne seront pas ingérés par OPC UA Collector. Unité : nombre Dimensions : <code>GatewayId</code>, <code>SourceName</code>
<code>OpcUaCollector.Heartbeat</code>	Généré toutes les minutes pour chaque source OPC UA (<code>sourceName</code>) connectée à une passerelle SiteWise Edge (<code>gatewayId</code>). Unité : Nombre (1 représentant la source est connectée et 0 représentant la source est déconnectée.) Dimensions : <code>GatewayId</code>, <code>SourceName</code>

Métrique	Description
<code>OpcUaCollector.IncomingValuesCount</code>	Nombre de points de données qu'une passerelle SiteWise Edge (<code>gatewayId</code>) a reçus pour une source OPC UA (<code>sourceName</code>), générés chaque minute. Unité : nombre GatewayIdDimensioni : SourceName, PropertyGroup
<code>OpcUaCollector.IncomingValueErrors</code>	Nombre de points de données qu'une passerelle SiteWise Edge (<code>gatewayId</code>) reçoit d'une source OPC UA (<code>sourceName</code>) qui ne sont pas des valeurs valides. Ces points de données ne sont pas ingérés par le collecteur OPC UA, généré toutes les minutes. Unité : nombre GatewayIdDimensioni : SourceName, PropertyGroup
<code>OpcUaCollector.IsConnectedToMqttBroker</code> (non disponible sur les streams classiques, les passerelles V2)	Généré chaque minute par le composant collecteur IoT SiteWise OPC UA de la passerelle SiteWise Edge. Unité : 1 (1 représentant le composant collecteur IoT SiteWise OPC UA est connecté à un broker MQTT) Dimensions : GatewayId
<code>OpcUaCollector.MqttMessagesDroppedCount</code> (non disponible sur les streams classiques, les passerelles V2)	Nombre de messages MQTT déposés par le composant collecteur IoT SiteWise OPC UA. Unité : nombre Dimensions : GatewayId, SourceName

Métrique	Description
<code>OpcUaCollector.MqttMessagesPublishedBytes</code> (non disponible sur les streams classiques, les passerelles V2)	Nombre d'octets de données de message MQTT publiés avec succès par le composant collecteur IoT SiteWise OPC UA sur le broker MQTT. Unité : nombre Dimensions : GatewayId, SourceName
<code>OpcUaCollector.MqttMessagesPublishedCount</code> (non disponible sur les streams classiques, les passerelles V2)	Nombre de messages MQTT publiés avec succès par le composant collecteur IoT SiteWise OPC UA auprès du courtier MQTT. Unité : nombre Dimensions : GatewayId, SourceName
<code>OpcUaCollector.NullValueCount</code> (non disponible sur les streams classiques, les passerelles V2)	Nombre de valeurs nulles reçues par le composant collecteur IoT SiteWise OPC UA depuis le serveur OPC UA. Unité : nombre GatewayIdDimensions : SourceName, PropertyGroup
<code>OpcUaCollector.NumberOfUniqueMqttTopicsPublished</code> (non disponible sur les streams classiques, les passerelles V2)	Le nombre de sujets MQTT uniques publiés par le collecteur IoT SiteWise OPC UA à l'intention du courtier MQTT. Unité : nombre Dimensions : GatewayId, SourceName

AWS IoT SiteWise métriques du processeur

Métrique	Description
<code>Gateway.DataProcessor.IngestionThrottled</code> (non disponible sur les passerelles V3 compatibles MQTT)	Le nombre de points de données qui ont été limités, générés chaque minute. Unité : nombre Dimensions : <code>ThrottledAt</code>
<code>Gateway.DataProcessor.MeasurementRejected</code> (non disponible sur les passerelles V3 compatibles MQTT)	Le nombre de mesures rejetées, généré chaque minute. Unité : nombre Dimensions : <code>raison</code>
<code>Gateway.DataProcessor.MessagesRemaining</code> (non disponible sur les passerelles V3 compatibles MQTT)	Le nombre de messages restant dans un flux, généré chaque minute. Unité : nombre Dimensions : <code>StreamName</code>
<code>Gateway.DataProcessor.ProcessingError</code> (non disponible sur les passerelles V3 compatibles MQTT)	Le nombre d'erreurs de traitement, générées chaque minute. Unité : nombre Dimensions : <code>raison</code>

Enregistrez les appels AWS IoT SiteWise d'API avec AWS CloudTrail

AWS IoT SiteWise 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 SiteWise. CloudTrail capture les appels d'API AWS IoT SiteWise sous forme d'événements. Les appels capturés incluent des appels provenant de la AWS IoT SiteWise console et des appels de code vers les opérations de l'AWS IoT SiteWise 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 SiteWise. 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 SiteWise, 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 SiteWise informations dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité événementielle prise en charge se produit dans AWS IoT SiteWise, cette activité est enregistrée dans un CloudTrail événement avec 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 SiteWise, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au 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](#)

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 faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).

- 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](#).

AWS IoT SiteWise événements de données dans CloudTrail

Les [événements de données](#) fournissent des informations sur les opérations de ressources effectuées sur ou dans une ressource (par exemple, lecture ou écriture de données dans un objet Amazon S3). Ils sont également connus sous le nom opérations de plans de données. Les événements de données sont souvent des activités dont le volume est élevé. Par défaut, CloudTrail n'enregistre pas les événements liés aux données. L'historique des CloudTrail événements n'enregistre pas les événements liés aux données.

Des frais supplémentaires s'appliquent pour les événements de données. Pour plus d'informations sur la CloudTrail tarification, consultez la section [AWS CloudTrail Tarification](#).

Vous pouvez enregistrer les événements de données pour les types de AWS IoT SiteWise ressources à l'aide de la CloudTrail console ou AWS CLI des opérations de CloudTrail l'API. Le [tableau](#) de cette section indique les types de ressources disponibles pour AWS IoT SiteWise.

- Pour enregistrer les événements de données à l'aide de la CloudTrail console, créez un [magasin de données de suivi ou d'événement](#) pour enregistrer les événements, ou [mettez à jour un magasin de données de suivi ou d'événement existant](#) pour enregistrer les événements de données.
 1. Choisissez Data events pour enregistrer les événements liés aux données.
 2. Dans la liste des types d'événements de données, choisissez le type de ressource pour lequel vous souhaitez enregistrer les événements de données.
 3. Choisissez le modèle de sélecteur de journal que vous souhaitez utiliser. Vous pouvez enregistrer tous les événements de données pour le type de ressource, consigner tous les readOnly événements, consigner tous les writeOnly événements ou créer un modèle de sélecteur de journal personnalisé pour filtrer les ressources. ARN champs readOnlyeventName, et.
- Pour enregistrer des événements de données à l'aide de AWS CLI, configurez le `--advanced-event-selectors` paramètre pour définir le `eventCategory` champ égal à la valeur du type

de ressource Data et le `resources.type` champ égal à la valeur du type de ressource (voir le [tableau](#)). Vous pouvez ajouter des conditions pour filtrer les valeurs des `resources.ARN` champs `readOnlyEventName`, et.

- Pour configurer un suivi afin de consigner les événements liés aux données, exécutez la [AWS CloudTrail put-event-selectors](#) commande. Pour plus d'informations, consultez la section [Enregistrement des événements de données pour les sentiers avec le AWS CLI](#).
- Pour configurer un magasin de données d'événements afin de consigner les événements, exécutez la [AWS CloudTrail create-event-data-store](#) commande pour créer un nouveau magasin de données d'événements pour enregistrer les événements ou exécutez la [AWS CloudTrail update-event-data-store](#) commande pour mettre à jour un magasin de données d'événements existant. Pour plus d'informations, consultez la section [Enregistrement des événements de données pour les magasins de données d'événements avec le AWS CLI](#).

Le tableau suivant répertorie les types de AWS IoT SiteWise ressources. La colonne Type d'événement de données (console) indique la valeur à choisir dans la liste des types d'événements de données de la CloudTrail console. La colonne de valeur `resources.type` indique la **`resources.type`** valeur que vous devez spécifier lors de la configuration de sélecteurs d'événements avancés à l'aide du ou. AWS CLI CloudTrail APIs La CloudTrail colonne Données APIs enregistrées indique les appels d'API enregistrés CloudTrail pour le type de ressource.

Type d'événement de données (console)	valeur <code>resources.type</code>	Données APIs enregistrées sur CloudTrail *
AWS IoT SiteWise asset	<code>AWS::IoTSiteWise::Asset</code>	• BatchPutAssetPropertyValues • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValues

Type d'événement de données (console)	valeur resources.type	Données APIs enregistrées sur CloudTrail *
		• BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise séries chronologiques	AWS::IoTSiteWise::TimeSeries	• BatchPutAssetPropertyValue • GetAssetPropertyValue • GetAssetPropertyValueHistory • GetAssetPropertyAggregates • GetInterpolatedAssetPropertyValues • BatchGetAssetPropertyValue • BatchGetAssetPropertyValueHistory • BatchGetAssetPropertyAggregates
AWS IoT SiteWise Assistante	AWS::SitewiseAssistant::Conversation	• InvokeAssistant

 Note

Le fichier resources.type enregistré dans l'événement Cloudtrail dépend de l'identifiant utilisé dans la demande d'API. Si un identifiant d'actif est spécifié dans la demande, le fichier Asset resources.type est enregistré, sinon le fichier TimeSeries resources.type est enregistré.

*Vous pouvez configurer des sélecteurs d'événements avancés pour filtrer les `eventName` `readOnly`, et des `resources` `ARN` champs pour enregistrer uniquement les événements qui sont importants pour vous. Pour plus d'informations sur ces champs, voir [AdvancedFieldSelector](#).

AWS IoT SiteWise événements de gestion dans CloudTrail

[Les événements de gestion de journalisation](#) fournissent des informations sur les opérations de gestion effectuées sur les ressources de votre AWS compte. Ils sont également connus sous le nom opérations de plan de contrôle. Par défaut, CloudTrail enregistre les événements de gestion.

AWS IoT SiteWise enregistre toutes les opérations AWS IoT SiteWise du plan de contrôle en tant qu'événements de gestion. Pour obtenir la liste des opérations du plan de AWS IoT SiteWise contrôle auxquelles AWS IoT SiteWise se connecte CloudTrail, consultez la [référence de l'AWS IoT SiteWise API](#).

Exemple : entrées de fichier AWS IoT SiteWise journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'opération demandée, la date et l'heure de l'opération, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateAsset` opération.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/Administrator",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "userName": "Administrator",
    "sessionContext": {
      "sessionIssuer": {},
      "webIdFederationData": {},
```

```
    "attributes": {
      "mfaAuthenticated": "false",
      "creationDate": "2020-03-11T17:26:40Z"
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2020-03-11T18:01:22Z",
  "eventSource": "iotsitewise.amazonaws.com",
  "eventName": "CreateAsset",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.0",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "assetName": "Wind Turbine 1",
    "assetModelId": "a1b2c3d4-5678-90ab-cdef-11111EXAMPLE",
    "clientToken": "a1b2c3d4-5678-90ab-cdef-00000EXAMPLE"
  },
  "responseElements": {
    "assetId": "a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetArn": "arn:aws:iotsitewise:us-east-1:123456789012:asset/a1b2c3d4-5678-90ab-cdef-22222EXAMPLE",
    "assetStatus": {
      "state": "CREATING"
    }
  },
  "requestID": "a1b2c3d4-5678-90ab-cdef-aaaaaEXAMPLE",
  "eventID": "a1b2c3d4-5678-90ab-cdef-bbbbbbEXAMPLE",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Marquez vos AWS IoT SiteWise ressources

Le balisage de vos AWS IoT SiteWise ressources constitue un moyen puissant de catégoriser, de gérer et de récupérer efficacement les actifs de l'organisation. En attribuant des balises, qui consistent en des paires clé-valeur, vous pouvez associer des métadonnées descriptives à vos ressources. Les métadonnées des balises peuvent être utilisées pour rationaliser les opérations. Par exemple, dans un scénario de parc éolien, les balises vous permettent d'étiqueter les turbines avec des attributs spécifiques tels que l'emplacement, la capacité et l'état opérationnel, ce qui permet une identification et une gestion rapides au sein de celles-ci AWS IoT SiteWise.

L'intégration de balises aux politiques AWS Identity and Access Management (IAM) améliore la sécurité et le contrôle opérationnel en définissant des règles d'accès conditionnel. Cela signifie que vous ne pouvez spécifier que les utilisateurs dotés de certaines balises. Par exemple, seules les personnes associées à un certain rôle ou département peuvent accéder à des ressources spécifiques ou les modifier.

Utiliser des balises dans AWS IoT SiteWise

Utilisez des balises pour classer vos AWS IoT SiteWise ressources par objectif, propriétaire, environnement ou toute autre classification adaptée à votre cas d'utilisation. Lorsque vous avez de nombreuses ressources de même type, vous pouvez rapidement identifier une ressource spécifique en fonction des balises que vous lui avez attribuées.

Chaque balise est composée d'une clé et d'une valeur facultative que vous spécifiez. Par exemple, vous pouvez définir une série de balises pour vos modèles d'actifs afin de les suivre en fonction des processus industriels qu'ils prennent en charge. Il est recommandé de développer un ensemble personnalisé de clés de balise pour chaque type de ressource que vous gérez. L'utilisation d'un ensemble cohérent de clés de balise peut faciliter la gestion des ressources.

Marquez avec le AWS Management Console

L'éditeur de balises AWS Management Console fournit un moyen centralisé et unifié de créer et de gérer vos balises pour les ressources de tous les AWS services. Pour plus d'informations, consultez la section [Mise en route avec l'éditeur de balises](#) dans le guide de l'utilisateur AWS des ressources de balisage et de l'éditeur de balises.

Marquer avec l' AWS IoT SiteWise API

L' AWS IoT SiteWise API utilise également des balises. Avant de créer des balises, tenez compte des restrictions liées aux balises. Pour de plus amples informations, veuillez consulter [Conventions de dénomination et d'utilisation des balises](#) dans le Références générales AWS.

- Pour ajouter des balises lorsque vous créez une ressource, définissez-les dans la propriété `tags` de la ressource.
- Pour ajouter des balises à une ressource existante ou pour mettre à jour les valeurs des balises, utilisez l'[TagResource](#) opération.
- Pour supprimer des balises d'une ressource, utilisez l'[UntagResource](#) opération.
- Pour récupérer les balises associées à une ressource, utilisez l'[ListTagsForResource](#) opération ou décrivez la ressource et inspectez ses `tags` propriétés.

Le tableau suivant répertorie les ressources que vous pouvez baliser à l'aide de l' AWS IoT SiteWise API, ainsi que leurs `Describe` opérations `Create` et opérations correspondantes.

Ressources taguables AWS IoT SiteWise

Ressource	Opération de création	Opération de description
Modèle d'actif ou modèle de composant	CreateAssetModel	DescribeAssetModel
Ressource	CreateAsset	DescribeAsset
SiteWise Passerelle Edge	CreateGateway	DescribeGateway
Portal	CreatePortal	DescribePortal
Projet	CreateProject	DescribeProject
Tableau de bord	CreateDashboard	DescribeDashboard
Stratégie d'accès	CreateAccessPolicy	DescribeAccessPolicy
Séries chronologiques	BatchPutAssetPropertyValue	DescribeTimeSeries

En [BatchPutAssetPropertyValue](#) effet, vous pouvez configurer vos sources de données auxquelles envoyer des données industrielles AWS IoT SiteWise avant de créer des modèles d'actifs et des actifs. AWS IoT SiteWise crée automatiquement des flux de données pour recevoir des flux de données brutes provenant de votre équipement. Pour plus d'informations, consultez la section [Gestion de l'ingestion de données](#).

Utilisez les opérations suivantes afin d'afficher et de gérer des balises pour les ressources qui prennent en charge le balisage :

- [TagResource](#)— Ajoute des balises à une ressource ou met à jour la valeur d'une balise existante.
- [ListTagsForResource](#)— Répertorie les balises d'une ressource.
- [UntagResource](#)— Supprime les balises d'une ressource.

Ajoutez ou supprimez des balises d'une ressource à tout moment. Pour mettre à jour la valeur d'une clé de balise existante, ajoutez une nouvelle balise avec la même clé et la nouvelle valeur souhaitée à la ressource. Cette action remplace l'ancienne valeur par la nouvelle. Bien qu'il soit possible d'attribuer une chaîne vide comme valeur de balise, vous ne pouvez pas attribuer de valeur nulle.

La suppression d'une ressource entraîne également la suppression de tous les tags qui y sont liés.

Utiliser des balises avec des politiques IAM

Utilisez des balises de ressources dans vos politiques IAM pour contrôler l'accès et les autorisations des utilisateurs. Par exemple, les politiques peuvent autoriser les utilisateurs à créer uniquement des ressources associées à une balise spécifique. Les stratégies peuvent également empêcher les utilisateurs de créer ou de modifier des ressources qui ont des balises spécifiques.

Note

Si vous utilisez des balises pour autoriser ou refuser l'accès des utilisateurs aux ressources, vous devez refuser aux utilisateurs la possibilité d'ajouter ou de supprimer ces balises pour les mêmes ressources. Dans le cas contraire, un utilisateur pourrait contourner vos restrictions et accéder à une ressource en modifiant ses balises.

Vous pouvez utiliser les clés et valeurs de contexte de condition suivantes dans l'élément `Condition` (également appelé le bloc `Condition`) d'une instruction de stratégie.

`aws:ResourceTag/tag-key: tag-value`

Accorder ou refuser aux utilisateurs des actions sur des ressources ayant des balises spécifiques.

`aws:RequestTag/tag-key: tag-value`

Exiger qu'une balise spécifique soit utilisée (ou non) lors de la création ou de la modification d'une ressource balisable.

`aws:TagKeys: [tag-key, ...]`

Exiger qu'un ensemble spécifique de clés de balise soit utilisé (ou non) lors de la création ou de la modification d'une ressource balisable.

Note

Les clés et valeurs de contexte de condition d'une politique IAM s'appliquent uniquement aux actions dont le paramètre obligatoire est une ressource balisable. Par exemple, vous pouvez définir un accès conditionnel basé sur des balises pour [ListAssets](#). Vous ne pouvez pas activer l'accès conditionnel basé sur des balises [PutLoggingOptions](#) car aucune ressource balisable n'est référencée dans la demande.

Pour plus d'informations, consultez la section [Contrôle de l'accès aux AWS ressources à l'aide de balises de ressources](#) et la [référence de politique IAM JSON](#) dans le guide de l'utilisateur IAM.

Exemples de politiques IAM utilisant des balises

- [Afficher les AWS IoT SiteWise ressources en fonction des balises](#)

Résolution des problèmes AWS IoT SiteWise

Utilisez les informations suivantes pour résoudre les problèmes liés à AWS IoT SiteWise.

Rubriques

- [Résolution des problèmes liés à une passerelle SiteWise Edge](#)
- [Résoudre les problèmes liés à un portail AWS IoT SiteWise](#)
- [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#)
- [Résolution des problèmes liés aux opérations d'importation et d'exportation en masse](#)

Résolution des problèmes liés à une passerelle SiteWise Edge

Résolvez les problèmes courants liés à la passerelle AWS IoT SiteWise Edge en explorant les rubriques pertinentes.

Vous pouvez également consulter CloudWatch les métriques signalées par vos passerelles SiteWise Edge pour résoudre les problèmes de connectivité ou de flux de données. Pour de plus amples informations, veuillez consulter [Surveillez AWS IoT SiteWise avec Amazon CloudWatch Metrics](#).

Rubriques

- [Configuration et accès aux journaux de la passerelle SiteWise Edge](#)
- [Résolution des problèmes liés à la passerelle SiteWise Edge](#)
- [Résolution des problèmes liés à l'application AWS IoT SiteWise Edge sur Siemens Industrial Edge](#)
- [Résolution des problèmes liés aux intégrations open source à la périphérie](#)
- [AWS IoT Greengrass Problèmes de résolution des problèmes](#)

Configuration et accès aux journaux de la passerelle SiteWise Edge

Avant de pouvoir consulter les journaux de la passerelle SiteWise Edge, vous devez configurer votre passerelle SiteWise Edge pour qu'elle envoie des journaux à Amazon CloudWatch Logs ou qu'elle les stocke sur le système de fichiers local.

- Utilisez CloudWatch les journaux si vous souhaitez utiliser le AWS Management Console pour afficher les fichiers journaux de votre passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Utiliser Amazon CloudWatch Logs](#).

- Utilisez les journaux du système de fichiers local si vous souhaitez utiliser la ligne de commande ou un logiciel local pour afficher les fichiers journaux de votre passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Utiliser les connexions au service AWS IoT SiteWise](#).

Résolution des problèmes liés à la passerelle SiteWise Edge

Utilisez les informations suivantes pour résoudre les problèmes liés à la passerelle SiteWise Edge.

Problèmes

- [Impossible de déployer des packs sur les passerelles SiteWise Edge](#)
- [AWS IoT SiteWise ne reçoit pas de données en provenance des serveurs OPC UA](#)
- [Aucune donnée n'apparaît dans le tableau de bord](#)
- [« Impossible de trouver ou de charger la classe principale » qui s'affiche dans le fichier aws.iot.SiteWiseEdgePublisher se connecte à/greengrass/v2/logerror](#)
- [Je vois « SESSION_TAKEN_OVER » ou « com.aws.greengrass.mqttclient ». MqttClient: Impossible de publier le message via Spooler et je vais réessayer. ' dans les journaux](#)
- [Je vois « com.aws.greengrass.deployment ». lotJobsHelper: aucune tâche de déploiement n'a été trouvée. ' ou « Le résultat du déploiement a déjà été signalé ». dans les journaux](#)
- [L'état « SYNC_FAILED » s'affiche lorsque je tente de configurer le paramètre d'horodatage dans un groupe de propriétés sur une source de données OPC UA](#)
- [Les types de données convertis ne sont pas inclus](#)
- [Problèmes liés à Trust Store](#)
- [Problèmes d'installation avec proxy](#)

Impossible de déployer des packs sur les passerelles SiteWise Edge

Si le composant AWS IoT Greengrass nucleus (`aws.greengrass.Nucleus`) est obsolète, il se peut que vous ne puissiez pas déployer de packs sur votre passerelle SiteWise Edge. Vous pouvez utiliser la AWS IoT Greengrass V2 console pour mettre à niveau le composant AWS IoT Greengrass Nucleus.

Pour mettre à niveau le composant AWS IoT Greengrass Nucleus (console)

1. Accédez à la [console AWS IoT Greengrass](#).
2. Dans le volet de navigation, sous AWS IoT Greengrass, choisissez Deployments.

3. Dans la liste des déploiements, sélectionnez le déploiement que vous souhaitez réviser.
4. Choisissez Réviser.
5. Sur la page Spécifier la cible, choisissez Next.
6. Sur la page Sélectionner les composants, sous Composants publics, dans la zone de recherche, entrez **aws.greengrass.Nucleus**, puis sélectionnez AWS.Greengrass.Nucleus.
7. Choisissez Suivant.
8. Sur la page Configurer les composants, choisissez Next.
9. Sur la page Configurer les paramètres avancés, choisissez Next.
10. Sur la page Review (Révision), choisissez Deploy (Déployer).

AWS IoT SiteWise ne reçoit pas de données en provenance des serveurs OPC UA

Si vos AWS IoT SiteWise actifs ne reçoivent pas les données envoyées par vos serveurs OPC UA, vous pouvez effectuer des recherches dans les journaux de votre passerelle SiteWise Edge pour résoudre les problèmes. Recherchez les `swPublisher` journaux au niveau des informations qui contiennent le message suivant.

```
Emitting diagnostic name=PublishError.SomeException
```

En fonction du type de contenu *SomeException* dans le journal, utilisez les types d'exception suivants et les problèmes correspondants pour résoudre les problèmes liés à votre passerelle SiteWise Edge :

- **ResourceNotFoundException**— Vos serveurs OPC UA envoient des données qui ne correspondent à aucun alias de propriété pour un actif. Cette exception peut se produire dans deux cas :
 - Les alias de vos propriétés ne correspondent pas exactement à vos variables OPC UA, y compris les préfixes de source que vous avez définis. Vérifiez que vos alias de propriété et vos préfixes source sont corrects.
 - Vous n'avez pas mappé vos variables OPC UA aux propriétés des actifs. Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).

Si vous avez déjà mappé toutes les variables OPC UA que vous souhaitez inclure AWS IoT SiteWise, vous pouvez filtrer les variables OPC UA envoyées par la passerelle SiteWise Edge. Pour de plus amples informations, veuillez consulter [Utiliser les filtres de nœuds OPC UA dans Edge SiteWise](#).

- **InvalidRequestException**— Les types de données de vos variables OPC UA ne correspondent pas aux types de données des propriétés de vos actifs. Par exemple, si une variable OPC UA possède un type de données entier, la propriété d'actif correspondante doit être de type entier. Une propriété d'actif de type double ne peut pas recevoir de valeurs entières OPC UA. Pour résoudre ce problème, définissez de nouvelles propriétés avec les types de données corrects.
- **TimestampOutOfRangeException**— Votre passerelle SiteWise Edge envoie des données qui se situent en dehors de la plage d' AWS IoT SiteWise acceptation. AWS IoT SiteWise rejette tous les points de données dont l'horodatage est antérieur à 7 jours dans le passé ou inférieur à 5 minutes dans le futur. En cas de perte d'alimentation ou de connexion au AWS cloud de votre passerelle SiteWise Edge, vous devrez peut-être vider le cache de votre passerelle SiteWise Edge.
- **ThrottlingException** ou **LimitExceededException**: votre demande a dépassé un quota de AWS IoT SiteWise service, tel que le taux de points de données ingérés ou le taux de demandes pour les opérations de l'API de données relatives aux propriétés des actifs. Vérifiez que votre configuration ne dépasse pas le [AWS IoT SiteWise quotas](#).

Aucune donnée n'apparaît dans le tableau de bord

Si aucune donnée n'apparaît dans votre tableau de bord, il est possible que la configuration de l'éditeur et la source de données de la passerelle SiteWise Edge ne soient pas synchronisées. S'ils ne sont pas synchronisés, la mise à jour du nom de la source de données peut accélérer la synchronisation entre le cloud et le périphérique, corrigeant ainsi l'erreur de désynchronisation.

Pour mettre à jour le nom d'une source de données

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle SiteWise Edge connectée au tableau de bord.
4. Sous Sources de données, sélectionnez Modifier.
5. Sélectionnez un nouveau nom de source, puis sélectionnez Enregistrer pour confirmer votre modification.
6. Vérifiez vos modifications en confirmant que le nom de la source de données a été mis à jour dans le tableau des sources de données.

« Impossible de trouver ou de charger la classe principale » qui s'affiche dans le fichier `aws.iot.SiteWiseEdgePublisher` se connecte à `/greengrass/v2/logerror`

Si cette erreur s'affiche, vous devrez peut-être mettre à jour la version Java de votre passerelle SiteWise Edge.

- Depuis un terminal, exécutez la commande suivante :

```
java -version
```

La version de Java utilisée par votre passerelle SiteWise Edge s'affichera sous `OpenJDK Runtime Environment`. Vous verrez une réponse semblable à la suivante :

```
openjdk version "11.0.20" 2023-07-18 LTS
OpenJDK Runtime Environment Corretto011.0.20.8.1 (build 11.0.20+8-LTS
OpenJDK 64-Bit Server VM Corretto-11.0.20.8.1 (build 11.0.20+8-LTS, mixed node)
```

Si vous utilisez la version 11.0.20.8.1 de Java, vous devez mettre à jour le pack IoT SiteWise Publisher vers la version 2.4.1 ou une version ultérieure. Seule la version 11.0.20.8.1 de Java est affectée. Les environnements dotés d'autres versions de Java peuvent continuer à utiliser les anciennes versions du composant IoT SiteWise Publisher. Pour plus d'informations sur la mise à jour d'un pack de composants, consultez [Modifier la version des packs de composants de la passerelle SiteWise Edge](#).

Je vois « `SESSION_TAKEN_OVER` » ou « `com.aws.greengrass.mqttclient` ». `MqttClient: Impossible de publier le message via Spooler et je vais réessayer.` ' dans les journaux

Si un avertissement `SESSION_TAKEN_OVER` ou une erreur s'affiche `com.aws.greengrass.mqttclient.MqttClient: Failed to publish the message via Spooler and will retry.` dans vos journaux à l'adresse `/greengrass/v2/logs/greengrass.log`, vous essayez peut-être d'utiliser le même fichier de configuration pour plusieurs passerelles SiteWise Edge sur plusieurs appareils. Chaque passerelle SiteWise Edge a besoin d'un fichier de configuration unique pour se connecter à votre AWS compte.

Je vois « `com.aws.greengrass.deployment` ». `lotJobsHelper`: aucune tâche de déploiement n'a été trouvée. ' ou « Le résultat du déploiement a déjà été signalé ». dans les journaux

Si vous voyez `com.aws.greengrass.deployment.IotJobsHelper: No deployment job found.` ou `Deployment result already reported.` dans vos journaux à `/greengrass/v2/logs/greengrass.log`adresse, vous essayez peut-être de réutiliser le même fichier de configuration.

Les solutions sont multiples :

- Si vous souhaitez réutiliser le fichier de configuration, procédez comme suit :
 1. Accédez à la [console AWS IoT SiteWise](#).
 2. Dans le volet de navigation, choisissez Edge gateways.
 3. Choisissez la passerelle SiteWise Edge que vous souhaitez réutiliser.
 4. Choisissez l'onglet Mises à jour.
 5. Sélectionnez une autre version de Publisher et choisissez Déployer.

Suivez les étapes décrites [Créez une passerelle pour Siemens Industrial Edge](#) pour créer un nouveau fichier de configuration.

L'état « `SYNC_FAILED` » s'affiche lorsque je tente de configurer le paramètre d'horodatage dans un groupe de propriétés sur une source de données OPC UA

Lors de la AWS IoT SiteWise mise à jour du composant collecteur OPC UA AWS IoT Greengrass dans la version 2.5.0, nous avons introduit une nouvelle option de configuration d'horodatage. Vous pouvez utiliser l'horodatage de votre appareil ou celui du serveur. Les anciennes versions du composant collecteur OPC UA ne prennent pas en charge cette option et ne se synchronisent pas.

Il existe deux méthodes pour résoudre un échec de l'état de synchronisation des sources de données. La méthode recommandée consiste à mettre à niveau le composant collecteur IoT SiteWise OPC UA vers la version 2.5.0 ou supérieure. Vous pouvez également continuer à utiliser l'ancienne version du composant du collecteur OPC UA, si vous définissez l'horodatage sur. `Source` Pour savoir comment mettre à niveau le composant collecteur IoT SiteWise OPC UA, voir [Mettre à jour la version d'un AWS IoT SiteWise composant](#). Nous vous recommandons d'utiliser les dernières versions de tous les composants.

 Note

Il n'y a aucune interruption de données en cas d'échec de l'état de synchronisation d'une source de données. Les données sources continuent d'affluer vers AWS IoT SiteWise. La configuration ne se synchronise tout simplement pas avec le composant collecteur IoT SiteWise OPC UA de votre AWS IoT Greengrass V2 déploiement.

Pour modifier la configuration d'horodatage d'un groupe de propriétés

1. Accédez à la [console AWS IoT SiteWise](#).
2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la passerelle à modifier.
4. Dans la section Sources de données, sélectionnez la source de données dont l'état de synchronisation a échoué, puis choisissez Modifier.
5. Développez la configuration avancée, puis développez les paramètres du groupe.
6. Dans Horodatage, sélectionnez Source. La sélection de Source supprime la `timestampToReturn` propriété de la configuration. Ce paramètre active la collecte de l'horodatage de la source de données à partir de votre appareil par défaut, ce qui permet à la source de données de se synchroniser avec le composant collecteur IoT SiteWise OPC UA.
7. Choisissez Enregistrer.

Les types de données convertis ne sont pas inclus

Si un message d'erreur s'affiche lors de la conversion de types de données OPC UA non pris en charge en chaînes de caractères AWS IoT SiteWise, plusieurs raisons sont possibles :

- Le type de données que vous essayez de convertir est un type de données complexe. Les types de données complexes ne sont pas pris en charge.
- Lorsque vous utilisez Destinations en tant que AWS IoT SiteWise mémoire tampon à l'aide d'Amazon S3, la valeur de chaîne complète est préservée dans les fichiers transférés vers un compartiment Amazon S3. Lorsque vous ingérez ultérieurement des données dans AWS IoT SiteWise, les valeurs de chaîne complètes de plus de 1 024 octets sont rejetées.

Problèmes liés à Trust Store

Si vous rencontrez des problèmes liés aux magasins de confiance dans SiteWise Edge, suivez les étapes de résolution des problèmes suivantes :

- Vérifiez que le certificat CA AWS IoT Greengrass racine est présent et correctement formaté dans les magasins de confiance appropriés
- Assurez-vous que le KeyStore mot de passe Java est correctement défini et accessible aux composants SiteWise Edge
- Vérifiez que tous les certificats personnalisés (par exemple pour les proxys HTTPS) sont au bon format (généralement PEM) et correctement importés dans les magasins de confiance
- Vérifiez que les magasins de confiance disposent des autorisations de fichier correctes et sont accessibles aux processus SiteWise Edge
- Consultez les journaux SiteWise Edge pour détecter toute erreur SSL/TLS associée, qui pourrait indiquer des problèmes liés au Trust Store
- Testez SSL/TLS les connexions de manière indépendante à l'aide `openssl` d'outils tels que la vérification des fonctionnalités de Trust Store

Problèmes d'installation avec proxy

Si vous rencontrez des problèmes lors du processus de configuration du proxy, considérez les étapes de résolution des problèmes suivantes :

- Vérifiez que l'URL du proxy est correctement formatée et inclut le schéma approprié (`http://` ou `https://`)
- Assurez-vous que toutes les informations d'identification du proxy sont codées en URL si elles contiennent des caractères spéciaux
- Vérifiez que la liste des adresses sans proxy inclut toutes les adresses locales et points de terminaison de AWS service nécessaires
- Pour les proxys HTTPS, vérifiez que le certificat CA fourni est au format PEM
- Consultez les journaux d'installation pour détecter les messages d'erreur spécifiques susceptibles d'indiquer la source du problème
- Testez la connexion proxy de manière indépendante pour vous assurer qu'elle fonctionne correctement

Résolution des problèmes liés à l'application AWS IoT SiteWise Edge sur Siemens Industrial Edge

Pour résoudre les problèmes liés à l'application AWS IoT SiteWise Edge sur votre Siemens Industrial Edge appareil, vous pouvez accéder aux journaux de l'application via le portail Siemens Industrial Edge Management ou le portail Siemens Industrial Edge Device (IED). Pour plus d'informations, consultez la section [Téléchargement des journaux](#) dans la documentation de Siemens.

Mes données ne s'affichent pas dans AWS IoT SiteWise

- Assurez-vous qu'il n'y a aucun problème avec vos Databus utilisateurs et que l'icône en forme de coche correspondant à la configuration de base de données est verte plutôt que grise.
- Il se peut que vous n'utilisiez pas une version contenant Secure Storage. Mettez à niveau votre version du système d'exploitation Siemens. Pour de plus amples informations, veuillez consulter [Siemens Secure Storage et l'application AWS IoT SiteWise Edge](#).

Je vois « Fichier de configuration manquant AWS_REGION » dans les journaux

Si vous voyez `Config file missing AWS_REGION` dans les journaux de Siemens que le JSON du fichier de configuration est endommagé. Vous devez créer un nouveau fichier de configuration. Suivez les étapes décrites [Créez une passerelle pour Siemens Industrial Edge](#) pour créer un nouveau fichier de configuration.

Un message d'erreur « Désynchronisation » s'affiche sur la configuration de la passerelle Edge

Si un message d'erreur `Out of sync` s'affiche sur votre Siemens Industrial Edge passerelle une fois le déploiement terminé, cela signifie que le composant SiteWise éditeur IoT n'est pas synchronisé avec votre passerelle. Le composant SiteWise éditeur IoT fonctionne en arrière-plan sur les Siemens Industrial Edge passerelles pour fournir des fonctionnalités thématiques MQTT. Nous avons mis à niveau Siemens Industrial Edge les passerelles pour utiliser l'espace de noms des capacités `iotsitewise:publisher:3` plutôt que `iotsitewise:publisher:2`. Vous pouvez effectuer une mise à jour vers la dernière version de l'éditeur pour résoudre ce problème.

Pour passer à la dernière version de l'éditeur IoT SiteWise

1. Accédez à la [console AWS IoT SiteWise](#).

2. Dans le volet de navigation, choisissez Edge gateways.
3. Sélectionnez la Siemens Industrial Edge passerelle à modifier.
4. Dans la section Fonctionnalités Edge, sélectionnez Afficher les versions du logiciel.
5. Sélectionnez la dernière version de l' SiteWise éditeur IoT dans le menu déroulant Publisher.
6. Sélectionnez Exécuté.

Résolution des problèmes liés aux intégrations open source à la périphérie

Cette section fournit des solutions aux problèmes courants que vous pouvez rencontrer lors de l'intégration d'outils open source à SiteWise Edge.

Note

Node-RED®, InfluxDB® et Grafana® ne sont ni des fournisseurs ni des fournisseurs d'Edge. SiteWise

Problèmes de connexion

Node-RED ne peut pas se connecter au courtier MQTT

Vérifiez que le broker MQTT est en cours d'exécution et qu'il est accessible sur le port spécifié. Vérifiez la configuration de votre réseau et assurez-vous que l'adresse du courtier est correcte.

Pour vérifier l'état du broker MQTT, exécutez :

```
docker ps | grep emqx
```

Erreurs de connexion InfluxDB

Assurez-vous que votre jeton d'authentification est valide et que vous avez spécifié les noms d'organisation et de compartiment corrects. Vérifiez qu'InfluxDB est en cours d'exécution et accessible.

Pour vérifier l'état d'InfluxDB, exécutez :

```
curl -I http://localhost:8086
```


Grafana ne peut pas se connecter à InfluxDB

Vérifiez que la configuration de la source de données InfluxDB dans Grafana est correcte, y compris l'URL, le jeton d'authentification, l'organisation et le compartiment.

Problèmes de flux de données

Aucune donnée n'apparaît dans AWS IoT SiteWise

Vérifiez que l'alias de votre propriété dans le flux Node-RED correspond au format attendu. Vérifiez que la structure des rubriques MQTT est correcte et que la passerelle SiteWise Edge est correctement configurée pour recevoir les données du courtier MQTT.

Aucune donnée SiteWise Edge stockée dans InfluxDB

Vérifiez que le flux de rétention Node-RED est correctement configuré et que le nœud d'écriture InfluxDB possède les paramètres de bucket et de mesure appropriés. Vérifiez la sortie de débogage de Node-RED pour détecter toute erreur.

Erreurs de format de données

Assurez-vous que vos fonctions de transformation des données convertissent correctement les données entre les formats. Utilisez les nœuds de débogage Node-RED pour inspecter les données à chaque étape du flux.

Problèmes de performance

Utilisation élevée du processeur ou de la mémoire

Surveillez l'utilisation des ressources et ajustez la configuration de vos composants selon les besoins. Envisagez de réduire la fréquence de collecte des données ou de mettre en œuvre un filtrage des données pour réduire la charge de traitement.

Pour surveiller l'utilisation des ressources, exécutez :

```
docker stats
```

Chargement lent du tableau de bord Grafana

Optimisez vos requêtes InfluxDB et envisagez d'ajouter des limites de plage de temps à vos panneaux de tableau de bord. Réduisez le nombre de points de données affichés en utilisant les fonctions d'agrégation appropriées.

Journalisation et diagnostic

Pour résoudre les problèmes, consultez les journaux de chaque composant :

Logs Nœud-Red

Affichez les journaux dans la console Node-RED ou exécutez :

```
docker logs node-red
```

Journaux InfluxDB

Accédez aux journaux en exécutant :

```
docker logs influxdb
```

Blogs Grafana

Afficher les journaux en exécutant :

```
docker logs grafana
```

SiteWise Journaux Edge

Consultez les journaux de la passerelle SiteWise Edge pour détecter les problèmes de connexion MQTT et de traitement des données. Pour de plus amples informations, veuillez consulter [Résolution des problèmes liés à une passerelle SiteWise Edge](#).

AWS IoT Greengrass Problèmes de résolution des problèmes

Pour trouver des solutions à de nombreux problèmes liés à la configuration ou au déploiement de votre passerelle SiteWise Edge AWS IoT Greengrass, consultez la section [Résolution des problèmes AWS IoT Greengrass](#) dans le Guide du AWS IoT Greengrass développeur.

Résoudre les problèmes liés à un portail AWS IoT SiteWise

Résolvez les problèmes courants liés à vos AWS IoT SiteWise portails.

Les utilisateurs et les administrateurs ne peuvent pas accéder au AWS IoT SiteWise portail

Si les utilisateurs ou les administrateurs ne peuvent pas accéder à votre AWS IoT SiteWise portail, il se peut que vous disposiez d'autorisations restreintes dans le cadre de politiques associées AWS Identity and Access Management (IAM) qui empêchent les connexions réussies.

Consultez les exemples suivants de politiques IAM susceptibles d'entraîner un échec de connexion :

Note

Toute politique IAM attachée qui inclut un "Condition" élément entraînera un échec de connexion.

Exemple 1 : La condition ici est une adresse IP limitée, ce qui entraînera un échec de connexion.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "REPLACESAMPLEIP"
          ]
        }
      }
    }
  ]
}
```

Exemple 2 : La condition ici est une balise incluse, ce qui provoquera un échec de connexion.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotsitewise:DescribePortal"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "aws:ResourceTag/project": "*"
        }
      }
    }
  ]
}
```

Lorsque vous ajoutez des utilisateurs ou des administrateurs au portail, évitez de créer des politiques IAM qui limitent les autorisations des utilisateurs, telles qu'une adresse IP limitée. Les politiques associées avec des autorisations restreintes ne pourront pas se connecter au AWS IoT SiteWise portail.

Résoudre les problèmes liés à une action de AWS IoT SiteWise règle

Pour résoudre les problèmes liés à l'action de votre AWS IoT SiteWise règle dans AWS IoT Core, vous pouvez suivre l'une des procédures suivantes :

- Configuration d'Amazon CloudWatch Logs
- Configurer une action d'erreur de republication pour votre règle

Ensuite, comparez les messages d'erreur avec les erreurs de cette rubrique pour résoudre le problème.

Rubriques

- [Configuration AWS IoT Core des journaux](#)

- [Configurer une action d'erreur de republication](#)
- [Résoudre les problèmes liés aux règles](#)
- [Résoudre les problèmes liés à une règle \(\)AWS IoT SiteWise](#)
- [Résolution des problèmes liés à une règle \(DynamoDB\)](#)

Configuration AWS IoT Core des journaux

Vous pouvez configurer AWS IoT pour consigner différents niveaux d'informations dans CloudWatch Logs.

Pour configurer les CloudWatch journaux et y accéder

1. Pour configurer la journalisation pour AWS IoT Core, consultez la section [Surveillance à l'aide CloudWatch des journaux](#) dans le guide du AWS IoT développeur.
2. Accédez à la [console CloudWatch](#).
3. Dans le panneau de navigation, choisissez Groupes de journaux.
4. Choisissez le groupe AWSIoTLogs.
5. Choisissez un flux de journaux récent. Par défaut, CloudWatch affiche le flux de journal le plus récent en premier.
6. Choisissez une entrée de journal pour développer le message de journal. Votre entrée de journal peut ressembler à la capture d'écran suivante.

The screenshot shows the AWS CloudWatch console interface. The breadcrumb navigation at the top reads: CloudWatch > Log Groups > AWSIoTLogs > 9ca6614a-00fc-4f9e-8100-5c2a34918e90_123456789012_0. On the right, there are controls for 'Expand all' (checked), 'Row' (selected), and 'Text'. Below this is a 'Filter events' input field and a dropdown menu showing 'all' and '2020-02-10 (19:36:11)'. The main log entry table has two columns: 'Time (UTC +00:00)' and 'Message'. The entry for '2020-02-11' at '19:36:11' shows a message starting with '2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWise'. The full message text is: '2020-02-11 19:36:11.823 TRACEID:d4cd3bd0-ac41-cd4a-4f59-74a242ec70e6 PRINCIPALID:AIDAZ2YMUHYHIEDEL3VA3 [ERROR] EVENT:IotSiteWiseActionFailure TOPICNAME:/tutorial/device/SiteWiseTutorialDevice1/cpu CLIENTID:iotconsole-1581444173801-0 MESSAGE:Failed to send message data to IoT SiteWise asset properties. [Code: InvalidRequestException, Message: Property value does not match data type DOUBLE]. Message arrived on: /tutorial/device/SiteWiseTutorialDevice1/cpu, Action: iotSiteWise'. The entry is flanked by yellow bars with the text 'No older events found at the moment. Retry.' and 'No newer events found at the moment. Retry.'.

7. Comparez les messages d'erreur avec les erreurs de cette rubrique pour résoudre le problème.

Configurer une action d'erreur de republication

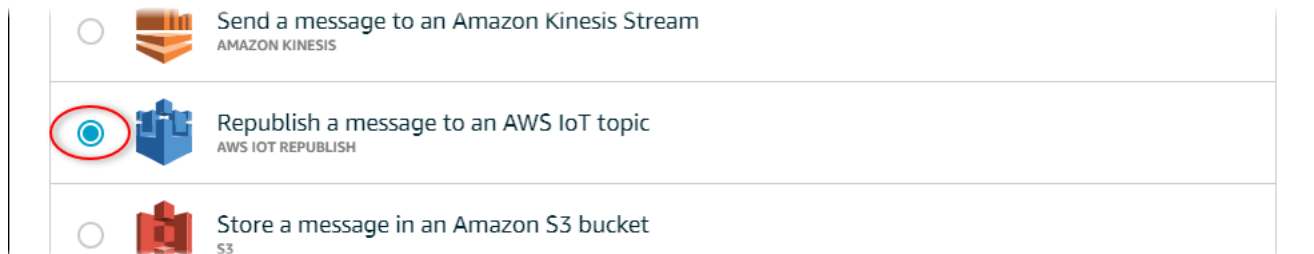
Vous pouvez configurer une action d'erreur au niveau d'une règle pour gérer les messages d'erreur. Dans cette procédure, vous configurez l'action de règle de republication en tant qu'action d'erreur pour afficher les messages d'erreur dans le client de test MQTT.

Note

L'action d'erreur de republication ne génère que l'équivalent des journaux de niveau ERROR. Si vous souhaitez des journaux plus détaillés, vous devez [configurer CloudWatch](#) les journaux.

Pour ajouter une action d'erreur de republication à une règle

1. Accédez à la [console AWS IoT](#).
2. Dans le panneau de navigation de gauche, choisissez Act (Agir) puis Rules (Règles).
3. Choisissez une règle.
4. Sous Error action (Action d'erreur), choisissez Add action (Ajouter une action).
5. Choisissez Republier un message dans un AWS IoT sujet.



6. En bas de la page, choisissez Configure action (Configurer l'action).
7. Dans Sujet, entrez un sujet unique (par exemple, **sitewise/windfarm/rule/error**). AWS IoT Core republiera les messages d'erreur dans cette rubrique.
8. Choisissez Sélectionner pour autoriser AWS IoT Core l'accès afin d'exécuter l'action d'erreur.
9. Choisissez Select (Sélectionner) en regard du rôle que vous avez créé pour la règle.
10. Choisissez Update Role (Mettre à jour le rôle) pour ajouter les autorisations supplémentaires au rôle.
11. Choisissez Add action.

L'action d'erreur de la règle devrait ressembler à la capture d'écran suivante.

Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.



Republish a message to an AWS IoT topic
sitewise/windfarm/rule/error

[Remove](#) [Edit](#) ▶

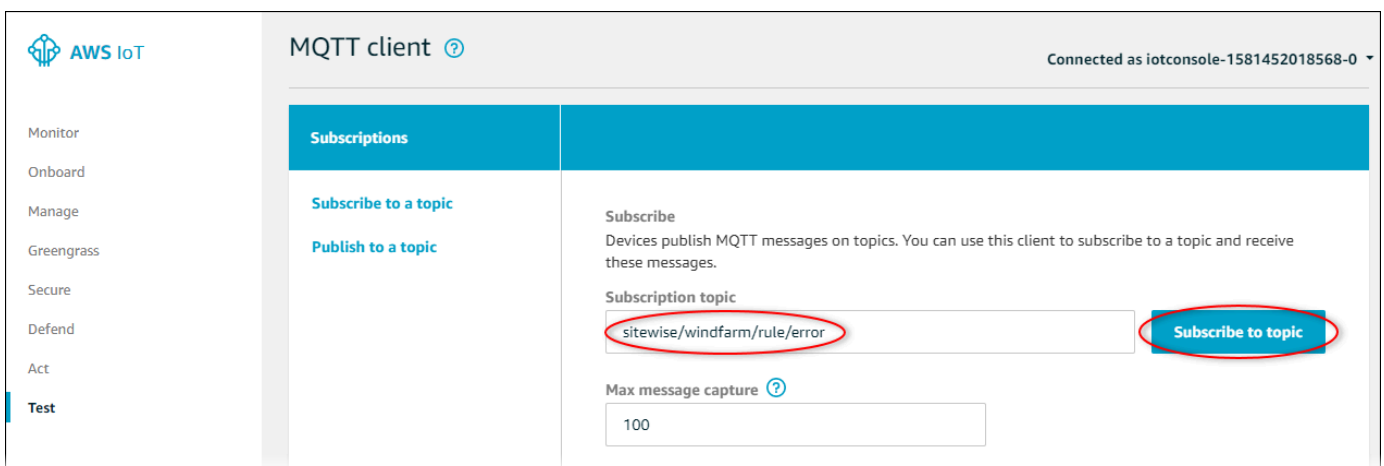
12. Cliquez sur la flèche de retour en haut à gauche de la console pour revenir à la page d'accueil de la AWS IoT console.

Après avoir configuré l'action d'erreur de republication, vous pouvez afficher les messages d'erreur dans le client de test MQTT dans AWS IoT Core.

Dans la procédure suivante, vous vous abonnez à la rubrique d'erreur dans le client de test MQTT. Le client de test MQTT vous permet de recevoir les messages d'erreur de la règle afin de résoudre le problème.

Pour vous abonner à la rubrique d'action d'erreur

1. Accédez à la [console AWS IoT](#).
2. Dans la page de navigation de gauche, choisissez Test pour ouvrir le client de test MQTT.
3. Dans le champ Subscription topic (Rubrique d'abonnement), entrez la rubrique d'erreur que vous avez configurée précédemment (par exemple, **sitewise/windfarm/rule/error**) et choisissez Subscribe to topic (S'abonner à la rubrique).



4. Surveillez les messages d'erreur qui s'affichent, puis développez le tableau failures dans chaque message d'erreur.

Ensuite, comparez les messages d'erreur avec les erreurs de cette rubrique pour résoudre le problème.

Résoudre les problèmes liés aux règles

Utilisez les informations suivantes pour résoudre les problèmes de règle.

Problèmes

- [Erreur : le membre doit être dans les 604800 secondes avant et 300 secondes après l'horodatage actuel](#)
- [Erreur : la valeur de la propriété ne correspond pas au type de données <type>](#)
- [Erreur : L'utilisateur <role-arn>n'est pas autorisé à exécuter : iotsitewise : aucune ressource BatchPutAssetPropertyValue](#)
- [Erreur : iot.amazonaws.com ne parvient pas à exécuter : sts : on resource : AssumeRole <role-arn>](#)
- [Info : Aucune demande n'a été envoyée. PutAssetPropertyValueEntries était vide après avoir effectué des modèles de substitution.](#)

Erreur : le membre doit être dans les 604800 secondes avant et 300 secondes après l'horodatage actuel

Votre horodatage date de plus de 7 jours ou de moins de 5 minutes, par rapport à l'époque Unix actuelle. Essayez les éléments suivants :

- Vérifiez que l'horodatage est au format d'heure Unix epoch (UTC). Si vous fournissez un horodatage avec un fuseau horaire différent, vous rencontrerez cette erreur.
- Vérifiez que votre horodatage est en secondes. AWS IoT SiteWise attend des horodatages divisés en secondes (à l'époque Unix) et décalés en nanosecondes.
- Vérifiez que vous téléchargez des données horodatées d'au plus 7 jours auparavant.

Erreur : la valeur de la propriété ne correspond pas au type de données <type>

Une entrée de votre action de règle comporte un type de données différent de celui de la propriété de ressource cible. Par exemple, la propriété de ressource cible est de type DOUBLE, tandis que le

type de données que vous avez sélectionné est Integer ou que vous avez transmis la valeur dans `integerValue`. Essayez les éléments suivants :

- Si vous configurez la règle depuis la AWS IoT console, vérifiez que vous avez choisi le bon type de données pour chaque entrée.
- Si vous configurez la règle à partir de l'API ou AWS Command Line Interface (AWS CLI), vérifiez que votre `value` objet utilise le champ de type correct (par exemple, `doubleValue` pour une DOUBLE propriété).

Erreur : L'utilisateur `<role-arn>` n'est pas autorisé à exécuter : `iotsitewise : aucune ressource BatchPutAssetPropertyValue`

Soit la règle n'est pas autorisée à accéder à la propriété de ressource cible, soit la propriété de ressource cible n'existe pas. Essayez les éléments suivants :

- Vérifiez que l'alias de la propriété est correct et que la propriété de ressource dispose de l'alias de propriété donné. Pour de plus amples informations, veuillez consulter [Gérez les flux de données pour AWS IoT SiteWise](#).
- Vérifiez que la règle est associée à un rôle et que le rôle `iotsitewise:BatchPutAssetPropertyValue` donne l'autorisation à la propriété de ressource cible, par exemple via la hiérarchie de la ressource cible. Pour de plus amples informations, veuillez consulter [Accordez AWS IoT l'accès requis](#).

Erreur : `iot.amazonaws.com` ne parvient pas à exécuter : `sts : on resource : AssumeRole <role-arn>`

Votre utilisateur n'est pas autorisé à assumer le rôle dans votre règle dans AWS Identity and Access Management (IAM).

Vérifiez que votre utilisateur est `iam:PassRole` autorisé à accéder au rôle indiqué dans votre règle. Pour plus d'informations, consultez la section [Transmettre les autorisations de rôle](#) dans le guide du AWS IoT développeur.

Info : Aucune demande n'a été envoyée. PutAssetPropertyValueEntries était vide après avoir effectué des modèles de substitution.

 Note

Ce message est un journal de niveau INFO.

Votre demande doit comporter au moins une entrée avec tous les paramètres requis.

Vérifiez que les paramètres de la règle, y compris les modèles de substitution, génèrent des valeurs non vides. Les modèles de substitution ne peuvent pas accéder aux valeurs définies dans les clauses AS de l'instruction de requête de la règle. Pour plus d'informations, consultez la section [Modèles de substitution](#) dans le Guide du AWS IoT développeur.

Résoudre les problèmes liés à une règle (AWS IoT SiteWise)

Suivez les étapes de cette procédure pour résoudre les problèmes liés à votre règle si les données d'utilisation du processeur et de la mémoire ne s'affichent pas AWS IoT SiteWise comme prévu. Dans cette procédure, vous configurez l'action de règle de republication en tant qu'action d'erreur pour afficher les messages d'erreur dans le client de test MQTT. Vous pouvez également configurer la journalisation dans CloudWatch Logs pour résoudre les problèmes. Pour de plus amples informations, veuillez consulter [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#).

Pour ajouter une action d'erreur de republication à une règle

1. Accédez à la [console AWS IoT](#).
2. Dans le volet de navigation de gauche, choisissez Routage des messages, puis Règles.
3. Choisissez la règle que vous avez créée précédemment, puis cliquez sur Modifier.
4. Sous Action d'erreur - facultatif, choisissez Ajouter une action d'erreur.
5. Choisissez Republier un message dans un AWS IoT sujet.
6. Dans Sujet, entrez le chemin de votre erreur (par exemple, **sitewise/rule/tutorial/error**). AWS IoT Core republiera les messages d'erreur dans cette rubrique.
7. Choisissez le rôle que vous avez créé précédemment (par exemple, SiteWiseTutorialDeviceRuleRole).
8. Choisissez Mettre à jour.

Après avoir configuré l'action d'erreur de republication, vous pouvez afficher les messages d'erreur dans le client de test MQTT dans AWS IoT Core.

Dans la procédure suivante, vous vous abonnez à la rubrique d'erreur dans le client de test MQTT.

Pour vous abonner à la rubrique d'action d'erreur

1. Accédez à la [console AWS IoT](#).
2. Dans la page de navigation de gauche, choisissez le client de test MQTT pour ouvrir le client de test MQTT.
3. Dans le champ Filtre par sujet, entrez **sitewise/rule/tutorial/error** et choisissez S'abonner.

Lorsque des messages d'erreur apparaissent, affichez le tableau `failures` dans n'importe quel message d'erreur pour diagnostiquer les problèmes. Pour plus d'informations sur les problèmes et les solutions possibles, consultez [Résoudre les problèmes liés à une action de AWS IoT SiteWise règle](#).

Si aucune erreur ne s'affiche, vérifiez que votre règle est activée et que vous vous êtes abonné à la même rubrique que celle que vous avez configurée dans l'action d'erreur de republication. Si des erreurs ne s'affichent toujours pas, vérifiez que le script du périphérique est en cours d'exécution et qu'il met à jour l'ombre du périphérique avec succès.

Note

Vous pouvez également vous abonner à la rubrique de mise à jour parallèle de votre appareil pour voir la charge utile analysée par votre AWS IoT SiteWise action. Pour ce faire, abonnez-vous à la rubrique suivante.

```
$aws/things/+/shadow/update/accepted
```

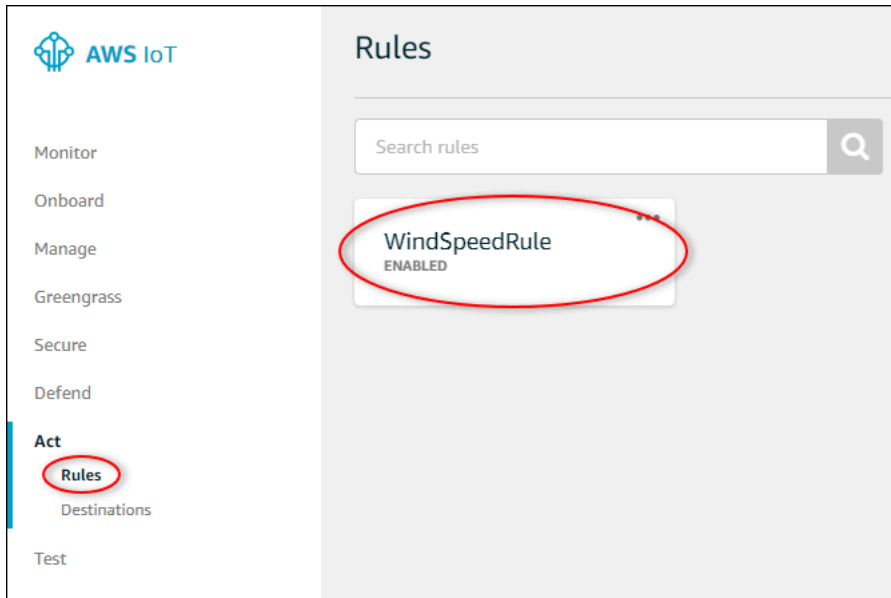
Résolution des problèmes liés à une règle (DynamoDB)

Suivez les étapes de cette procédure pour résoudre les problèmes liés à votre règle si les données des actifs de démonstration n'apparaissent pas dans le tableau DynamoDB comme prévu. Dans cette procédure, vous configurez l'action de règle de republication en tant qu'action d'erreur pour afficher les messages d'erreur dans le client de test MQTT. Vous pouvez également configurer la

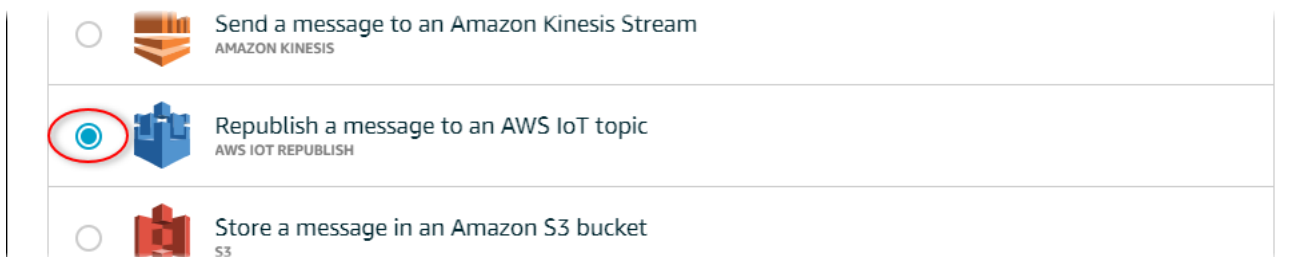
journalisation dans CloudWatch Logs pour résoudre les problèmes. Pour plus d'informations, veuillez consulter la rubrique [Surveillance avec les journaux CloudWatch](#) dans le Guide du développeur AWS IoT .

Pour ajouter une action d'erreur de republication à une règle

1. Accédez à la [console AWS IoT](#).
2. Dans le panneau de navigation de gauche, choisissez Act (Agir) puis Rules (Règles).
3. Choisissez la règle que vous avez créée précédemment.



4. Sous Error action (Action d'erreur), choisissez Add action (Ajouter une action).
5. Choisissez Republier un message dans un AWS IoT sujet.



6. En bas de la page, choisissez Configure action (Configurer l'action).
7. Dans Sujet, entrez **windspeed/error**. AWS IoT Core republiera les messages d'erreur dans cette rubrique.

Configure action

 **Republish a message to an AWS IoT topic**
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

*Topic 

windspeed/error

Quality of Service 

☒ 0 - The message is delivered zero or more times.

☐ 1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected Create Role Select

Cancel Add action

- Choisissez Sélectionner pour autoriser AWS IoT Core à exécuter l'action d'erreur en utilisant le rôle que vous avez créé précédemment.
- Choisissez Select (Sélectionner) en regard de votre rôle.

Choose or create a role to grant AWS IoT access to perform this action.

No role selected Refresh Create Role Close

 Search for IAM roles

WindSpeedDataRole Select

- Choisissez Update Role (Mettre à jour le rôle) pour ajouter les autorisations supplémentaires au rôle.

1 - The message is delivered one or more times.

Choose or create a role to grant AWS IoT access to perform this action.

WindSpeedDataRole **Update Role** Create Role Select

Cancel **Add action**

11. Choisissez Add action (Ajouter une action) pour terminer l'ajout de l'action d'erreur.
12. Cliquez sur la flèche de retour en haut à gauche de la console pour revenir à la page d'accueil de la console AWS IoT Core.

Après avoir configuré l'action d'erreur de republication, vous pouvez afficher les messages d'erreur dans le client de test MQTT dans AWS IoT Core.

Dans la procédure suivante, vous vous abonnez à la rubrique d'erreur dans le client de test MQTT.

Pour vous abonner à la rubrique d'action d'erreur

1. Dans la page de navigation de gauche de la console AWS IoT Core, choisissez Test.
2. Dans le champ Subscription topic (Rubrique Abonnement) saisissez **windspeed/error** et choisissez Subscribe to topic (S'abonner à la rubrique).

AWS IoT

MQTT client ?

Connected as iotconsole-1578083417073-0

Subscriptions

Subscribe to a topic

Publish to a topic

Subscribe

Devices publish MQTT messages on topics. You can use this client to subscribe to a topic and receive these messages.

Subscription topic

windspeed/error **Subscribe to topic**

Max message capture ?

100

3. Vérifiez les messages d'erreur qui s'affichent et explorez la zone **failures** d'un message d'erreur pour diagnostiquer les problèmes courants suivants :
- Fautes de frappe dans l'instruction de requête de règle

- Autorisations de rôle insuffisantes

Si aucune erreur ne s'affiche, vérifiez que votre règle est activée et que vous vous êtes abonné à la même rubrique que celle que vous avez configurée dans l'action d'erreur de republication. Si, malgré cela, aucune erreur ne s'affiche, vérifiez que les ressources de votre parc éolien de démonstration existent toujours et que vous avez activé les notifications sur les propriétés de vitesse du vent. Si vos ressources de démonstration ont expiré et ont disparu AWS IoT SiteWise, vous pouvez créer une nouvelle démo et mettre à jour l'énoncé de la requête de règle pour refléter le modèle et les propriétés des actifs mis à jour IDs.

Résolution des problèmes liés aux opérations d'importation et d'exportation en masse

Pour gérer et diagnostiquer les erreurs produites lors d'une tâche de transfert, consultez l' AWS IoT TwinMaker GetMetadataTransferJobAPI :

1. Après avoir créé et exécuté une tâche de transfert, appelez l'GetMetadataTransferJobAPI :

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

2. L'état de la tâche passe à l'un des états ci-dessous :

- TERMINÉ
- CANCELLED
- ERROR

3. L'GetMetadataTransferJobAPI renvoie un [MetadataTransferJobProgress](#)objet.

4. L'MetadataTransferJobProgressobjet contient les paramètres suivants :

- FailedCount : indique le nombre d'actifs défaillants pendant le processus de transfert.
- SkippedCount : indique le nombre d'actifs ignorés pendant le processus de transfert.
- SucceededCount : indique le nombre d'actifs qui ont réussi pendant le processus de transfert.
- TotalCount : indique le nombre total d'actifs impliqués dans le processus de transfert.

5. En outre, un élément ReportURL est renvoyé par l'appel d'API, qui contient une URL pré-signée. Si votre tâche de transfert comporte des erreurs nécessitant une enquête, vous pouvez télécharger un rapport d'erreur complet à cette adresse URL.

AWS IoT SiteWise points de terminaison et quotas

Les sections suivantes décrivent les points de terminaison et les quotas pour AWS IoT SiteWise.

Rubriques

- [AWS IoT SiteWise points de terminaison](#)
- [AWS IoT SiteWise quotas](#)

AWS IoT SiteWise points de terminaison

Le Références générales AWS guide répertorie les AWS IoT SiteWise points de terminaison d'un Compte AWS. Pour plus d'informations, consultez la section [AWS IoT SiteWise Points de terminaison et quotas](#) dans le Références générales AWS Guide.

AWS IoT SiteWise quotas

Les tableaux suivants décrivent les quotas dans AWS IoT SiteWise. Pour plus d'informations sur les quotas et sur la manière de demander des augmentations de quotas, consultez la section sur [les quotas de AWS service](#) dans le Références générales AWS. Pour plus d'informations sur les AWS IoT SiteWise quotas, consultez la section sur [les quotas de AWS IoT SiteWise service](#) dans le Références générales AWS.

Quotas pour les AWS IoT SiteWise actifs et les modèles d'actifs

Ressource	Description	Quota	Ajustable
Nombre de modèles d'actifs dans chacun Région AWS Compte AWS	Le nombre maximum de modèles de ressources que vous pouvez créer dans un Région AWS pour un Compte AWS.	1 000	Oui
Nombre d'actifs dans chaque modèle d'actifs	Le nombre maximum d'actifs que vous pouvez créer pour	10 000	Oui

Ressource	Description	Quota	Ajustable
	chaque modèle d'actif.		
Nombre d'actifs enfants dans chaque actif parent	Le nombre maximum d'actifs enfants que vous pouvez associer à un actif parent.	2000	Oui
Profondeur de l'arbre hiérarchique du modèle d'actifs	Profondeur maximale de l'arborescence hiérarchique des actifs pour un modèle d'actifs.	30	Oui
Nombre de définitions hiérarchiques dans chaque modèle d'actif	Le nombre maximum de définitions hiérarchiques que vous pouvez avoir dans un modèle d'actifs.	30	Oui
Nombre de propriétés au niveau racine de chaque modèle d'actif	Le nombre maximum de <code>assetModelProperties</code> pour chaque modèle d'actif. Ce décompte n'inclut pas <code>compositeModelProperties</code> . Ce quota s'applique également à tout actif unique créé à partir de ce modèle d'actif.	500	Oui

Ressource	Description	Quota	Ajustable
Nombre de propriétés dans un modèle d'actifs	Le nombre maximum de propriétés d'un modèle d'actif de type <code>ASSET_MODEL</code> ou <code>COMPONENT_MODEL</code> . Ce nombre est déterminé en combinant les propriétés du modèle d'actif racine et de tout modèle composite inclus component-model-based ou en ligne. Ce quota s'applique également à tout actif unique créé à partir de ce modèle d'actif.	5000	Oui
Nombre de propriétés dans chaque modèle composite	Nombre maximal de propriétés autorisées pour les modèles composites. En outre, le nombre maximum de propriétés autorisées pour un modèle d'actif de type <code>COMPONENT_MODEL</code> .	100	Oui

Ressource	Description	Quota	Ajustable
Profondeur de l'arbre des propriétés dans un modèle d'actifs	Par exemple, un modèle avec une propriété de transformation C qui consomme une propriété de transformation B qui consomme une propriété de mesure A a une profondeur de 3.	10	Non
Nombre de modèles d'actifs dans chaque arbre hiérarchique	Nombre maximal de modèles d'actifs que vous pouvez inclure dans une arborescence hiérarchique unique.	100	Oui

Ressource	Description	Quota	Ajustable
Nombre de propriétés directement dépendantes pour un modèle d'actifs	Ce quota limite le nombre de propriétés pouvant dépendre directement d'une propriété unique, tel que défini dans les expressions de formule de propriété . Le nombre de propriétés dépendantes doit être supérieur au nombre de propriétés directement dépendantes pour un modèle d'actifs. Demandez une augmentation pour les deux quotas s'il existe plus de propriétés directement dépendantes que de propriétés dépendantes pour un modèle d'actifs.	20	Oui
Nombre de propriétés dépendantes dans un modèle d'actifs	Ce quota limite le nombre de propriétés pouvant dépendre directement ou indirectement d'une propriété unique, tel que défini dans les expressions de formule de propriété.	30	Non

Ressource	Description	Quota	Ajustable
Nombre de modèles composites dans un modèle d'actifs	Le nombre maximum de modèles composites que vous pouvez avoir sur un seul modèle d'actif.	50	Oui
Profondeur du modèle composite	Profondeur maximale de l'arbre du modèle composite dans chaque modèle d'actif, y compris les modèles en ligne et component-model-based composites.	2	Oui
Nombre de modèles d'actifs uniques utilisant le même modèle de composant	Nombre maximal de modèles d'actifs uniques dotés d'au moins un modèle component-model-based composite faisant directement référence à un modèle d'actif spécifique de type COMPONENT_MODEL.	20	Oui

Ressource	Description	Quota	Ajustable
Nombre de variables de propriété dans une expression de formule de propriété	Par exemple, il existe deux variables de propriété, <code>power</code> et <code>temp</code> , dans l'expression <code>avg(power) + max(temp)</code> . Cela s'applique également aux résultats des calculs de transformation.	10	Non
Nombre de fonctions dans une expression de formule de propriété	Par exemple, il existe deux fonctions, <code>avg</code> et <code>max</code> , dans l'expression <code>avg(power) + max(temp)</code> .	10	Non

Quotas pour les données relatives aux propriétés des AWS IoT SiteWise actifs

Ressource	Description	Quota	Ajustable
Taux de demande pour les opérations d'API de données des ressources	Le nombre maximal de demandes d'API relatives aux données relatives aux propriétés des actifs par seconde que vous pouvez effectuer Région AWS au cours de chaque seconde Compte AWS. Ce quota s'applique aux opérations d'API telles que <code>GetAssetP</code>	1 000	Oui

Ressource	Description	Quota	Ajustable
	propertyValue et BatchPutAssetPropertyValue .		
Nombre de points de données par seconde pour chaque qualité de données pour chaque propriété d'actif	Ce quota s'applique au nombre maximum de points de données timestamp-quality-value (TQV) avec le même horodatage en secondes pour chaque qualité de données pour chaque propriété d'actif. Vous pouvez stocker jusqu'à ce nombre de points de données de bonne qualité, de qualité incertaine et de mauvaise qualité par seconde pour chaque propriété d'actif.	10	Non

Ressource	Description	Quota	Ajustable
Nombre d'BatchPutAssetPropertyValues entrées ingérées chaque seconde dans chaque propriété d'actif pour chacune d'elles Région AWS pour un Compte AWS.	Le nombre maximum d'entrées dans chaque propriété d'actif pour toutes les BatchPutAssetPropertyValues sources, y compris les passerelles SiteWise Edge, AWS IoT Core les règles et les appels d'API.	10	Non
Taux de points de données ingérés	Le nombre maximum de points de données timestamp-quality-value (TQV) ingérés par seconde dans chacun d'eux Région AWS pendant un. Compte AWS	5000	Oui
Tarif demandé pour BatchGetAssetPropertyAggregates	Le nombre maximum de BatchGetAssetPropertyAggregates demandes que vous pouvez effectuer par seconde Région AWS dans chaque seconde Compte AWS.	200	Oui

Ressource	Description	Quota	Ajustable
Tarif demandé pour BatchGetAssetPropertyValues	Le nombre maximum de BatchGetAssetPropertyValues demandes que vous pouvez effectuer par seconde Région AWS dans chaque seconde Compte AWS.	500	Oui
Tarif demandé pour BatchGetAssetPropertyHistory	Le nombre maximal de BatchGetAssetPropertyHistory demandes que vous pouvez effectuer par seconde.	200	Oui
Nombre d'BatchPutAssetPropertyValues entrées ingérées chaque seconde pour chaque propriété d'actif pour chacune d'entre elles Région AWS dans un Compte AWS.	Ce quota s'applique aux entrées dans chaque propriété d'actif pour toutes les BatchPutAssetPropertyValues sources, y compris les passerelles SiteWise Edge, AWS IoT Core les règles et les appels d'API.	10	Non

Ressource	Description	Quota	Ajustable
Taux de GetAssetPropertyAggregates et de demandes de BatchGetAssetPropertyAggregates saisie pour chaque propriété d'actif	Le nombre total maximum de GetAssetPropertyAggregates et d'BatchGetAssetPropertyAggregates entrées pour chaque propriété d'actif par seconde Région AWS dans chacune d'elles Compte AWS.	50	Non
Taux de GetAssetPropertyValue demandes et de demandes de BatchGetAssetPropertyValue saisie pour chaque propriété d'actif	Le nombre total maximum de GetAssetPropertyValue demandes et d'BatchGetAssetPropertyValue entrées pour chaque propriété d'actif par seconde Compte AWS. Région AWS	500	Non

Ressource	Description	Quota	Ajustable
Taux de GetAssetPropertyValueHistory demandes et de demandes de BatchGetAssetPropertyHistory saisie pour chaque propriété d'actif	Le nombre total maximum de GetAssetPropertyValueHistory demandes et d'BatchGetAssetPropertyHistory entrées pour chaque propriété d'actif par seconde Compte AWS. Région AWS	30	Non
Taux de GetInterpolatedAssetPropertyValues demandes	Le nombre maximum de GetInterpolatedAssetPropertyValues demandes que vous pouvez effectuer par seconde Région AWS dans chaque seconde Compte AWS.	500	Oui
Nombre de résultats pour chaque GetInterpolatedAssetPropertyValues demande	Le nombre maximum de résultats à renvoyer pour une GetInterpolatedAssetPropertyValues demande paginée.	10	Oui

Ressource	Description	Quota	Ajustable
Taux de points de données extraits de GetAssetPropertyValueHistory et BatchGetAssetPropertyHistory	Le débit d'octets maximal (Mo/seconde) des points de données récupérés chaque seconde pour chacun des points compris Région AWS dans un point transversal. Compte AWS GetAssetPropertyValueHistory BatchGetAssetPropertyHistory La charge utile de réponse évaluée pour ce quota utilise des champs Timestamp-Quality-Value (TQV) pour chaque point de données et arrondit la taille en octets de chaque demande d'API au prochain incrément de 4 Ko. Timestamp-quality-value (TQV) les points de données récupérés chaque seconde varient pour chaque type de données :	100	Oui

Ressource	Description	Quota	Ajustable
	• Nombre entier : jusqu'à 5 millions de TQV par seconde • Double : jusqu'à 4 millions de TQV par seconde • Booléen : jusqu'à 6 millions de TQV par seconde • Chaîne : varie en fonction de la taille de chaque valeur de chaîne.		

Quotas pour les passerelles SiteWise Edge

Ressource	Description	Quota	Ajustable
Nombre de passerelles SiteWise Edge dans chacune Région AWS pour un Compte AWS	Le nombre maximum de passerelles SiteWise Edge que vous pouvez créer dans un Région AWS pour un Compte AWS.	100	Oui
Nombre de sources OPC UA dans une passerelle SiteWise Edge	Le nombre maximum de sources OPC UA que vous pouvez configurer dans une passerelle SiteWise Edge.	100	Non

Ressource	Description	Quota	Ajustable
Nombre total de destinations dans une passerelle SiteWise Edge	Le nombre maximum de destinations que vous pouvez configurer dans une passerelle SiteWise Edge.	100	Non

Quotas pour AWS IoT SiteWise Monitor

Ressource	Description	Quota	Ajustable
Nombre de portails dans chacun Région AWS pour un Compte AWS	Le nombre maximum de portails de SiteWise surveillance que vous pouvez créer dans et Région AWS pour un Compte AWS.	100	Oui
Nombre de projets sur un portail	Le nombre maximum de projets que vous pouvez créer dans un portail SiteWise Monitor.	100	Oui
Nombre de tableaux de bord dans un projet	Le nombre maximum de tableaux de bord que vous pouvez créer dans un projet dans SiteWise Monitor.	100	Oui
Nombre de ressources principales dans un projet	Le nombre maximum de ressources de haut niveau que vous	1	Non

Ressource	Description	Quota	Ajustable
	pouvez ajouter à un projet dans SiteWise Monitor.		
Nombre de visualisations dans un tableau de bord	Nombre maximal d'éléments visuels (tels que des diagrammes, des graphiques ou des tableaux) que vous pouvez ajouter à un tableau de bord dans SiteWise Monitor.	10	Oui
Nombre de mesures dans chaque visualisation de tableau de bord	Le nombre maximal de mesures ou de points de données que vous pouvez afficher dans une seule visualisation sur un tableau de bord dans SiteWise Monitor.	5	Oui
Nombre de seuils pour chaque visualisation du tableau de bord	Le nombre maximal de niveaux de seuil que vous pouvez définir pour chaque visualisation sur un tableau de bord dans SiteWise Monitor.	12	Non

Quotas pour l'importation et l'exportation AWS IoT SiteWise groupées de métadonnées

Ressource	Description	Quota	Ajustable
Nombre de tâches de transfert de métadonnées en attente	Nombre maximal de tâches de transfert de PENDING métadonnées dans la file d'attente.	10	Oui
Taille du fichier d'importation des tâches de transfert de métadonnées	Taille maximale du fichier importé (en Mo).	100	Oui
Nombre de ressources AWS IoT SiteWise d'importation dans une tâche	Le nombre maximum de ressources AWS IoT SiteWise d'importation dans une seule tâche. Une ressource inclut des actifs et des modèles d'actifs.	5000	Oui
Nombre de ressources AWS IoT SiteWise d'exportation associées à un emploi	Le nombre maximum de ressources d'AWS IoT SiteWise exportation par tâche. Une ressource inclut des actifs et des modèles d'actifs.	5000	Oui

Quotas pour l'importation AWS IoT SiteWise massive de données

Ressource	Description	Quota	Ajustable
Nombre de tâches d'importation en bloc en cours d'exécution	Nombre maximal de tâches d'importation en bloc pouvant être exécutées simultanément.	100	Non
Taille du fichier CSV	Taille maximale du fichier CSV (en Go) dans une tâche d'importation en bloc.	10	Non
Taille du fichier de parquet non compressé	Taille de fichier maximale (en Mo) pour un fichier parquet non compressé dans le cadre d'une tâche d'importation groupée.	256 Mo	Non
Taille du fichier CSV pour l'ingestion en mémoire tampon	Taille maximale du fichier CSV (en Mo) lors de l'utilisation de l'ingestion en mémoire tampon dans le cadre d'une tâche d'importation en masse.	256 Mo	Non
Taille du groupe de rangées de parquet non compressé	Taille maximale d'un groupe de rangées de parquet non compressé.	64 MO	Non

Ressource	Description	Quota	Ajustable
Nombre de mesures uniques dans un dossier de parquet	Le nombre maximum de mesures uniques dans un dossier de parquet.	10 000	Non
Nombre de jours entre l'horodatage antérieur et actuel pour l'ingestion en mémoire tampon	Le nombre maximal de jours entre un horodatage antérieur et la date d'aujourd'hui lors de l'utilisation d'une ingestion en mémoire tampon.	30	Oui
Taux de CreateBulkImportJobs demande pour chaque Région AWS Compte AWS		10	Oui
Taux de demande ListBulkImportJobs pour chacun Région AWS Compte AWS		50	Oui
Taux de demande DescribeBulkImportJobs pour chacun Région AWS Compte AWS		50	Oui

Quotas pour la limitation de l'API AWS IoT SiteWise Assistant

Quotas pour les AWS IoT SiteWise limites de limitation de l'API Assistant

Ressource	Description	Quota	Ajustable
Taux de demandes d'InvokeAssistant exploitation	Le nombre maximum de transactions par minute (TPM) qui peuvent être effectuées vers l' AWS IoT SiteWise InvokeAssistant API dans un Compte AWS. Les limites du TPM s'appliquent à toutes les régions prises en charge et sont ajustables dans certaines régions.	10	Non

Quotas pour la détection des anomalies

Les quotas de détection des anomalies sont partagés entre Amazon Lookout for Equipment AWS IoT SiteWise et Amazon Lookout for Equipment. Pour plus d'informations, consultez la section [Quotas d'utilisation de Lookout for Equipment](#).

Historique du document pour le guide de AWS IoT SiteWise l'utilisateur

Le tableau suivant décrit la documentation de cette version de AWS IoT SiteWise.

- Version de l'API : 2019-12-02

Modification	Description	Date
Connectez des applications externes au broker EMQX à l'aide de noms d'utilisateur et de mots de passe	Ajout de la prise en charge de la connexion d'applications externes à l'aide de noms d'utilisateur et de mots de passe au broker AWS IoT Greengrass EMQX qui est déployé par défaut lors de la création d'une nouvelle passerelle V3 compatible MQTT.	1er mai 2025
Support des passerelles V3 compatibles MQTT sur Edge SiteWise	Ajout de nouvelles fonctionnalités et suppression du contenu obsolète • Ajout du support pour les passerelles V3 compatibles MQTT. Configuration de destination améliorée à l'aide de filtres de chemin pour s'abonner aux rubriques MQTT, y compris l'ingestion de données en temps réel directement AWS IoT SiteWise ou l'ingestion de données mise en	26 février 2025

mémoire tampon à l'aide d'Amazon S3.

- Publication de la version 3.0.0 du collecteur IoT SiteWise OPC UA et de la version 4.0.0 du composant SiteWise éditeur IoT pour AWS IoT Greengrass V2
- La version précédente des passerelles SiteWise Edge auto-hébergées a été renommée en flux classiques, passerelles V2.
- Les références AWS IoT Greengrass V1 à ce terme ont été supprimées dans la documentation SiteWise Edge, car son utilisation avec AWS IoT SiteWise.

[Prise en charge pour AWS IoT SiteWise Assistant](#)

Ajout du support pour l' AWS IoT SiteWise assistant, un assistant génératif alimenté par l'IA.

18 novembre 2024

[Ajout de délais d'expiration de session configurables pour Edge SiteWise APIs](#)

Ajout de paramètres de délai d'expiration de session configurables pour gérer les périodes d'inactivité pour AWS OpsHub et SiteWise Edge APIs

31 octobre 2024

[Ajout de paramètres de proxy configurables pour SiteWise Edge APIs](#)

Ajout de la gestion des informations du trust store pour permettre la prise en charge du proxy HTTPS pour les passerelles SiteWise Edge.

31 octobre 2024

<u>Activer CORS sur Edge SiteWise APIs</u>	Ajout du support CORS pour SiteWise Edge afin APIs de permettre un accès sécurisé aux applications Web interdomaines.	30 septembre 2024
<u>Support pour CloudRail les sources de données Litmus Edge partenaires</u>	Support supplémentaire pour les deux CloudRail et Litmus Edge en tant que sources de données partenaires.	5 septembre 2024
<u>Disponibilité générale pour exécuter SiteWise Edge sur Siemens Industrial Edge</u>	AWS IoT SiteWise prend désormais en charge la disponibilité générale de Running SiteWise Edge sur les appareils Siemens Industrial Edge.	24 juillet 2024
<u>Ajout du support pour la configuration de l'horodatage sur les sources de données OPC UA</u>	AWS IoT SiteWise prend désormais en charge la configuration d'horodatage pour les sources de données OPC UA.	24 juillet 2024
<u>Ajout du support pour la conversion des types de données sur les sources de données OPC UA</u>	AWS IoT SiteWise prend désormais en charge la conversion de type de données pour les types de données OPC UA non pris en charge.	24 juillet 2024
<u>Ajout de la prise en charge de l'exécution d'une version préliminaire d' SiteWise Edge sur Siemens Industrial Edge</u>	AWS IoT SiteWise prend désormais en charge l'exécution d'une version préliminaire d' SiteWise Edge sur les appareils Siemens Industrial Edge.	26 novembre 2023

<u>Support supplémentaire pour le stockage à chaud</u>	AWS IoT SiteWise prend désormais en charge le stockage à chaud, un niveau de stockage entièrement géré qui permet aux clients de stocker et d'accéder facilement en toute sécurité aux données industrielles.	15 novembre 2023
<u>Ajout de la prise en charge des identifiants uniques définis par l'utilisateur</u>	AWS IoT SiteWise prend désormais en charge l'utilisation d'identifiants uniques définis par l'utilisateur pour les actifs, les modèles d'actifs, les propriétés et les hiérarchies.	15 novembre 2023
<u>Ajout de la prise en charge de la détection d'anomalies à plusieurs variables dans les actifs industriels</u>	AWS IoT SiteWise prend désormais en charge la détection des anomalies multidimensionnelles des actifs industriels en intégrant les données historiques et en temps réel des équipements à Amazon Lookout for Equipment.	15 novembre 2023
<u>Support supplémentaire pour une ingestion rentable et évolutive de données de séries chronologiques dans AWS IoT SiteWise</u>	AWS IoT SiteWise prend désormais en charge l'ingestion rentable et évolutive des données chronologiques nécessaires aux cas d'utilisation analytiques.	15 novembre 2023

<u>Ajout du support pour l'importation, l'exportation et la mise à jour en masse</u>	AWS IoT SiteWise prend désormais en charge l'importation, l'exportation et la mise à jour en masse des métadonnées des équipements industriels.	15 novembre 2023
<u>Ajout de la prise en charge des composants du modèle d'actifs</u>	AWS IoT SiteWise prend désormais en charge les composants du modèle Asset pour aider les clients industriels à créer des composants réutilisables.	15 novembre 2023
<u>Ajout de la prise en charge de l'application de tableau de bord IoT</u>	AWS IoT SiteWise prend désormais en charge une application de tableau de bord open source dans laquelle vous pouvez visualiser et interagir avec les données opérationnelles.	15 novembre 2023
<u>Mise à jour des rôles liés aux services pour AWS IoT SiteWise</u>	AWS IoT SiteWise possède de nouveaux rôles liés aux services et peut exécuter une requête de recherche de métadonnées dans la AWS IoT TwinMaker base de données.	6 novembre 2023
<u>Balisateur mis à jour pour les ressources AWS IoT SiteWise de flux de données</u>	Ajout de la prise en charge du balisateur des ressources de flux de données.	18 août 2022

Passerelles SiteWise Edge mises à jour	Vous pouvez désormais configurer l'éditeur pour contrôler les données envoyées de la périphérie vers le cloud et l'ordre dans lequel elles sont envoyées vers le cloud.	12 janvier 2022
Mise à jour de la AWS IoT SiteWise démo	Vous pouvez désormais utiliser la démo pour créer un portail SiteWise Monitor.	10 janvier 2022
Gestion du stockage actualisée	Vous pouvez désormais définir une période de conservation pour contrôler la durée de conservation de vos données dans le hot tier.	29 novembre 2021
Support supplémentaire pour la gestion des flux de données	Vous pouvez désormais ingérer des données AWS IoT SiteWise avant de créer des modèles d'actifs et des actifs.	24 novembre 2021
Hiérarchies de modèles d'actifs mises à jour	Un modèle d'actif enfant peut désormais être associé à plusieurs modèles d'actifs parents.	28 octobre 2021
Lancement de la région	Lancé AWS IoT SiteWise en AWS GovCloud (ouest des États-Unis).	29 septembre 2021

Fonctions mises à jour	Ajout des fonctionnalités suivantes	10 août 2021
	- Dans les métriques, vous pouvez utiliser des expressions imbriquées dans les fonctions d'agrégation et les fonctions temporelles. - Dans les transformations, vous pouvez utiliser la fonction pretrigger () pour récupérer la valeur d'une variable avant la mise à jour de la propriété qui a déclenché le calcul de transformation en cours.	
Intervalle de temps métrique personnalisé	Ajout de la prise en charge des intervalles de temps personnalisés et des décalages dans les métriques.	3 août 2021
Utilisation AWS IoT SiteWise sur le bord	La fonction de traitement des bords est désormais disponible pour tous.	29 juillet 2021
Exportation de données vers Amazon S3	AWS IoT SiteWise peut désormais exporter des données vers Amazon S3.	27 Juillet 2021
Points de terminaison VPC ()AWS PrivateLink	Le point de terminaison VPC de l'interface pour les opérations de l'API du plan de contrôle est désormais généralement disponible.	15 juillet 2021

Transforme	Les transformations peuvent désormais saisir plusieurs variables de propriétés d'actifs.	8 juillet 2021
Mise à jour de la fonction timestamp ()	Dans les transformations, vous pouvez désormais fournir une variable comme argument à la timestamp() fonction.	16 juin 2021
Disponibilité générale des alarmes	La fonction d'alarmes est désormais disponible pour tous.	27 mai 2021
Sortie de la version 2 de l'adaptateur de protocole Modbus-TCP	La version 2 du connecteur adaptateur de protocole Modbus-TCP est disponible. Cette version a ajouté le support pour les chaînes source ASCII et ISO8859 codées. UTF8	24 mai 2021
Quotas de service actualisés	Ajout des quotas suivants pour l' GetInterpolatedAssetPropertyValues API : taux de GetInterpolatedAssetPropertyValues demandes, nombre de résultats par GetInterpolatedAssetPropertyValues demande et nombre de jours entre la date de début passée et la date d'aujourd'hui pour GetInterpolatedAssetPropertyValues .	29 avril 2021

[Expressions de formule mises à jour](#)

Les opérateurs et fonctions suivants ont été ajoutés :

22 avril 2021

- Les [opérateurs](#) suivants ont été ajoutés : < > <=, >=, ==, !=, !=, and, or, et not.
- La [fonction de comparaison](#) suivante a été ajoutée : neq(x, y).
- Les [fonctions de chaîne](#) suivantes ont été ajoutées : join(), format(), et f ' '.

[Points de terminaison VPC \(AWS PrivateLink\)](#)

Ajout d'informations sur la façon d'établir une connexion privée entre votre cloud privé virtuel (VPC) et le plan de AWS IoT SiteWise contrôle en APIs créant un point de terminaison VPC d'interface.

16 mars 2021

[Fédération IAM](#)

Les administrateurs et utilisateurs de votre portail SiteWise Monitor peuvent désormais se connecter aux portails qui leur sont assignés avec leurs informations d'identification IAM.

16 mars 2021

[Lancement de la région](#)

Lancé AWS IoT SiteWise en Chine (Pékin).

3 février 2021

Sortie de la version 10 du SiteWise connecteur IoT	La version 10 du SiteWise connecteur IoT est disponible. Cette version est configurée StreamManager pour améliorer la gestion lorsque la connexion source est perdue puis rétablie. Cette version accepte également les valeurs OPC UA avec un ServerTimestamp lorsque non SourceTimestamp est disponible.	22 janvier 2021
Fonctions de date et d'heure	AWS IoT SiteWise prend désormais en charge les fonctions de date et d'heure.	21 janvier 2021
Syntaxe des fonctions	Vous pouvez désormais utiliser la syntaxe UFCS (Uniform Function Call Syntax) pour les AWS IoT SiteWise fonctions.	11 janvier 2021
Intégration à Grafana	Ajout d'informations sur la façon de visualiser les AWS IoT SiteWise données dans les tableaux de bord Grafana.	15 décembre 2020

[AWS IoT SiteWise sortie de fonctionnalités](#)

15 décembre 2020

Vous pouvez désormais surveiller vos données à l'aide d'alarmes, traiter les données industrielles en périphérie, utiliser Modbus TCP et les Ethernet/IP sources vers votre passerelle SiteWise Edge, filtrer les données entrantes avec des zones mortes, etc.

- Ajout de la section [Surveillance des données avec alarmes](#) que vous pouvez utiliser pour définir, configurer et répondre aux alarmes AWS IoT SiteWise.
- Ajout de la section [Traitement Edge](#) que vous pouvez utiliser pour configurer le traitement de vos données industrielles sur vos appareils Edge.
- Les sections [Modbus TCP et EtherNet/IP ont été ajoutées à la documentation](#) source de la passerelle SiteWise Edge.
- Ajout de la section [source et destination](#) que vous pouvez utiliser pour personnaliser l'endroit où vous envoyez vos données industrielles entrantes.
- Ajout de la section de [filtrage OPC UA](#) que vous pouvez utiliser pour

contrôler la fréquence et le type de données envoyées à votre passerelle SiteWise Edge depuis votre serveur local industriel.

[AWS IoT SiteWise prend désormais en charge la gestion par le client CMKs.](#)

AWS IoT SiteWise prend désormais en charge le chiffrement géré par le client CMKs.

24 novembre 2020

[Sortie de la version 8 du SiteWise connecteur IoT](#)

La version 8 du SiteWise connecteur IoT est disponible. Cette version améliore la stabilité lorsque le connecteur est confronté à une connectivité réseau intermittente.

19 novembre 2020

[Utilisation de chaînes et de conditions dans les expressions de formule](#)

Ajout d'informations sur la façon d'utiliser des chaînes et des fonctions conditionnelles dans les expressions de formule pour les transformations et les métriques.

16 novembre 2020

[Ingestion de données à l'aide du gestionnaire de AWS IoT Greengrass flux](#)

Ajout d'informations sur la façon d'ingérer de gros volumes de données IoT à partir de sources de données locales à l'aide d'un appareil AWS IoT Greengrass périphérique.

16 septembre 2020

Points de terminaison VPC ()AWS PrivateLink	Ajout d'informations sur la façon d'établir une connexion privée entre votre cloud privé virtuel (VPC) et les AWS IoT SiteWise données en APIs créant un point de terminaison VPC d'interface.	4 septembre 2020
Sortie de la version 7 du SiteWise connecteur IoT	La version 7 du SiteWise connecteur IoT est disponible. Cette version résout un problème lié aux métriques de la passerelle SiteWise Edge.	14 août 2020
Création d'utilisateurs IAM Identity Center depuis la console AWS IoT SiteWise	Ajout d'informations sur la façon dont vous pouvez créer des utilisateurs IAM Identity Center dans la AWS IoT SiteWise console. Vous pouvez désormais créer des utilisateurs IAM Identity Center lorsque vous attribuez des utilisateurs à un portail nouveau ou existant. Mise à jour du didacticiel sur la visualisation et le partage de données de parcs éoliens pour utiliser cette fonctionnalité. Cette modification réduit le nombre d'étapes du didacticiel.	4 août 2020
Résolution améliorée des problèmes liés à la passerelle SiteWise Edge	Ajout d'informations supplémentaires sur le dépannage d'une passerelle SiteWise Edge et l'exportation du certificat client OPC UA pour une source.	18 juin 2020

[Documentation des tâches liées à la console](#)

Ajout de la documentation des tâches de la console pour [Modélisation des ressources industrielles](#), [Interrogation des données de propriété de ressource](#) et [Interaction avec d'autres services](#). Vous pouvez suivre ces instructions pour effectuer des tâches dans la console AWS IoT SiteWise .

11 juin 2020

[Tutoriel d'analyse des données exportées](#)

Ajout d'un didacticiel que vous pouvez suivre pour apprendre à utiliser Amazon Athena pour analyser les données d'actifs que vous avez exportées vers Amazon S3 à l'aide du modèle de [fonctionnalité AWS CloudFormation d'exportation](#).

27 mai 2020

[Amélioré à l'aide d'expressions de formule](#)

Ajout d'informations détaillées sur le comportement des propriétés des AWS IoT SiteWise formules et ajout d'un exemple expliquant comment compter les points de données filtrés.

18 mai 2020

[Sortie de la version 6 du SiteWise connecteur IoT](#)

La version 6 du SiteWise connecteur IoT est disponible. Cette version ajoute la prise en charge des CloudWatch métriques et de la découverte automatique des nouvelles balises OPC UA. Cela signifie que vous n'avez pas besoin de redémarrer votre passerelle SiteWise Edge lorsque les balises de vos sources OPC UA changent. Cette version du connecteur nécessite le gestionnaire de flux et le logiciel AWS IoT Greengrass Core v1.10.0 ou supérieur.

29 avril 2020

[AWS IoT SiteWise sortie de fonctionnalités](#)

29 avril 2020

AWS IoT SiteWise sortie de fonctionnalité. Vous pouvez désormais gérer les passerelles SiteWise Edge avec l'API, ajouter votre logo aux portails, consulter les statistiques des passerelles SiteWise Edge, etc.

- La section [Exportation de données vers Amazon S3](#) a été ajoutée avec un AWS CloudFormation modèle que vous pouvez utiliser pour exporter de nouvelles valeurs de données vers un compartiment Amazon S3.
- Ajout de la section [Configuration des sources de données](#) qui améliore la documentation source de la passerelle SiteWise Edge et inclut la nouvelle passerelle SiteWise Edge APIs.
- Ajout de la section des métriques de passerelle SiteWise Edge qui décrit les CloudWatch métriques publiées par les passerelles SiteWise Edge.
- Ajout de la EC2 section Configuration d'une passerelle SiteWise Edge sur Amazon avec un AWS CloudFormation modèle que vous pouvez utiliser

pour configurer rapidement les dépendances d'une passerelle SiteWise Edge sur une EC2 instance Amazon.

- Ajout de la section sur [les rôles des services de portail](#) qui décrit la nouvelle fonctionnalité d'autorisation des portails SiteWise Monitor.
- Mise à jour [de la documentation](#) du portail pour les rôles de service du portail et les logos du portail.
- Ajout de la section [Marquage de vos AWS IoT SiteWise ressources](#).
- Mise à jour de la section [Création de tableaux de bord \(CLI\)](#) pour la nouvelle structure de définition de tableau de bord.
- Ajout de la section [Sécurité](#).

[Ingestion de données provenant de AWS IoT Events](#)

Ajout d'informations sur la façon d'ingérer des données à partir du AWS IoT Events moment où un événement se produit.

20 avril 2020

[Visualisation et partage de données de parcs éoliens dans le didacticiel SiteWise Monitor](#)

Ajout d'un didacticiel que vous pouvez suivre pour apprendre à visualiser et AWS IoT SiteWise Monitor à partager les données des actifs.

12 mars 2020

AWS IoT SiteWise concepts	Ajout d'un glossaire de AWS IoT SiteWise concepts que vous pouvez utiliser pour en savoir plus sur le service et ses termes courants.	5 mars 2020
Instructions AWS IoT Greengrass d'installation supprimées	Les instructions d'installation du logiciel AWS IoT Greengrass Core ont été supprimées du guide de AWS IoT SiteWise l'utilisateur. Le guide du AWS IoT Greengrass développeur propose un script de configuration de l'appareil et des instructions à configurer AWS IoT Greengrass sur d'autres plateformes telles qu'Amazon EC2 et Docker.	14 février 2020
Amélioration de l'ingestion de données à l'aide de règles AWS IoT Core	Ajout d'informations détaillées sur la façon d'utiliser et de résoudre les problèmes liés à l'action de la AWS IoT SiteWise règle, que vous pouvez utiliser pour ingérer des données provenant de messages MQTT via. AWS IoT Core	14 février 2020
Sortie de la version 5 du SiteWise connecteur IoT	La version 5 du SiteWise connecteur IoT est disponible. Cette version corrige un problème de compatibilité avec le logiciel AWS IoT Greengrass Core v1.9.4.	12 février 2020

[Sortie de la version 4 du SiteWise connecteur IoT](#)

La version 4 du SiteWise connecteur IoT est disponible. Cette version corrige un problème de reconnexion au serveur OPC UA.

7 février 2020

[Modélisation restructurée des actifs industriels](#)

Restructuration de la section sur la mise à jour des ressources et des modèles en plusieurs rubriques dans la section sur la modélisation des ressources industrielles.

4 février 2020

- [État des ressources et des modèles](#)
- [Gérez les flux de données pour AWS IoT SiteWise](#)
- [Mettre à jour les valeurs des attributs](#)
- [Associer et dissocier des actifs](#)
- [Mettre à jour les actifs et les modèles](#)
- [Supprimer des actifs et des modèles dans AWS IoT SiteWise](#)

[Tutoriel sur l'ingestion de données depuis AWS IoT des objets](#)

Ajout d'un didacticiel que vous pouvez suivre pour apprendre à configurer une action de AWS IoT SiteWise règle pour ingérer des données provenant d'un parc d' AWS IoT objets nouveau ou existant.

4 février 2020

Extraction de données restructurée à partir de AWS IoT SiteWise	Restructuration de la section Extraction des données en deux sections de niveau supérieur : Interaction avec les valeurs et les agrégats des propriétés des actifs et interaction avec d'autres services. AWS	21 janvier 2020
Publication de mises à jour de la valeur des propriétés dans le didacticiel Amazon DynamoDB	Ajout d'un didacticiel que vous pouvez suivre pour apprendre à utiliser les notifications de valeur de propriété pour stocker les données des actifs dans DynamoDB.	8 janvier 2020
Utilisation d'expressions de formule	Ajout de la référence d'expression de formule pour l'organisation des constantes et des fonctions disponibles pour une utilisation dans les propriétés de transformation et de métriques Restructuration de la section sur les propriétés de ressource en rubriques distinctes pour chaque type de propriété.	7 janvier 2020
Utilisation des filtres de nœuds OPC UA	Ajout d'informations sur l'utilisation des filtres de nœuds OPC UA pour améliorer les performances de la passerelle SiteWise Edge lors de l'ajout de sources de passerelle SiteWise Edge.	3 janvier 2020

Mise à niveau d'un connecteur	Ajout d'informations sur la mise à niveau d'une passerelle SiteWise Edge lorsqu'une nouvelle version du connecteur est publiée.	30 décembre 2019
Sortie de la version 3 du SiteWise connecteur IoT	La version 3 du SiteWise connecteur IoT est disponible. Cette version supprime l'exigence d'autorisations <code>iot:*</code> .	17 décembre 2019
Sortie de la version 2 du SiteWise connecteur IoT	La version 2 du SiteWise connecteur IoT est disponible. Cette version ajoute la prise en charge de plusieurs ressources secrètes OPC UA.	10 décembre 2019
Création de tableaux de bord (AWS CLI)	Ajout d'informations sur la création d'un tableau de bord à AWS IoT SiteWise Monitor l'aide du AWS CLI.	6 décembre 2019

[AWS IoT SiteWise sortie de la version 2](#)

Aperçu publié pour la version 2 de AWS IoT SiteWise. Vous pouvez désormais ingérer des données via OPC UA, MQTT et HTTP, modéliser vos données dans des hiérarchies d'actifs et visualiser vos données avec Monitor. SiteWise

2 décembre 2019

- Réécriture de la section relative à la [modélisation des ressources](#) pour les modifications apportées aux ressources, aux modèles de ressources et aux hiérarchies de ressources.
- Mise à jour de la section [d'ingestion de données](#) pour inclure les étapes AWS IoT Greengrass du connecteur et les sections d'ingestion de données non liées à la passerelle.
- Ajout de la [AWS IoT SiteWise Monitor](#) section et d'un [guide d'application distinct](#) qui montre comment utiliser l'application Web SiteWise Monitor.
- Ajout des sections [Interrogez les données de AWS IoT SiteWise](#) et [Interagissez avec d'autres AWS services](#).
- Réécriture de la section relative à la [mise en route](#)

pour correspondance avec
l'expérience de démonstra
tion mise à jour.

[AWS IoT SiteWise sortie de la
version 1](#)

Aperçu initial publié pour
la version 1 de AWS IoT
SiteWise.

25 février 2019

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.