Guide de l'utilisateur

AWS IoT Analytics

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

AWS IoT Analytics: Guide de l’utilisateur

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

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

Table of Contents

Qu'est-ce que c'est AWS IoT Analytics ? ... 1
Comment utiliser AWS IoT Analytics ... 1
Fonctions principales ... 2
AWS IoT Analytics composants et concepts ... 4
Accès AWS IoT Analytics ... 7
Cas d'utilisation ... 8
AWS IoT Analytics fin du support ... 9
Options de migration ... 9
Guide de migration ... 13
Étape 1 : Rediriger l'ingestion de données en cours ... 14
Étape 2 : Exporter les données précédemment ingérées ... 16
Exécutez des requêtes à la demande pour les deux modèles ... 24
Résumé ... 24
Démarrer (console) ... 26
Connectez-vous à la AWS IoT Analytics console ... 27
Création d'une chaîne ... 27
Création d'un magasin de données ... 29
Crée un pipeline. ... 30
Créer un jeu de données ... 32
Envoyer des données de message avec AWS IoT ... 34
Vérifiez la progression des AWS IoT messages ... 36
Accéder aux résultats des requêtes ... 36
Explorez vos données ... 37
Modèles de bloc-notes ... 39
Prise en main ... 41
Création d'un canal ... 41
Création d'un magasin de données ... 43
Politiques Amazon S3 ... 43
Formats de fichier ... 45
Cloisons personnalisées ... 49
Création d'un pipeline ... 52
Ingestion de données pour AWS IoT Analytics ... 53
Utilisation du courtier de AWS IoT messages ... 53
Utilisation de l' BatchPutMessage API ... 58

Surveillance des données ingérées ... 59
Création d'un jeu de données ... 61
Interrogation de données ... 62
Accès aux données demandées ... 62
Exploration AWS IoT Analytics des données ... 37
Amazon S3 ... 64
AWS IoT Events ... 64
Suite rapide ... 65
Bloc-notes Jupyter ... 65
Conservation de plusieurs versions d'ensembles de données ... 65
Syntaxe de la charge utile des messages ... 66
Travailler avec des AWS IoT SiteWise données ... 67
Créer un jeu de données ... 68
Accéder au contenu du jeu de données ... 71
Tutoriel : AWS IoT SiteWise données de requête ... 73
Activités liées au pipeline ... 82
Activité de la chaîne ... 82
Activité de la banque de données ... 82
AWS Lambda activité ... 83
Exemple de fonction Lambda 1 ... 83
Exemple de fonction Lambda 2 ... 86
AddAttributes activité ... 87
RemoveAttributes activité ... 88
SelectAttributes activité ... 89
Activité du filtre ... 90
DeviceRegistryEnrich activité ... 90
DeviceShadowEnrich activité ... 92
Activité mathématique ... 95
Opérateurs et fonctions d'activités mathématiques ... 95
RunPipelineActivity ... 112
Retraitement des messages du canal ... 114
Paramètres ... 114
Retraitement des messages du canal (console) ... 115
Retraitement des messages du canal (API) ... 116
Annulation des activités de retraitement des canaux ... 117
Automatiser votre flux de travail ... 118

Cas d'utilisation ... 119
Utilisation d'un conteneur Docker ... 120
Variables de conteneur input/output Docker personnalisées ... 123
Permissions ... 125
CreateDataset (Java et AWS CLI) ... 128
Exemple 1 : création d'un jeu de données SQL (Java) ... 128
Exemple 2 : création d'un jeu de données SQL avec une fenêtre delta (Java) ... 129
Exemple 3 : création d'un jeu de données conteneur avec son propre déclencheur de planification (Java) ... 130
Exemple 4 : création d'un ensemble de données conteneur avec un ensemble de données SQL comme déclencheur (Java) ... 131
Exemple 5 : création d'un jeu de données SQL (CLI) ... 132
Exemple 6 : création d'un jeu de données SQL avec une fenêtre delta (CLI) ... 133
Conteneurisation d'un bloc-notes ... 134
Activer la conteneurisation des instances de bloc-notes non créées via la console AWS IoT Analytics ... 135
Mettez à jour l'extension de conteneurisation de votre bloc-notes ... 138
Création d'une image conteneurisée ... 138
Utilisation d'un conteneur personnalisé ... 143
Visualisation des données ... 152
Visualisation (console) ... 152
Visualisation () QuickSight ... 153
Identification ... 157
Principes de base des étiquettes ... 157
Utilisation des balises avec des politiques IAM ... 158
Restrictions liées aux étiquettes ... 161
Expressions SQL ... 162
Fonctionnalité SQL prise en charge ... 163
Types de données pris en charge ... 163
Fonctions prises en charge ... 164
Résoudre les problèmes courants ... 165
Sécurité ... 166
Gestion des identités et des accès AWS ... 166
Public ciblé ... 166
Authentification par des identités ... 167
Gestion des accès ... 168

Travailler avec IAM ... 169
Prévention du cas de figure de l'adjoint désorienté entre services ... 174
Exemples de politique IAM ... 181
Résolution des problèmes d'identité et d'accès avec ... 187
Journalisation et surveillance ... 189
Outils de surveillance automatique ... 189
Outils de surveillance manuelle ... 190
Surveillance à l'aide de CloudWatch journaux ... 190
Surveillance à l'aide d' CloudWatch événements ... 196
Journalisation des appels d'API CloudTrail avec ... 204
Validation de conformité ... 209
Résilience ... 209
Sécurité de l'infrastructure ... 210
Quotas ... 211
Commandes ... 212
AWS IoT Analytics actions ... 212
AWS IoT Analytics données ... 212
Résolution de problème ... 213
Comment savoir si mes messages arrivent AWS IoT Analytics ? ... 213
Pourquoi mon pipeline perd-il des messages ? Comment puis-je résoudre ce problème ? ... 214
Pourquoi n'y a-t-il aucune donnée dans mon magasin de données ? ... 215
Pourquoi mon jeu de données s'affiche-t-il simplement `__dt` ? ... 215
Comment coder un événement provoqué par la complétion de l'ensemble de données ? ... 216
Comment configurer correctement mon instance de bloc-notes à utiliser AWS IoT Analytics ? ... 216
Pourquoi ne puis-je pas créer de blocs-notes dans une instance ? ... 216
Pourquoi est-ce que je ne vois pas mes ensembles de données dans Quick Suite ? ... 217
Pourquoi le bouton de conteneurisation ne s'affiche-t-il pas sur mon bloc-notes Jupyter existant ? ... 217
Pourquoi l'installation de mon plugin de conteneurisation échoue-t-elle ? ... 218
Pourquoi mon plugin de conteneurisation génère-t-il une erreur ? ... 218
Pourquoi est-ce que je ne vois pas mes variables pendant la conteneurisation ? ... 219
Quelles variables puis-je ajouter à mon conteneur en tant qu'entrée ? ... 219
Comment définir la sortie de mon conteneur comme entrée pour une analyse ultérieure ? ... 219
Pourquoi mon jeu de données de conteneurs échoue-t-il ? ... 219
Historique de la documentation ... 221
Mises à jour antérieures ... 223

..... ccxxiv

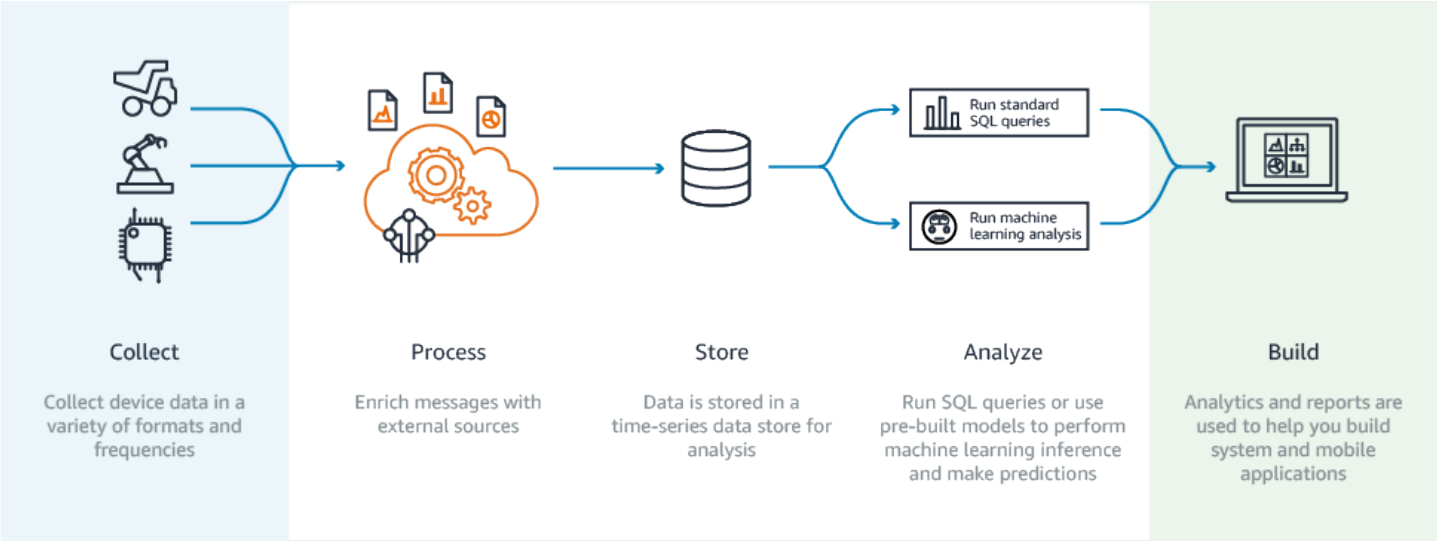
Qu'est-ce que c'est AWS IoT Analytics ?

AWS IoT Analytics automatise les étapes requises pour analyser les données des appareils IoT. AWS IoT Analytics filtre, transforme et enrichit les données IoT avant de les stocker dans un magasin de données chronologiques à des fins d'analyse. Vous pouvez configurer le service pour collecter uniquement les données dont vous avez besoin pour vos appareils, appliquer des transformations mathématiques afin de traiter les données, et enrichir celles-ci avec des métadonnées spécifiques aux appareils, telles que le type et l'emplacement de l'appareil, avant de les stocker. Vous pouvez ensuite analyser vos données en exécutant des requêtes à l'aide du moteur de requêtes SQL intégré, ou effectuer des analyses plus complexes et des inférences basées sur le machine learning. AWS IoT Analytics permet une exploration avancée des données grâce à l'intégration avec Jupyter Notebook. AWS IoT Analytics permet également la visualisation des données grâce à l'intégration à Quick Suite. Quick Suite est disponible dans les régions suivantes.

Les outils d'analyse et d'informatique décisionnelle traditionnels sont conçus pour traiter des données structurées. Les données IoT brutes proviennent souvent d'appareils qui enregistrent des données moins structurées (telles que la température, le mouvement ou le son). Par conséquent, les données issues de ces appareils peuvent présenter des lacunes importantes, des messages endommagés et des relevés erronés qui doivent faire l'objet d'un nettoyage avant de pouvoir être analysés. En outre, les données de l'IoT n'ont souvent de sens que dans le contexte d'autres données provenant de sources externes. AWS IoT Analytics vous permet de résoudre ces problèmes et de collecter de grandes quantités de données sur l'appareil, de traiter les messages et de les stocker. Vous pouvez ensuite interroger les données et les analyser. AWS IoT Analytics inclut des modèles prédéfinis pour les cas d'utilisation courants de l'IoT afin que vous puissiez répondre à des questions telles que les appareils sur le point de tomber en panne ou les clients risquant d'abandonner leurs appareils portables.

Comment utiliser AWS IoT Analytics

Le graphique suivant montre un aperçu de la façon dont vous pouvez l'utiliser AWS IoT Analytics.



Fonctions principales

Collecte

- Intégré à AWS IoT Core—AWS IoT Analytics est entièrement intégré AWS IoT Core afin de pouvoir recevoir des messages provenant d'appareils connectés lors de leur diffusion.
- Utilisez une API par lots pour ajouter des données provenant de n'importe quelle source.AWS IoT Analytics Vous pouvez recevoir des données de n'importe quelle source via HTTP. Cela signifie que tout appareil ou service connecté à Internet peut envoyer des données à AWS IoT Analytics. Pour plus d'informations, consultez BatchPutMessage dans la Référence d'API AWS IoT Analytics .
- Collectez uniquement les données que vous souhaitez stocker et analyser : vous pouvez utiliser la AWS IoT Analytics console pour configurer la réception de messages provenant AWS IoT Analytics d'appareils via des filtres thématiques MQTT dans différents formats et fréquences. AWS IoT Analytics vérifie que les données sont conformes aux paramètres spécifiques que vous définissez et crée des canaux. Le service achemine ensuite les canaux vers des pipelines appropriés en vue du traitement, de la transformation et de l'enrichissement de messages.

Processus

- Nettoyage et filtrage : vous AWS IoT Analytics permet de définir les AWS Lambda fonctions qui sont déclenchées lorsque AWS IoT Analytics des données sont détectées, afin que vous puissiez exécuter du code pour estimer et combler les lacunes. Vous pouvez également définir des filtres maximum et minimum ainsi que des seuils percentiles pour supprimer les valeurs aberrantes dans vos données.

- Transformation :AWS IoT Analytics permet de transformer les messages à l'aide de la logique mathématique ou conditionnelle que vous définissez, afin que vous puissiez effectuer des calculs courants tels que la conversion de degrés Celsius en degrés Fahrenheit.
- Enrichir :AWS IoT Analytics permet d'enrichir les données à l'aide de sources de données externes telles que les prévisions météorologiques, puis de les acheminer vers le magasin de AWS IoT Analytics données.

Stockage

- Stockage de données chronologiques :AWS IoT Analytics stocke les données de l'appareil dans un magasin de données de séries chronologiques optimisé pour une récupération et une analyse plus rapides. Vous pouvez également gérer les autorisations d'accès, implémenter des stratégies de rétention des données et exporter vos données vers des points d'accès externes.
- Stocker les données traitées et brutes :AWS IoT Analytics stocke les données traitées et stocke automatiquement les données brutes ingérées afin que vous puissiez les traiter ultérieurement.

Analyser

- Exécuter des requêtes SQL ad hoc :AWS IoT Analytics fournit un moteur de requêtes SQL qui vous permet d'exécuter des requêtes ad hoc et d'obtenir des résultats rapidement. Le service vous permet d'utiliser des requêtes SQL standard pour extraire des données du magasin de données afin de répondre à des questions telles que la distance moyenne parcourue par un parc de véhicules connectés ou le nombre de portes verrouillées après 19 heures dans un bâtiment intelligent. Ces requêtes peuvent être réutilisées même en cas de modification des appareils connectés, de la taille de groupe et des exigences d'analyse.
- Analyse des séries chronologiques :AWS IoT Analytics prend en charge l'analyse des séries chronologiques afin que vous puissiez analyser les performances des appareils au fil du temps et comprendre comment et où ils sont utilisés, surveiller en permanence les données des appareils pour prévoir les problèmes de maintenance et surveiller les capteurs pour prévoir les conditions environnementales et y réagir.
- Carnets hébergés pour des analyses sophistiquées et l'apprentissage automatique :AWS IoT Analytics inclut la prise en charge des blocs-notes hébergés dans Jupyter Notebook pour l'analyse statistique et l'apprentissage automatique. Le service inclut un ensemble de modèles de blocs-notes contenant des modèles d'apprentissage automatique et des AWS visualisations créés par des auteurs. Vous pouvez utiliser les modèles pour commencer à étudier les cas d'utilisation de l'IoT liés au profilage des défaillances des appareils, à la prévision d'événements tels qu'une faible utilisation susceptible d'indiquer que le client abandonnera le produit, ou à la segmentation des appareils en fonction du niveau d'utilisation du client (par exemple, utilisateurs intensifs, utilisateurs le week-end) ou de l'état de santé de l'appareil. Après avoir

créé un bloc-notes, vous pouvez le conteneuriser et l'exécuter selon un calendrier que vous spécifiez. Pour plus d'informations, consultez Automatisation de votre flux de travail.

- Prédiction : vous pouvez effectuer une classification statistique à l'aide d'une méthode appelée régression logistique. Vous pouvez également utiliser Long-Short-Term la mémoire (LSTM), une puissante technique de réseau neuronal permettant de prédire le résultat ou l'état d'un processus qui varie dans le temps. Les modèles de blocs-notes préconfigurés prennent aussi en charge les algorithmes de clustering k-moyennes pour la segmentation d'appareil, qui regroupe vos appareils en groupe de mêmes appareils. Ces modèles sont en général utilisés pour profiler l'intégrité et l'état d'un appareil comme les unités CVC d'une fabrique de chocolat ou de protection des pales d'une éolienne. Encore une fois, ces modèles de bloc-notes peuvent être contenus et exécutés selon un calendrier.

Construisez et visualisez

- Intégration à Quick Suite :AWS IoT Analytics fournit un connecteur à Quick Suite afin que vous puissiez visualiser vos ensembles de données dans un QuickSight tableau de bord.
- Intégration à la console : vous pouvez également visualiser les résultats ou votre analyse ad hoc dans le bloc-notes Jupyter intégré à la « console ». AWS IoT Analytics

AWS IoT Analytics composants et concepts

Canal

Un canal collecte des données à partir d'une rubrique MQTT et archive les messages bruts non traités avant de publier les données dans un pipeline. Vous pouvez également envoyer des messages à une chaîne directement à l'aide de l'BatchPutMessageAPI. Les messages non traités sont stockés dans un compartiment Amazon Simple Storage Service (Amazon S3) que vous gérez ou gérez. AWS IoT Analytics

Pipeline

Un pipeline consomme les messages d'un canal et vous permet de les traiter avant de les stocker dans un magasin de données. Les étapes de traitement, appelées activités (activités du pipeline), effectuent des transformations sur vos messages, telles que la suppression, le renommage ou l'ajout d'attributs de message, le filtrage des messages en fonction des valeurs d'attribut, l'appel de vos fonctions Lambda sur les messages pour un traitement avancé ou l'exécution de transformations mathématiques pour normaliser les données de l'appareil.

Banque de données

Les pipelines stockent leurs messages traités dans une banque de données. Un magasin de données n'est pas une base de données, mais un référentiel scalable et interrogeable de vos messages. Vous pouvez avoir plusieurs banques de données pour les messages provenant de différents appareils ou emplacements, ou filtrés par attributs de message selon la configuration et les exigences de votre pipeline. Comme pour les messages de canal non traités, les messages traités d'un magasin de données sont stockés dans un compartiment Amazon S3 que vous AWS IoT Analytics gérez ou gérez.

Jeu de données

Vous récupérez les données d'un magasin de données en créant un ensemble de données. AWS IoT Analytics vous permet de créer un ensemble de données SQL ou un ensemble de données de conteneur.

Une fois que vous avez un ensemble de données, vous pouvez explorer et mieux comprendre vos données grâce à l'intégration à l'aide de Quick Suite. Vous pouvez également exécuter des fonctions analytiques plus avancées grâce à l'intégration avec Jupyter Notebook. Jupyter Notebook fournit de puissants outils de science des données capables d'effectuer de l'apprentissage automatique et de nombreuses analyses statistiques. Pour plus d'informations, consultez la section Modèles de bloc-notes.

Vous pouvez envoyer le contenu d'un ensemble de données vers un compartiment Amazon S3, ce qui permet l'intégration à vos lacs de données existants ou l'accès à partir d'applications et d'outils de visualisation internes. Vous pouvez également envoyer le contenu d'un ensemble de données en entrée à AWS IoT Eventsun service qui vous permet de surveiller les appareils ou les processus pour détecter les défaillances ou les changements de fonctionnement, et de déclencher des actions supplémentaires lorsque de tels événements se produisent.

Ensemble de données SQL

Un ensemble de données SQL est similaire à une vue matérialisée d'une base de données SQL. Vous pouvez créer un ensemble de données SQL en appliquant une action SQL. Les ensembles de données SQL peuvent être générés automatiquement selon un calendrier récurrent en spécifiant un déclencheur.

Ensemble de données de conteneur

Un ensemble de données de conteneur vous permet d'exécuter automatiquement vos outils d'analyse et de générer des résultats. Pour plus d'informations, consultez Automatisation de votre flux de travail. Il rassemble un ensemble de données SQL en entrée, un conteneur Docker avec

vos outils d'analyse et les fichiers de bibliothèque requis, les variables d'entrée et de sortie, et un déclencheur de planification facultatif. Les variables d'entrée et de sortie indiquent à l'image exécutable où obtenir les données et stocker les résultats. Le déclencheur peut exécuter votre analyse lorsqu'un ensemble de données SQL finit de créer son contenu ou selon une expression de planification temporelle. Un ensemble de données de conteneur exécute, génère et enregistre automatiquement les résultats des outils d'analyse.

Déclencheur

Vous pouvez créer automatiquement un ensemble de données en spécifiant un déclencheur. Le déclencheur peut être un intervalle de temps (par exemple, créer cet ensemble de données toutes les deux heures) ou lorsque le contenu d'un autre ensemble de données a été créé (par exemple, créer cet ensemble de données lorsque la création de son contenu est `myOtherDataset` terminée). Vous pouvez également générer le contenu d'un ensemble de données manuellement à l'aide de l'CreateDatasetContentAPI.

Conteneur Docker

Vous pouvez créer votre propre conteneur Docker pour emballer vos outils d'analyse ou utiliser les options proposées par l' SageMaker IA. Pour plus d'informations, consultez la section Conteneur Docker. Vous pouvez créer votre propre conteneur Docker pour empaqueter vos outils d'analyse ou utiliser les options fournies par l'SageMaker IA. Vous pouvez stocker un conteneur dans un registre Amazon ECR que vous spécifiez afin de pouvoir l'installer sur la plateforme de votre choix. Les conteneurs Docker sont capables d'exécuter votre code analytique personnalisé préparé avec Matlab, Octave, Wise.io, SPSS, R, Fortran, Python, Scala, Java, C++, etc. Pour plus d'informations, consultez la section Conteneurisation d'un bloc-notes.

Fenêtres delta

Les fenêtres delta sont une série d'intervalles temporels contigus, non superposés, définis par l'utilisateur. Les fenêtres Delta vous permettent de créer le contenu de l'ensemble de données avec les nouvelles données arrivées dans le magasin de données depuis la dernière analyse et d'effectuer une analyse sur celles-ci. Vous créez une fenêtre delta `deltaTime` en définissant la `filters` partie d'un ensemble `queryAction` de données. Pour en savoir plus, consultez l'API `CreateDataset`. En général, vous souhaiterez créer automatiquement le contenu de l'ensemble de données en configurant également un déclencheur d'intervalle de temps (`triggers:schedule:expression`). Cela vous permet de filtrer les messages arrivés pendant une période donnée, afin que les données contenues dans les messages des fenêtres temporelles précédentes ne soient pas comptées deux fois. Pour plus d'informations, voir Exemple 6 : création d'un jeu de données SQL avec une fenêtre Delta (CLI).

Accès AWS IoT Analytics

Dans le cadre de AWS IoT, AWS IoT Analytics fournit les interfaces suivantes pour permettre à vos appareils de générer des données et à vos applications d'interagir avec les données qu'ils génèrent :

AWS Command Line Interface (AWS CLI)

Exécutez AWS IoT Analytics des commandes pour Windows, OS X et Linux. Ces commandes vous permettent de créer et de gérer des objets, des certificats, des règles et des politiques. Consultez le AWS Command Line Interface Guide de l'utilisateur pour démarrer. Pour plus d'informations sur les commandes pour AWS IoT, voir iot dans la AWS Command Line Interface référence.

> **Important**
>
> Utilisez la `aws iotanalytics` commande pour interagir avec AWS IoT Analytics. Utilisez la `aws iot` commande pour interagir avec d'autres parties du système IoT.

AWS IoT API

Créez vos applications IoT en utilisant des requêtes HTTP ou HTTPS. Ces actions d'API vous permettent de créer et de gérer des éléments, des certificats, des règles et des politiques. Pour de plus amples informations, veuillez consulter Actions dans la Référence d'API AWS IoT .

AWS SDKs

Créez vos AWS IoT Analytics applications en utilisant des langages spécifiques. APIs Ils SDKs encapsulent les API HTTP et HTTPS et vous permettent de programmer dans l'un des langages pris en charge. Pour plus d'informations, consultez la section AWS SDKs et outils.

AWS IoT Device SDKs

Créez des applications qui s'exécutent sur les appareils auxquels vous envoyez des messages AWS IoT Analytics. Pour de plus amples informations, veuillez consulter AWS IoT SDKs.

AWS IoT Analytics Console

Vous pouvez créer les composants pour visualiser les résultats dans la AWS IoT Analytics console.

Cas d’utilisation

Maintenance prédictive

AWS IoT Analytics fournit des modèles pour créer des modèles de maintenance prédictive et les appliquer à vos appareils. Par exemple, vous pouvez l'utiliser pour prévoir AWS IoT Analytics à quel moment les systèmes de chauffage et de refroidissement sont susceptibles de tomber en panne sur les véhicules utilitaires connectés afin que les véhicules puissent être réacheminés afin d'éviter d'endommager l'expédition. Ou, un fabricant de voitures peut détecter quels véhicules ont des plaquettes de frein usées et ainsi avertir ses clients de faire réviser leur véhicule.

Réapprovisionnement proactif des fournitures

AWS IoT Analytics vous permet de créer des applications IoT capables de surveiller les stocks en temps réel. Par exemple, une société de restauration peut analyser les données des distributeurs de nourritures et passer commande proactivement de marchandise à chaque fois que le stock est bas.

Notation de l'efficacité des processus

Vous pouvez ainsi créer des applications IoT qui surveillent en permanence l'efficacité des différents processus et prennent des mesures pour améliorer le processus. AWS IoT Analytics Par exemple, une société minière peut améliorer l'efficacité de ses camions de minerais en maximisant la charge par trajet. L'entreprise peut ainsi identifier la charge la plus efficace pour un site ou un camion au fil du temps, puis comparer les éventuels écarts par rapport à la charge cible en temps réel et mieux planifier les principales directives pour améliorer l'efficacité. AWS IoT Analytics

Agriculture intelligente

AWS IoT Analytics peut enrichir les données des appareils IoT avec des métadonnées contextuelles à l'aide de données de AWS IoT registre ou de sources de données publiques afin que votre analyse tienne compte du temps, de l'emplacement, de la température, de l'altitude et d'autres conditions environnementales. Avec cette analyse, vous pouvez écrire des modèles qui produisent les actions recommandées réalisées par vos appareils sur le terrain. Par exemple, pour déterminer à quel moment arroser, les systèmes d'irrigation peuvent enrichir les données des capteurs d'humidité avec des données sur les précipitations, ce qui permet une utilisation plus efficace de l'eau.

AWS IoT Analytics fin du support

Après mûre réflexion, nous avons décidé de mettre fin au AWS IoT Analytics support à compter du 15 décembre 2025. AWS IoT Analytics n'acceptera plus de nouveaux clients à compter du 24 juillet 2024. En tant que client existant disposant d'un compte inscrit au service avant le 23 juillet 2024, vous pouvez continuer à utiliser les AWS IoT Analytics fonctionnalités. Après le 15 décembre 2025, vous ne pourrez plus utiliser AWS IoT Analytics.

À end-of-service l' AWS IoT Analytics approche du 15 décembre 2025, il est important que les clients comprennent leurs options de migration. Cette page fournit un aperçu des principales fonctionnalités des AWS services alternatifs utilisés pour reproduire ces fonctionnalités AWS IoT Analytics et les met en correspondance avec celles-ci. En comprenant les fonctionnalités de ces services alternatifs, les clients peuvent planifier et exécuter une migration fluide, garantissant ainsi la continuité de leurs flux de travail d'analyse de AWS IoT données.

Rubriques

- Options de migration
- Guide de migration

Options de migration

Lorsque vous envisagez une migration depuis AWS IoT Analytics, il est important de comprendre les avantages et les raisons de ce changement. Le tableau ci-dessous propose des options alternatives et un mappage vers les AWS IoT Analytics fonctionnalités existantes.

Action	AWS IoT Analytics	Service alternatif	Raison
Collecte	AWS IoT Analytics permet d'ingérer facilement des données provenant directement de AWS IoT Core ou d'autres sources à l'aide de l'`BatchPutMessage` API. Cette	• Amazon Kinesis Data Streams • Amazon Data Firehose	Amazon Kinesis Data Streams propose une solution robuste. Kinesis diffuse les données en temps réel, ce qui permet un traitement et une analyse immédiats, ce qui est essentiel

Action	AWS IoT Analytics	Service alternatif	Raison
	intégration garantit un flux fluide de données entre vos appareils et la plateforme d'analyse.		pour les applicati ons nécessitant des informations en temps réel et la détection des anomalies. Amazon Data Firehose simplifie le processus de capture et de transformation des données de streaming avant leur arrivée dans Amazon S3, en s'adaptant automatiquement à votre débit de données.

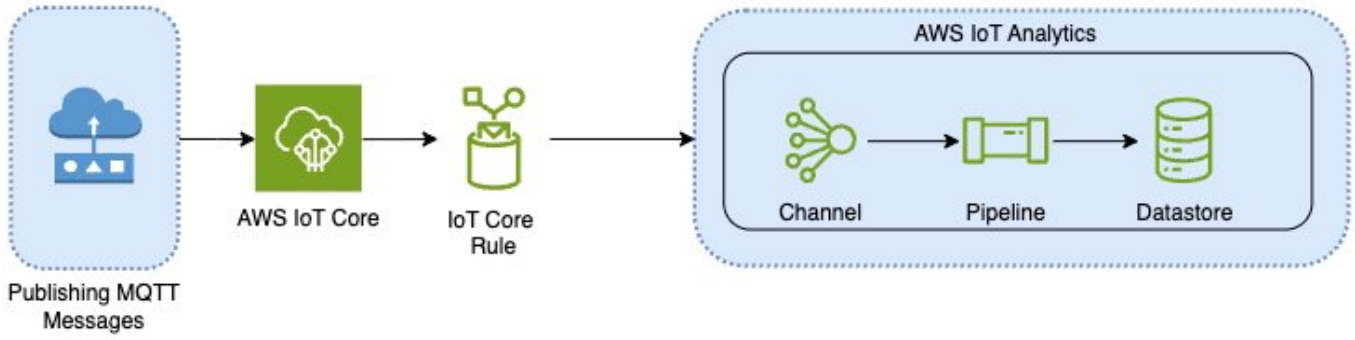
Action	AWS IoT Analytics	Service alternatif	Raison
Processus	Le traitement des données AWS IoT Analytics implique de les nettoyer, de les filtrer, de les transformer et de les enrichir avec des sources externes.	• Service géré Amazon pour Apache Flink • Amazon Data Firehose	Amazon Managed Service pour Apache Flink prend en charge le traitement d'événements complexes, tels que la correspondance de modèles et les agrégations, qui sont essentiels pour les scénarios sophistiqués AWS IoT Analytics . Amazon Data Firehose gère des transformations plus simples et peut invoquer des AWS Lambda fonctions pour un traitement personnalisé, offrant ainsi une flexibilité sans la complexité de Flink.

Action	AWS IoT Analytics	Service alternatif	Raison
Stockage	AWS IoT Analytics utilise un magasin de données chronologiques optimisé pour les AWS IoT données, qui inclut des fonctionnalités telles que les politiques de conservation des données et la gestion des accès.	• Amazon S3 • Amazon Timestream	Amazon S3 propose une solution de stockage évolutive, durable et rentable. L'intégration d'Amazon S3 à d'autres AWS services en fait un excellent choix pour le stockage à long terme et l'analyse d'ensembles de données volumineux. Amazon Timestream est une base de données de séries chronologiques spécialement conçue. Vous pouvez charger des données par lots depuis Amazon S3.

Action	AWS IoT Analytics	Service alternatif	Raison
Analyser	AWS IoT Analytics fournit des fonctionnalités intégrées de requête SQL, d'analyse de séries chronologiques et de prise en charge des blocs-notes Jupyter hébergés, ce qui facilite la réalisation d'analyses avancées et d'apprentissage automatique.	• AWS Glue • Amazon Athena	AWS Glue simplifie le processus ETL en facilitant l'extraction, la transformation et le chargement des données, tout en fournissant un catalogue de données intégré à Athena pour faciliter les requêtes. Amazon Athena va encore plus loin en vous permettant d'exécuter des requêtes SQL directement sur les données stockées dans Amazon S3 sans avoir à gérer d'infrastructure.
Visualisation	AWS IoT Analytics s'intègre à Quick Suite, permettant la création de visualisations et de tableaux de bord riches.	• Connexion rapide	Continuez à utiliser Quick Suite en fonction de la banque de données alternative que vous décidez d'utiliser, comme Amazon S3.

Guide de migration

Dans l'architecture actuelle, AWS IoT les données circulent d'une AWS IoT Core règle AWS IoT Core à AWS IoT Analytics l'autre. AWS IoT Analytics gère l'ingestion, la transformation et le stockage.



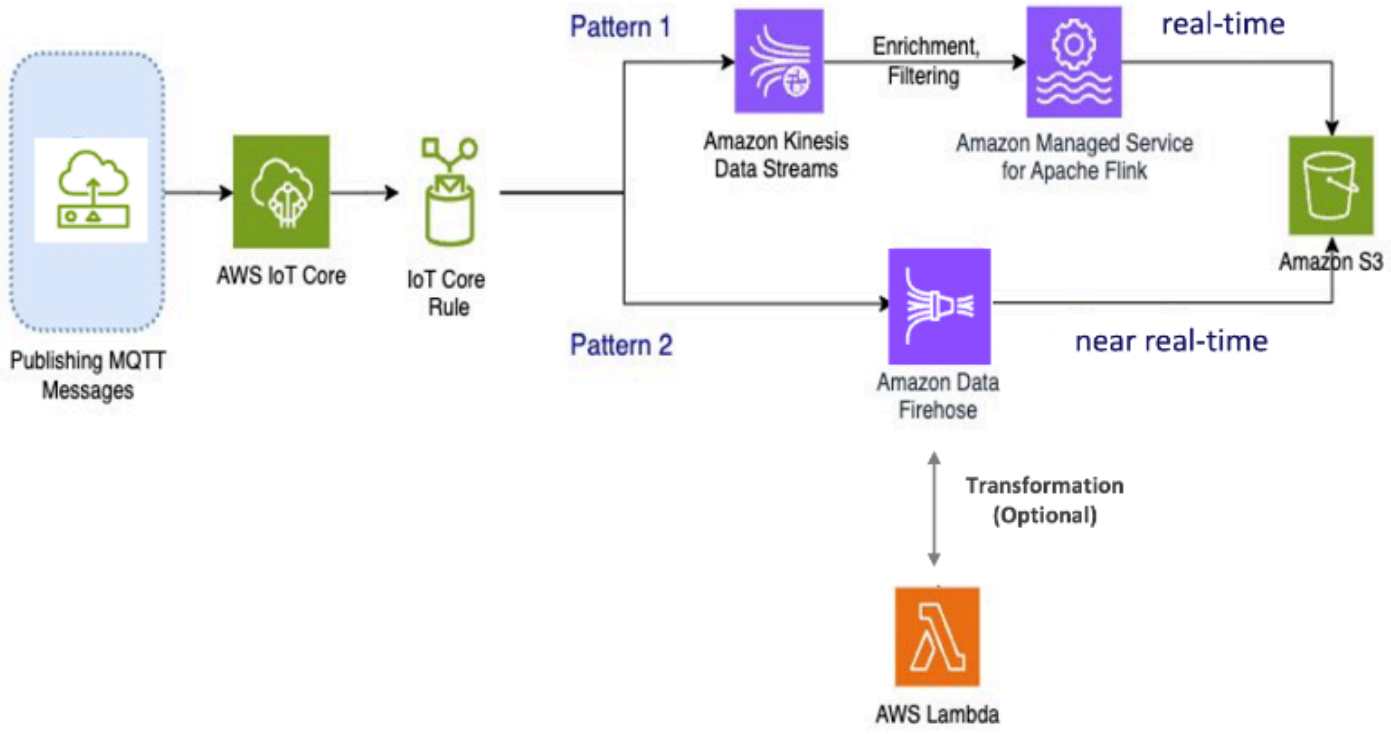
Pour terminer la migration, suivez les deux étapes suivantes :

Rubriques

- Étape 1 : Rediriger l'ingestion de données en cours
- Étape 2 : Exporter les données précédemment ingérées
- Exécutez des requêtes à la demande pour les deux modèles
- Résumé

Étape 1 : Rediriger l'ingestion de données en cours

La première étape de votre migration consiste à rediriger votre ingestion de données en cours vers un nouveau service. Nous recommandons deux modèles en fonction de votre cas d'utilisation spécifique :



Modèle 1 : Amazon Kinesis Data Streams avec Amazon Managed Service pour Apache Flink

Dans ce modèle, vous commencez par publier des données AWS IoT Core qui s'intègrent à Amazon Kinesis Data Streams, ce qui vous permet de collecter, de traiter et d'analyser une large bande passante de données en temps réel.

Métriques et analyses

1. Ingestion de données : AWS IoT les données sont ingérées dans un Amazon Kinesis Data Streams en temps réel. Amazon Kinesis Data Streams peut gérer un débit élevé de données provenant de millions d'appareils, ce qui permet d'effectuer AWS IoT des analyses en temps réel et de détecter les anomalies.
2. Traitement des données : utilisez Amazon Managed Service pour Apache Flink pour traiter, enrichir et filtrer les données issues d'Amazon Kinesis Data Streams. Flink fournit des fonctionnalités robustes pour le traitement d'événements complexes, tels que les agrégations, les jointures et les opérations temporelles.

3. Données du magasin : Flink envoie les données traitées à Amazon S3 pour le stockage et une analyse plus approfondie. Ces données peuvent ensuite être consultées à l'aide d'Amazon Athena ou intégrées à d'autres services d' AWS analyse.

Utilisez ce modèle si votre application implique le streaming de données à bande passante élevée et nécessite un traitement avancé, tel que la correspondance de modèles ou le fenêtrage, ce modèle est le mieux adapté.

Modèle 2 : utiliser Amazon Data Firehose

Dans ce modèle, les données sont publiées sur Amazon Data Firehose AWS IoT Core, qui s'intègre, ce qui vous permet de stocker des données directement dans Amazon S3. Ce modèle prend également en charge les transformations de base à l'aide de AWS Lambda.

Métriques et analyses

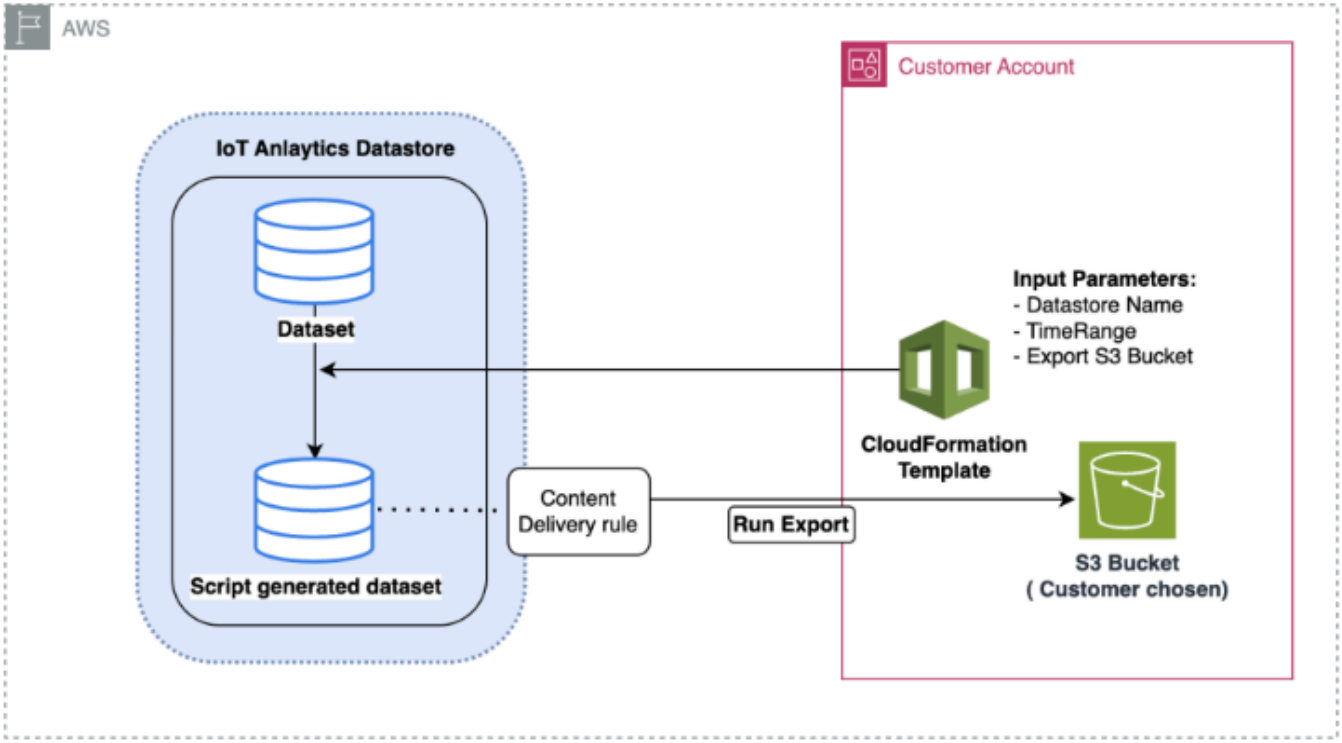
1. Ingestion de données : AWS IoT les données sont ingérées directement depuis vos appareils ou dans AWS IoT Core Amazon Data Firehose.
2. Données de traitement : Amazon Data Firehose effectue des transformations et des traitements de base sur les données, tels que la conversion de format et l'enrichissement. Vous pouvez activer la transformation des données Firehose en la configurant pour appeler des AWS Lambda fonctions permettant de transformer les données sources entrantes avant de les transmettre aux destinations.
3. Données du magasin : les données traitées sont transmises à Amazon S3 en temps quasi réel. Amazon Data Firehose s'adapte automatiquement au débit des données entrantes, garantissant ainsi une diffusion fiable et efficace des données.

Utilisez ce modèle pour les charges de travail qui nécessitent des transformations et un traitement de base. En outre, Amazon Data Firehose simplifie le processus en proposant des fonctionnalités de mise en mémoire tampon et de partitionnement dynamique pour les données stockées dans Amazon S3.

Étape 2 : Exporter les données précédemment ingérées

Pour les données précédemment ingérées et stockées AWS IoT Analytics, vous devez les exporter vers Amazon S3. Pour simplifier ce processus, vous pouvez utiliser un CloudFormation modèle pour

automatiser l'ensemble du flux de travail d'exportation des données. Vous pouvez utiliser le script pour une extraction de données partielle (basée sur une plage de temps).



CloudFormation modèle pour exporter des données vers Amazon S3

Le schéma ci-dessus illustre le processus d'utilisation d'un CloudFormation modèle pour créer un ensemble de données au sein de la même AWS IoT Analytics banque de données, en permettant une sélection basée sur un horodatage. Cela permet aux utilisateurs de récupérer des points de données spécifiques dans le délai souhaité. En outre, une règle de diffusion de contenu est créée pour exporter les données dans un compartiment Amazon S3.

La procédure ci-dessous illustre les étapes à suivre.

1. Préparez le CloudFormation modèle et enregistrez-le en tant que fichier YAML. Par exemple, `migrate-datasource.yaml`.

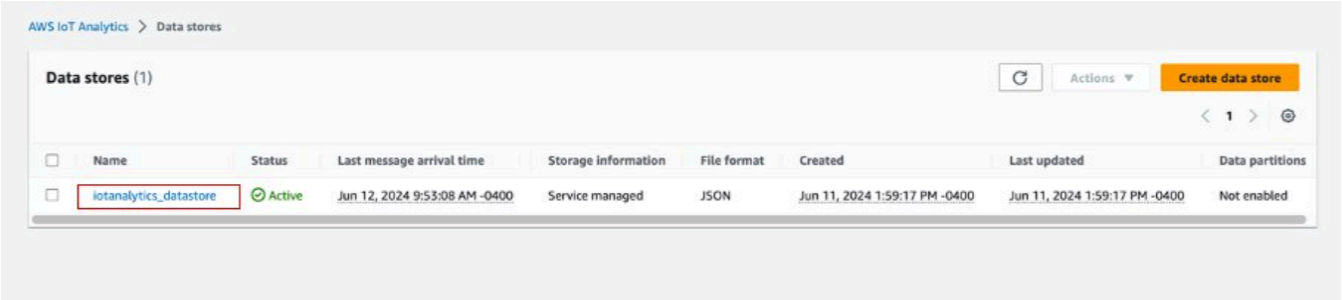
```
# Cloudformation Template to migrate an AWS IoT Analytics datastore to an external
 dataset
AWSTemplateFormatVersion: 2010-09-09
Description: Migrate an AWS IoT Analytics datastore to an external dataset
Parameters:
```

```
  DatastoreName:
    Type: String
    Description: The name of the datastore to migrate.
    AllowedPattern: ^[a-zA-Z0-9_]+$
  TimeRange:
    Type: String
    Description: |
      This is an optional argument to split the source data into multiple files.
      The value should follow the SQL syntax of WHERE clause.
      E.g. WHERE DATE(Item_TimeStamp) BETWEEN '09/16/2010 05:00:00' and '09/21/2010
 09:00:00'.
    Default: ''
  MigrationS3Bucket:
    Type: String
    Description: The S3 Bucket where the datastore will be migrated to.
    AllowedPattern: (?!(^xn--|.+-s3alias$))^[a-z0-9][a-z0-9-]{1,61}[a-z0-9]$
  MigrationS3BucketPrefix:
    Type: String
    Description: The prefix of the S3 Bucket where the datastore will be migrated
 to.
    Default: ''
    AllowedPattern: (^([a-zA-Z0-9.\-_]*\/)*$)|(^$)
Resources:
  # IAM Role to be assumed by the AWS IoT Analytics service to access the external
 dataset
  DatastoreMigrationRole:
    Type: AWS::IAM::Role
    Properties:
      AssumeRolePolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Principal:
              Service: iotanalytics.amazonaws.com
            Action: sts:AssumeRole
      Policies:
        - PolicyName: AllowAccessToExternalDataset
          PolicyDocument:
            Version: 2012-10-17
            Statement:
              - Effect: Allow
                Action:
                  - s3:GetBucketLocation
                  - s3:GetObject
```

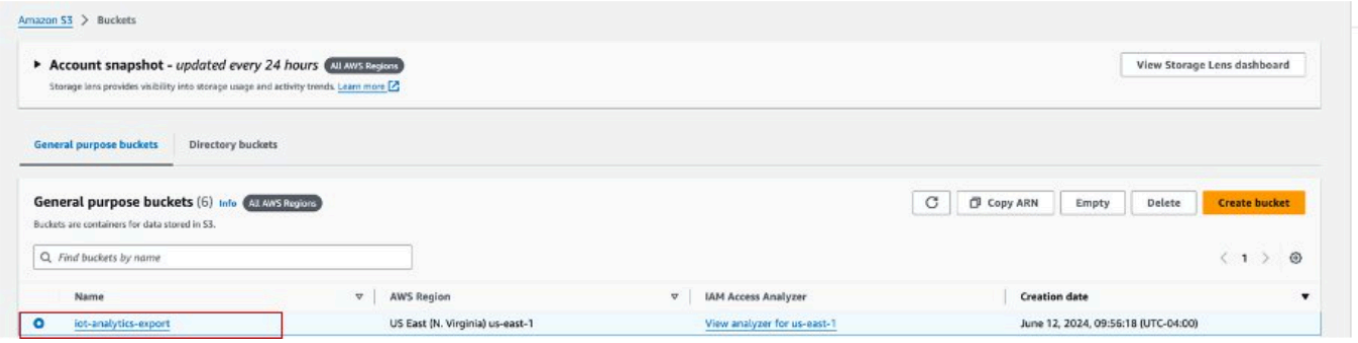
```
                  - s3:ListBucket
                  - s3:ListBucketMultipartUploads
                  - s3:ListMultipartUploadParts
                  - s3:AbortMultipartUpload
                  - s3:PutObject
                  - s3:DeleteObject
                Resource:
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}
                  - !Sub arn:aws:s3:::${MigrationS3Bucket}/
${MigrationS3BucketPrefix}*

  # This dataset that will be created in the external S3 Export
  MigratedDataset:
    Type: AWS::IoTAnalytics::Dataset
    Properties:
      DatasetName: !Sub ${DatastoreName}_generated
      Actions:
        - ActionName: SqlAction
          QueryAction:
            SqlQuery: !Sub SELECT * FROM ${DatastoreName} ${TimeRange}
      ContentDeliveryRules:
        - Destination:
            S3DestinationConfiguration:
              Bucket: !Ref MigrationS3Bucket
              Key: !Sub ${MigrationS3BucketPrefix}${DatastoreName}/!
{iotanalytics:scheduleTime}/!{iotanalytics:versionId}.csv
              RoleArn: !GetAtt DatastoreMigrationRole.Arn
      RetentionPeriod:
        Unlimited: true
      VersioningConfiguration:
        Unlimited: true
```

2. Déterminez la AWS IoT Analytics banque de données dont les données doivent être exportées. Pour ce guide, nous utiliserons un exemple de banque de données nommé`iot_analytics_datastore`.

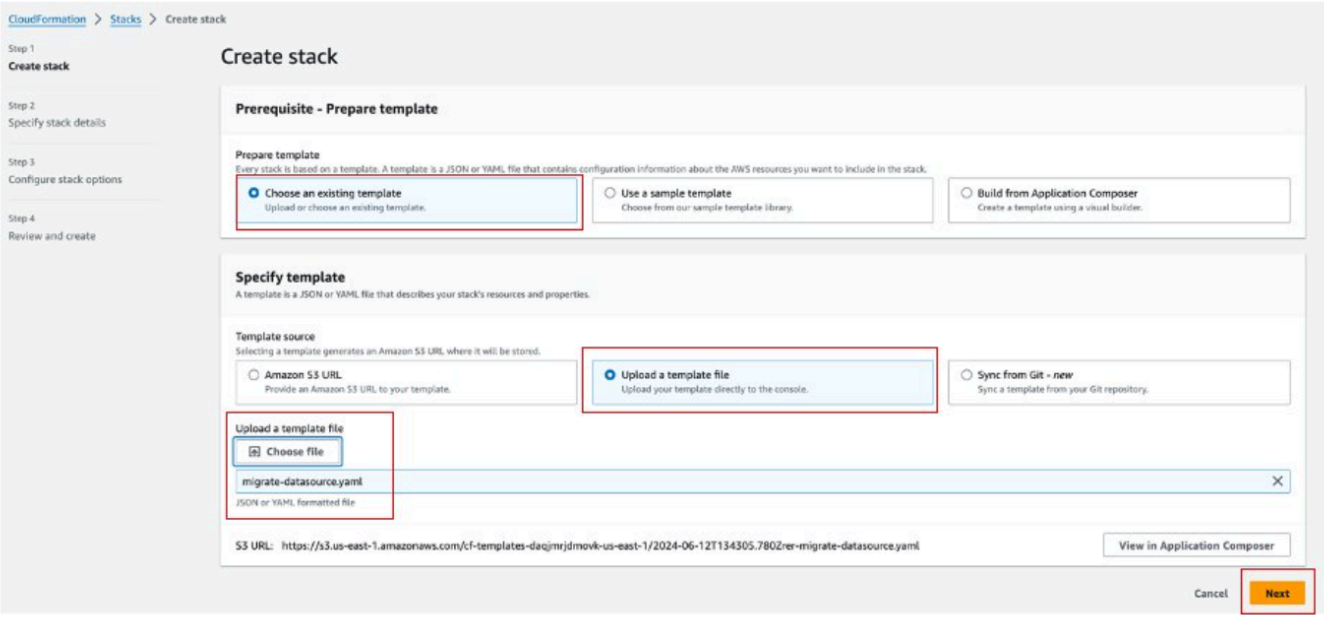


3. Créez ou identifiez un compartiment Amazon S3 dans lequel les données seront exportées. Pour ce guide, nous utiliserons le `iot-analytics-export` godet.



4. Créez la CloudFormation pile.

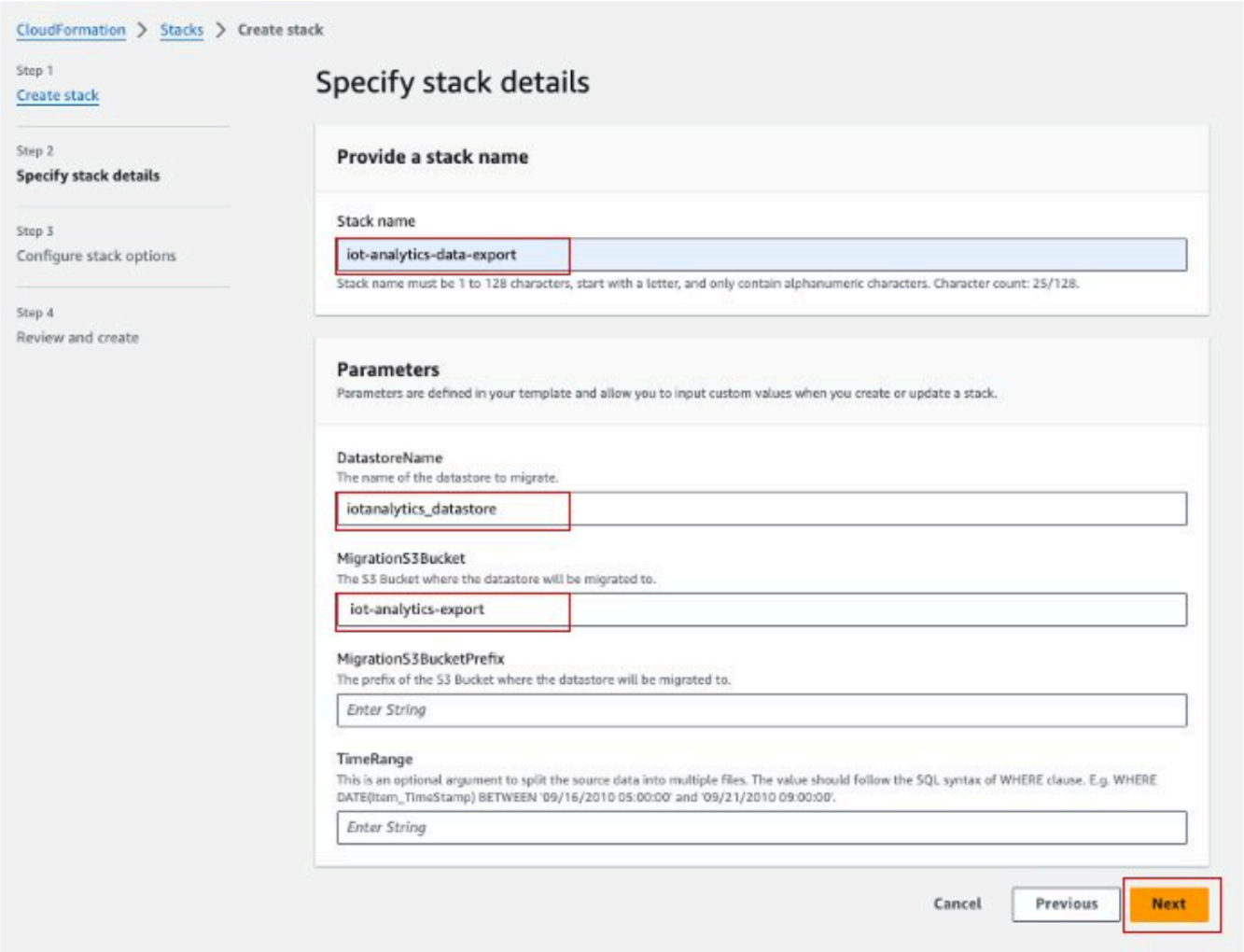
   - Accédez au fichier https://console.aws.amazon.com/cloudformation.
   - Cliquez sur Créer une pile et sélectionnez Avec de nouvelles ressources (standard).
   - Chargez le fichier `migrate-datasource.yaml`.



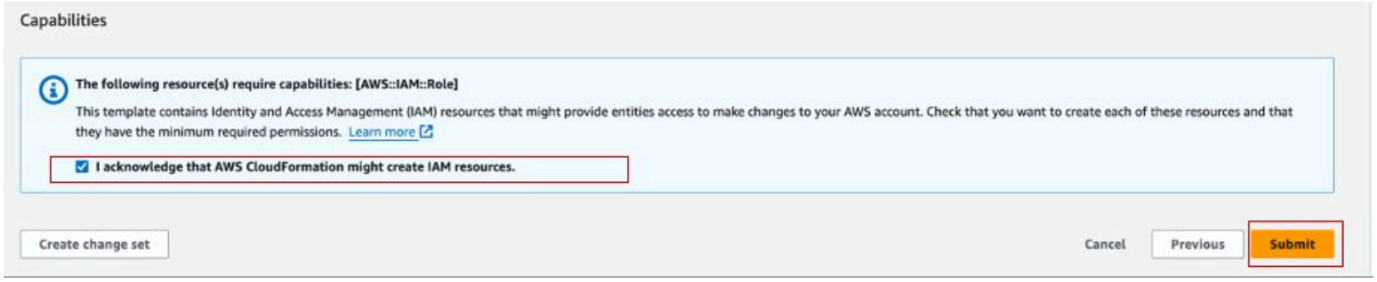
5. Entrez un nom de pile et fournissez les paramètres suivants :

   - DatastoreName: nom de la AWS IoT Analytics banque de données que vous souhaitez migrer.
   - Migrations3bucket : le compartiment Amazon S3 dans lequel les données migrées sont stockées.

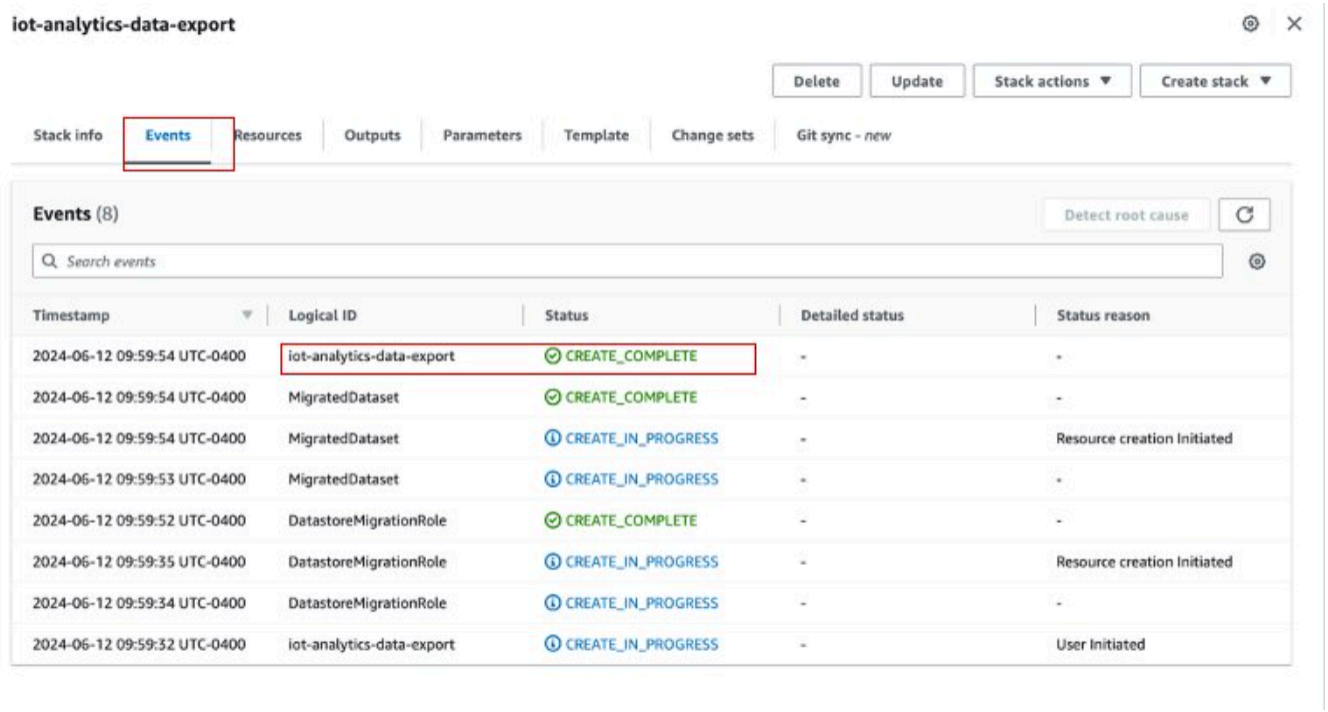
- Migrations3 BucketPrefix (facultatif) : préfixe du compartiment Amazon S3.
- TimeRange(Facultatif) : `SQL WHERE` clause permettant de filtrer les données exportées, permettant de diviser les données sources en plusieurs fichiers en fonction de la plage de temps spécifiée.



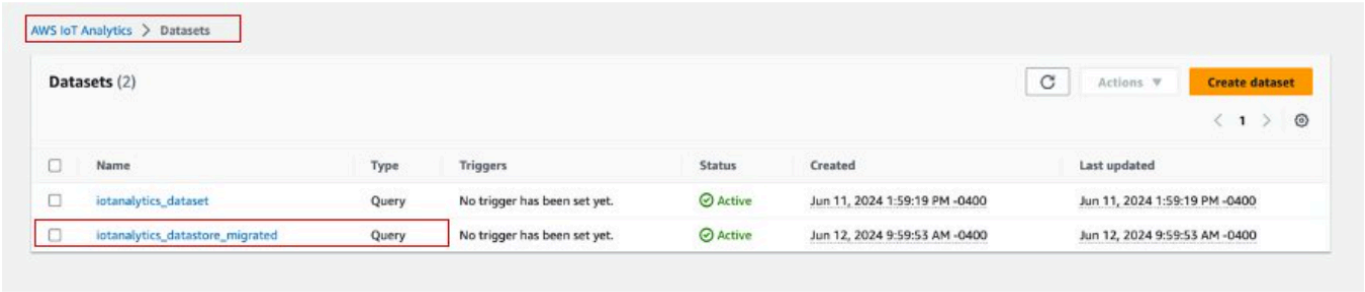
6. Cliquez sur Suivant sur l'écran Configurer les options de pile.
7. Cochez la case pour confirmer la création des ressources IAM, puis cliquez sur Soumettre.



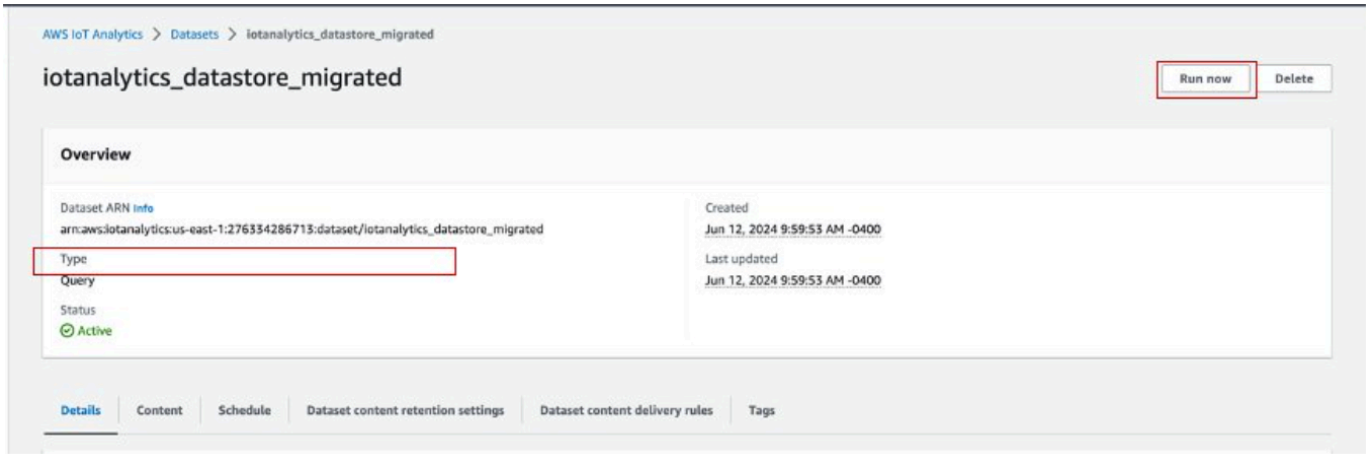
8. Vérifiez que la création de la pile est terminée dans l'onglet Événements.



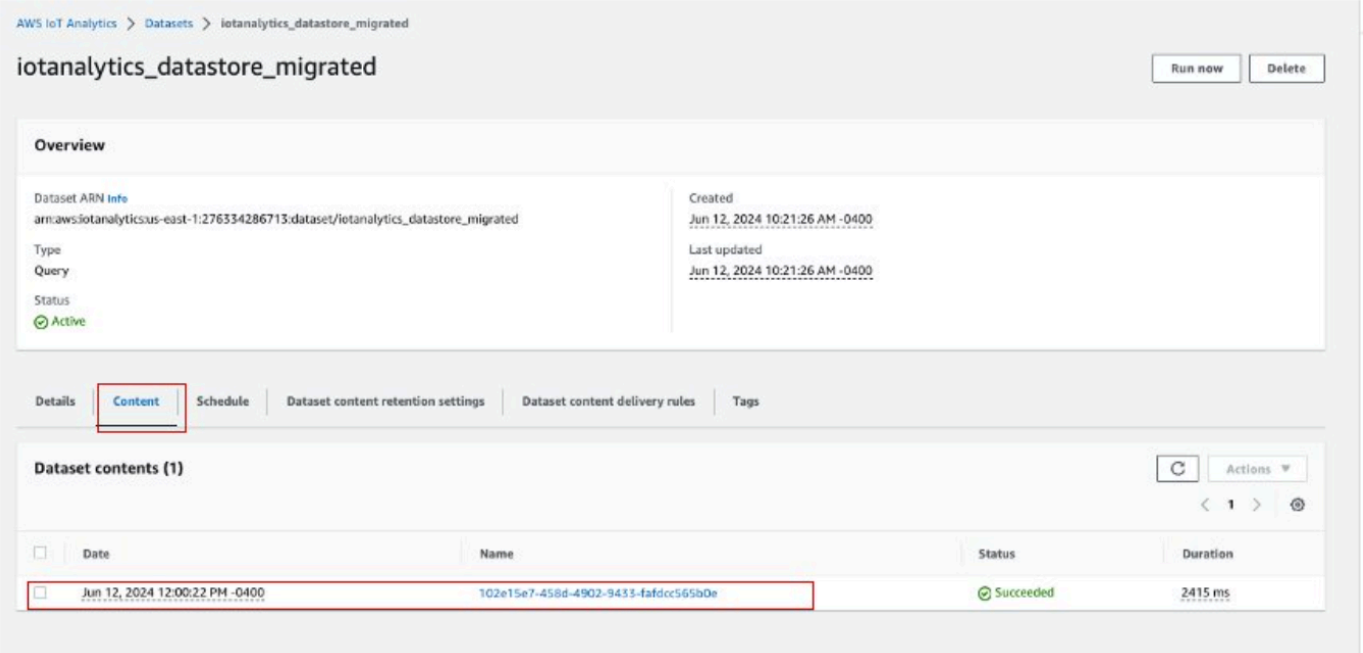
9. Une fois le stack terminé, accédez à AWS IoT Analytics → Ensembles de données pour afficher le jeu de données migré.



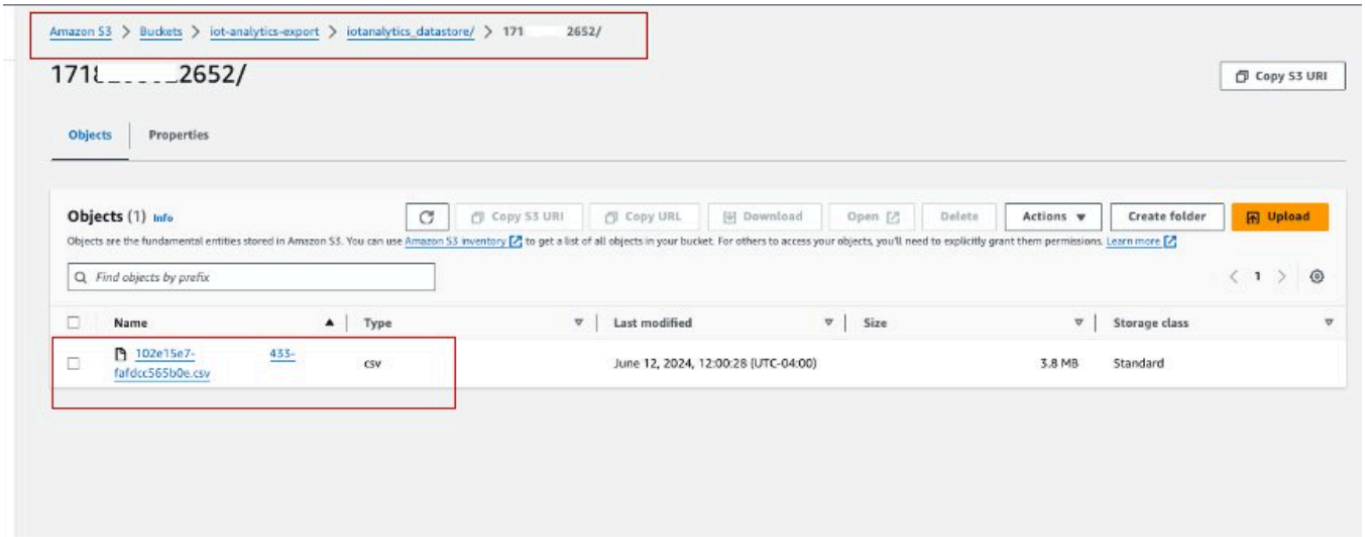
10. Sélectionnez le jeu de données généré, puis cliquez sur Exécuter maintenant pour exporter le jeu de données.



11. Le contenu peut être consulté dans l'onglet Contenu de l'ensemble de données.

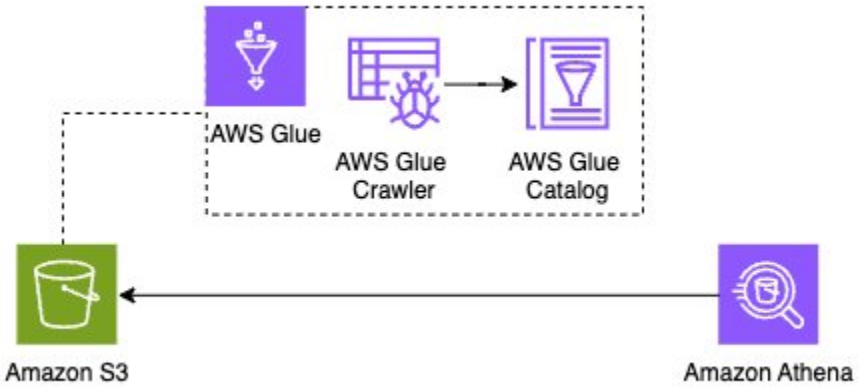


12. Enfin, passez en revue le contenu exporté en ouvrant le iot-analytics-exportcompartiment dans la console Amazon S3.



Exécutez des requêtes à la demande pour les deux modèles

Lorsque vous migrez vos AWS IoT Analytics charges de travail vers Amazon Kinesis Data Streams ou Amazon Data Firehose, vous pouvez tirer parti d' AWS Glue Amazon Athena pour rationaliser davantage votre processus d'analyse des données. AWS Glue simplifie la préparation et la transformation des données, tandis qu'Amazon Athena permet d'interroger rapidement vos données sans serveur. Ensemble, ils fournissent une solution puissante, évolutive et rentable pour l'analyse AWS IoT des données.



Résumé

La migration de votre AWS IoT Analytics charge AWS IoT Analytics de travail depuis Amazon Kinesis Data Streams, Amazon S3, améliore votre capacité à gérer des données complexes à grande

échelle. AWS IoT Cette architecture fournit un stockage évolutif et durable ainsi que de puissantes fonctionnalités d'analyse, vous permettant d'obtenir des informations plus approfondies à partir de vos données IoT en temps réel.

Le nettoyage des ressources créées à l'aide CloudFormation est essentiel pour éviter des coûts imprévus une fois la migration terminée.

Consultez la page de AWS IoT Analytics tarification pour connaître les coûts liés à la migration des données. Envisagez de supprimer le jeu de données nouvellement créé une fois terminé afin d'éviter toute dépense inutile.

Exportation complète du jeu de données : pour exporter le jeu de données complet sans fractionnement temporel, vous pouvez également utiliser la AWS IoT Analytics console et définir une règle de diffusion de contenu en conséquence.

En suivant le guide de migration, vous pouvez effectuer une transition fluide de vos pipelines d'ingestion et de traitement des données, afin de garantir un flux de données continu et fiable. L'utilisation AWS Glue d'Amazon Athena simplifie davantage la préparation des données et l'interrogation, ce qui vous permet d'effectuer des analyses sophistiquées sans gérer aucune infrastructure.

Cette approche vous permet d'intensifier vos AWS IoT Analytics efforts de manière efficace, de vous adapter plus facilement aux demandes croissantes de votre entreprise et de tirer le meilleur parti de vos AWS IoT données.

Mise en route avec AWS IoT Analytics (console)

Utilisez ce didacticiel pour créer les AWS IoT Analytics ressources (également appelées composants) dont vous avez besoin pour découvrir des informations utiles sur les données de vos appareils IoT.

> **Remarques**
>
> - Si vous entrez des majuscules dans le didacticiel suivant, remplacez-les AWS IoT Analytics automatiquement par des minuscules.
> - La AWS IoT Analytics console dispose d'une fonctionnalité de démarrage en un clic pour créer un canal, un pipeline, un magasin de données et un ensemble de données. Vous pouvez accéder à cette fonctionnalité lorsque vous vous connectez à la AWS IoT Analytics console.
> - Ce didacticiel vous explique chaque étape de création de vos AWS IoT Analytics ressources.

Suivez les instructions ci-dessous pour créer un AWS IoT Analytics canal, un pipeline, un magasin de données et un ensemble de données. Le didacticiel explique également comment utiliser la AWS IoT Core console pour envoyer des messages qui seront ingérés. AWS IoT Analytics

Rubriques

- Connectez-vous à la AWS IoT Analytics console
- Création d'une chaîne
- Création d'un magasin de données
- Crée un pipeline.
- Créer un jeu de données
- Envoyer des données de message avec AWS IoT
- Vérifiez la progression des AWS IoT messages
- Accéder aux résultats des requêtes
- Explorez vos données
- Modèles de bloc-notes

Connectez-vous à la AWS IoT Analytics console

Pour commencer, vous devez avoir un AWS compte. Si vous avez déjà un AWS compte, accédez au https://console.aws.amazon.com/iotanalytics/.

Si vous n'avez pas de AWS compte, suivez ces étapes pour en créer un.

Pour créer un AWS compte

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

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

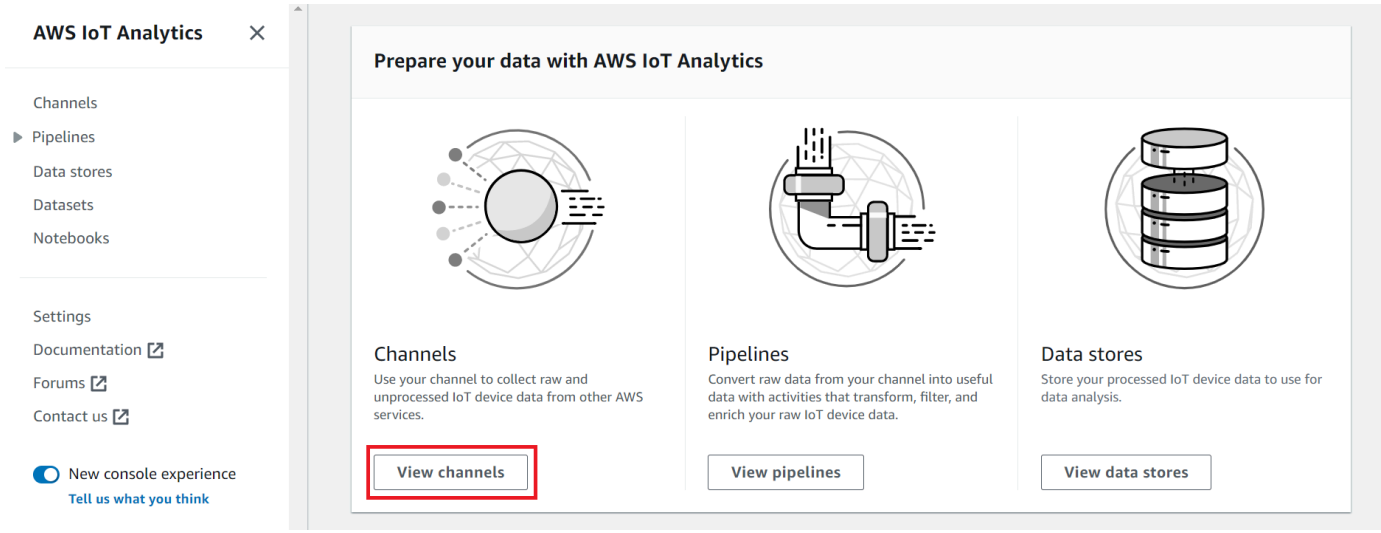
   Lorsque vous vous inscrivez à un Compte AWS, un Utilisateur racine d'un compte AWSest créé. Par défaut, seul l'utilisateur racine a accès à l'ensemble des Services AWS et des ressources de ce compte. La meilleure pratique de sécurité consiste à attribuer un accès administratif à un utilisateur, et à utiliser uniquement l'utilisateur racine pour effectuer les tâches nécessitant un accès utilisateur racine.
3. Connectez-vous au AWS Management Console et naviguez jusqu'au https://console.aws.amazon.com/iotanalytics/.

Création d'une chaîne

Un canal collecte et archive les données brutes, non traitées et non structurées des appareils IoT. Suivez ces étapes pour créer votre chaîne.

Pour créer un canal

1. Dans https://console.aws.amazon.com/iotanalytics/la AWS IoT Analytics section Préparez vos données avec, choisissez Afficher les chaînes.



> **Tip**
>
> Vous pouvez également sélectionner Channels dans le volet de navigation.

2. Sur la page Channels (Canaux), sélectionnez Create channel (Créer un canal).
3. Sur la page Spécifier les détails de la chaîne, entrez les informations relatives à votre chaîne.
   a. Entrez un nom de chaîne unique et facilement identifiable.
   b. (Facultatif) Pour les tags, ajoutez un ou plusieurs tags personnalisés (paires clé-valeur) à votre chaîne. Les balises peuvent vous aider à identifier les ressources pour lesquelles vous créez AWS IoT Analytics.
   c. Choisissez Suivant.
4. AWS IoT Analytics stocke les données brutes non traitées de vos appareils IoT dans un bucket Amazon Simple Storage Service (Amazon S3). Vous pouvez choisir votre propre compartiment Amazon S3, auquel vous pouvez accéder et le gérer, ou vous AWS IoT Analytics pouvez gérer le compartiment Amazon S3 à votre place.
   a. Dans ce didacticiel, pour Type de stockage, choisissez Stockage géré par le service.
   b. Pour Choisir la durée de stockage de vos données brutes, sélectionnez Indéfiniment.
   c. Choisissez Suivant.
5. Sur la page Configurer la source, entrez les informations AWS IoT Analytics à partir desquelles collecter les données des messages AWS IoT Core.

   a. Entrez un filtre de AWS IoT Core sujet, par exemple,`update/environment/dht1`. Plus loin dans ce didacticiel, vous utiliserez ce filtre de rubrique pour envoyer des données de message à votre chaîne.
   b. Dans la zone des rôles IAM, choisissez Create new. Dans la fenêtre Créer un nouveau rôle, entrez le nom du rôle, puis choisissez Créer un rôle. Cela crée automatiquement un rôle auquel est attachée une politique appropriée.
   c. Choisissez Suivant.
6. Passez en revue vos choix, puis choisissez Créer une chaîne.
7. Vérifiez que votre nouvelle chaîne apparaît sur la page Chaînes.

Création d'un magasin de données

Un magasin de données reçoit et stocke les données de vos messages. Un magasin de données n'est pas une base de données. Un magasin de données est plutôt un référentiel évolutif et interrogeable dans un compartiment Amazon S3. Vous pouvez utiliser plusieurs banques de données pour les messages provenant de différents appareils ou emplacements. Vous pouvez également filtrer les données des messages en fonction de la configuration et des exigences de votre pipeline.

Procédez comme suit pour créer un magasin de données.

Pour créer un magasin de données

1. Dans la https://console.aws.amazon.com/iotanalytics/ AWS IoT Analytics section Préparez vos données avec, choisissez Afficher les magasins de données.
2. Sur la page Stockages de données, choisissez Créer un magasin de données.
3. Sur la page Spécifier les détails du magasin de données, entrez les informations de base concernant votre magasin de données.
   a. Pour ID de banque de données, entrez un ID de banque de données unique. Vous ne pouvez pas modifier cet identifiant une fois que vous l'avez créé.
   b. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une ou plusieurs balises personnalisées (paires clé-valeur) à votre banque de données. Les balises peuvent vous aider à identifier les ressources pour lesquelles vous créez AWS IoT Analytics.
   c. Choisissez Suivant.
4. Sur la page Configurer le type de stockage, spécifiez le mode de stockage de vos données.

   a. Pour Type de stockage, choisissez Stockage géré par le service.
   b. Pour Configurer la durée pendant laquelle vous souhaitez conserver vos données traitées, choisissez Indéfiniment.
   c. Choisissez Suivant.
5. AWS IoT Analytics les magasins de données prennent en charge les formats de fichier JSON et Parquet. Pour le format de données de votre magasin de données, choisissez JSON ou Parquet. Consultez Formats de fichier pour plus d'informations sur les types de fichiers AWS IoT Analytics pris en charge.

   Choisissez Suivant.
6. (Facultatif) AWS IoT Analytics prend en charge les partitions personnalisées dans votre banque de données afin que vous puissiez effectuer des requêtes sur des données élaguées afin d'améliorer la latence. Pour plus d'informations sur les partitions personnalisées prises en charge, consultezCloisons personnalisées.

   Choisissez Suivant.
7. Passez en revue vos choix, puis choisissez Créer un magasin de données.
8. Vérifiez que votre nouveau magasin de données apparaît sur la page Stockages de données.

Crée un pipeline.

Vous devez créer un pipeline pour connecter un canal à un magasin de données. Un pipeline de base indique uniquement le canal qui collecte les données et identifie le magasin de données auquel les messages sont envoyés. Pour plus d'informations, consultez la section Activités du pipeline.

Dans le cadre de ce didacticiel, vous allez créer un pipeline qui connecte uniquement un canal à un magasin de données. Vous pourrez ultérieurement ajouter des activités de pipeline pour traiter ces données.

Suivez ces étapes pour créer un pipeline.

Pour créer un pipeline

1. Dans https://console.aws.amazon.com/iotanalytics/la AWS IoT Analytics section Préparez vos données avec, choisissez Afficher les pipelines.

> **Tip**
>
> Vous pouvez également sélectionner Pipelines dans le volet de navigation.

2. Sur la page Pipelines, choisissez Créer un pipeline.
3. Entrez les détails de votre pipeline.
   a. Dans Configurer l'ID et les sources du pipeline, entrez un nom de pipeline.
   b. Choisissez la source de votre pipeline, qui est un AWS IoT Analytics canal à partir duquel votre pipeline lira les messages.
   c. Spécifiez la sortie de votre pipeline, qui est le magasin de données dans lequel les données de vos messages traitées sont stockées.
   d. (Facultatif) Pour les balises, ajoutez une ou plusieurs balises personnalisées (paires clé-valeur) à votre pipeline.
   e. Sur la page Déduire les attributs du message, entrez un nom d'attribut et un exemple de valeur, choisissez un type de données dans la liste, puis choisissez Ajouter un attribut.
   f. Répétez l'étape précédente pour autant d'attributs que nécessaire, puis choisissez Next.
   g. Vous n'ajouterez aucune activité de pipeline pour le moment. Sur la page Enrichir, transformer et filtrer les messages, choisissez Next.
4. Passez en revue vos choix, puis choisissez Créer un pipeline.
5. Vérifiez que votre nouveau pipeline apparaît sur la page Pipelines.

> **Note**
>
> Vous avez créé AWS IoT Analytics des ressources afin qu'elles puissent effectuer les opérations suivantes :
>
> - Collectez des données brutes non traitées sur les messages des appareils IoT via un canal.
> - Stockez les données des messages de votre appareil IoT dans un magasin de données.
> - Nettoyez, filtrez, transformez et enrichissez vos données grâce à un pipeline.

Vous allez ensuite créer un jeu de données AWS IoT Analytics SQL pour découvrir des informations utiles sur votre appareil IoT.

Créer un jeu de données

> **Note**
>
> Un ensemble de données est généralement un ensemble de données qui peuvent ou non être organisées sous forme de tableau. En revanche, AWS IoT Analytics crée votre ensemble de données en appliquant une requête SQL aux données de votre magasin de données.

Vous disposez désormais d'un canal qui achemine les données brutes des messages vers un pipeline qui stocke les données dans un magasin de données où elles peuvent être consultées. Pour interroger les données, vous créez un ensemble de données. Un ensemble de données contient des instructions et des expressions SQL que vous utilisez pour interroger le magasin de données, ainsi qu'un calendrier facultatif qui répète la requête au jour et à l'heure que vous spécifiez. Vous pouvez utiliser des expressions similaires aux expressions CloudWatch de planification Amazon pour créer les plannings facultatifs.

Pour créer un jeu de données

1. Dans le https://console.aws.amazon.com/iotanalytics/volet de navigation de gauche, sélectionnez Datasets.
2. Sur la page Créer un jeu de données, choisissez Create SQL.
3. Sur la page Spécifier les détails du jeu de données, spécifiez les détails de votre ensemble de données.
   a. Entrez un nom pour votre jeu de données.
   b. Pour Source du magasin de données, choisissez l'ID unique qui identifie le magasin de données que vous avez créé précédemment.
   c. (Facultatif) Pour les balises, ajoutez une ou plusieurs balises personnalisées (paires clé-valeur) à votre ensemble de données.

4. Utilisez des expressions SQL pour interroger vos données et répondre à des questions analytiques. Les résultats de votre requête sont stockés dans cet ensemble de données.

   a. Dans le champ de requête Auteur, entrez une requête SQL qui utilise un caractère générique pour afficher jusqu'à cinq lignes de données.

   ```
   SELECT * FROM my_data_store LIMIT 5
   ```

   Pour plus d'informations sur les fonctionnalités SQL prises en charge dans AWS IoT Analytics, consultezExpressions SQL dans AWS IoT Analytics.

   b. Vous pouvez choisir Tester la requête pour vérifier que votre saisie est correcte et afficher les résultats dans un tableau à la suite de la requête.

   > **Note**
   >
   > - À ce stade du didacticiel, votre banque de données est peut-être vide. L'exécution d'une requête SQL sur une banque de données vide ne renverra aucun résultat. Il se peut donc que vous ne `__dt` voyiez que des résultats.
   > - Vous devez veiller à limiter votre requête SQL à une taille raisonnable afin qu'elle ne s'exécute pas pendant une période prolongée, car Athena limite le nombre maximum de requêtes en cours d'exécution. C'est pourquoi vous devez veiller à limiter la taille de la requête SQL à une taille raisonnable.
   >
   > Nous vous suggérons d'utiliser une `LIMIT` clause dans votre requête lors des tests. Une fois le test réussi, vous pouvez supprimer cette clause.

5. (Facultatif) Lorsque vous créez le contenu d'un ensemble de données à partir de données issues d'une période spécifiée, certaines données peuvent ne pas arriver à temps pour être traitées. Pour autoriser un délai, vous pouvez spécifier un décalage, ou delta. Pour de plus amples informations, veuillez consulter Recevoir des notifications de données en retard via Amazon CloudWatch Events.

   Vous n'allez pas configurer de filtre de sélection de données pour le moment. Sur la page Configurer le filtre de sélection des données, choisissez Next.

6. (Facultatif) Vous pouvez planifier l'exécution régulière de cette requête afin d'actualiser le jeu de données. Les plannings des ensembles de données peuvent être créés et modifiés à tout moment.

Vous n'allez pas planifier une exécution récurrente de la requête à ce stade. Sur la page Définir le calendrier des requêtes, choisissez Suivant.

7. AWS IoT Analytics créera des versions du contenu de cet ensemble de données et stockera les résultats de vos analyses pour la période spécifiée. Nous recommandons 90 jours, mais vous pouvez choisir de définir votre politique de conservation personnalisée. Vous pouvez également limiter le nombre de versions stockées du contenu de votre ensemble de données.

   Vous pouvez utiliser la période de conservation du jeu de données par défaut comme étant indéfinie et désactiver le contrôle de version. Sur la page Configurer les résultats de vos analyses, choisissez Next.

8. (Facultatif) Vous pouvez configurer les règles de livraison des résultats de votre jeu de données vers une destination spécifique, telle que AWS IoT Events.

   Vous ne fournirez pas vos résultats ailleurs dans ce didacticiel. Sur la page Configurer les règles de diffusion du contenu du jeu de données, choisissez Suivant.

9. Passez en revue vos choix, puis choisissez Créer un ensemble de données.

10. Vérifiez que votre nouveau jeu de données apparaît sur la page Ensembles de données.

Envoyer des données de message avec AWS IoT

Si vous disposez d'un canal qui achemine les données vers un pipeline, qui les stocke dans un magasin de données où elles peuvent être consultées, vous êtes prêt à y envoyer les données des appareils IoT. AWS IoT Analytics Vous pouvez envoyer des données à AWS IoT Analytics l'aide des options suivantes :

- Utilisez le courtier de AWS IoT messages.
- Utilisez l'opération d'API AWS IoT Analytics BatchPutMessage.

Au cours des étapes suivantes, vous envoyez des données de message depuis le courtier de AWS IoT messages de la AWS IoT Core console afin que AWS IoT Analytics celui-ci puisse les ingérer.

> **Note**
>
> Lorsque vous créez des noms de sujets pour vos messages, tenez compte des points suivants :

- Les noms de sujets ne distinguent pas les majuscules et minuscules. Les champs nommés `example` et appartenant `EXAMPLE` à la même charge utile sont considérés comme des doublons.
- Les noms des sujets ne peuvent pas commencer par le $ personnage. Les sujets commençant par $ sont des sujets réservés et ne peuvent être utilisés que par AWS IoT.
- N'incluez pas d'informations personnellement identifiables dans les noms de vos sujets, car ces informations peuvent apparaître dans des communications et des rapports non cryptés.
- AWS IoT Core Impossible d'envoyer des messages entre les AWS comptes ou AWS les régions.

Pour envoyer des données de message avec AWS IoT

1. Connectez-vous à la console AWS IoT.
2. Dans le volet de navigation, choisissez Test, puis choisissez MQTT test client.
3. Sur la page du client de test MQTT, choisissez Publier dans un sujet.
4. Dans Nom du sujet, entrez un nom qui correspondra au filtre de sujet que vous avez saisi lors de la création d'une chaîne. Cet exemple utilise `update/environment/dht1`.
5. Pour la charge utile du message, entrez le contenu JSON suivant.

```
{
  "thingid": "dht1",
  "temperature": 26,
  "humidity": 29,
  "datetime": "2018-01-26T07:06:01"
}
```

6. (Facultatif) Choisissez Ajouter une configuration pour obtenir des options de protocole de message supplémentaires.
7. Choisissez Publish.

   Cela publie un message qui est capturé par votre chaîne. Votre pipeline achemine ensuite le message vers votre magasin de données.

Vérifiez la progression des AWS IoT messages

Vous pouvez vérifier que les messages sont bien ingérés dans votre chaîne en suivant ces étapes.

Pour vérifier la progression des AWS IoT messages

1. Connectez-vous à la https://console.aws.amazon.com/iotanalytics/.
2. Dans le volet de navigation, choisissez Channels, puis choisissez le nom de canal que vous avez créé précédemment.
3. Sur la page de détails de la chaîne, faites défiler la page vers le bas jusqu'à la section Surveillance, puis ajustez la période affichée (1h 3h 12h 1d 3d 1w). Choisissez une valeur telle que 1w pour afficher les données de la semaine dernière.

Vous pouvez utiliser une fonctionnalité similaire pour surveiller l'activité du pipeline, le temps d'exécution et les erreurs sur la page de détails du pipeline. Dans ce didacticiel, vous n'avez pas spécifié d'activités dans le cadre du pipeline. Vous ne devriez donc pas voir d'erreurs d'exécution.

Pour surveiller l'activité du pipeline

1. Dans le volet de navigation, choisissez Pipelines, puis choisissez le nom du pipeline que vous avez créé précédemment.
2. Sur la page de détails du pipeline, faites défiler la page vers le bas jusqu'à la section Surveillance, puis ajustez la période affichée en choisissant l'un des indicateurs de période (1h 3h 12h 1d 3d 1w).

Accéder aux résultats des requêtes

Le contenu du jeu de données est un fichier contenant le résultat de votre requête, au format CSV.

1. Dans le https://console.aws.amazon.com/iotanalytics/volet de navigation de gauche, sélectionnez Datasets.
2. Sur la page Ensembles de données, choisissez le nom du jeu de données que vous avez créé précédemment.
3. Sur la page d'informations du jeu de données, dans le coin supérieur droit, sélectionnez Exécuter maintenant.

4. Pour vérifier si le jeu de données est prêt, recherchez sous le jeu de données un message similaire à « Vous avez lancé avec succès la requête pour votre ensemble de données ». L'onglet de contenu du jeu de données contient les résultats de la requête et affiche Successed.
5. Pour prévisualiser les résultats de votre requête réussie, dans l'onglet Contenu du jeu de données, sélectionnez le nom de la requête. Pour afficher ou enregistrer le fichier CSV contenant les résultats de la requête, choisissez Télécharger.

> **Note**
>
> AWS IoT Analytics peut intégrer la partie HTML d'un bloc-notes Jupyter sur la page de contenu du jeu de données. Pour de plus amples informations, veuillez consulter Visualisation des AWS IoT Analytics données avec la console.

Explorez vos données

Plusieurs options s'offrent à vous pour stocker, analyser et visualiser vos données.

Amazon Simple Storage Service

Vous pouvez envoyer le contenu d'un ensemble de données vers un compartiment Amazon S3, ce qui permet l'intégration à vos lacs de données existants ou l'accès à partir d'applications internes et d'outils de visualisation. Voir le champ `contentDeliveryRules::destination::s3DestinationConfiguration` dans l'CreateDatasetopération.

AWS IoT Events

Vous pouvez envoyer le contenu du jeu de données en entrée à AWS IoT Events un service qui vous permet de surveiller les appareils ou les processus pour détecter les défaillances ou les changements de fonctionnement, et de lancer des actions supplémentaires lorsque de tels événements se produisent.

Pour ce faire, créez un ensemble de données à l'aide de l'CreateDatasetopération et spécifiez une AWS IoT Events entrée dans le champ`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. Vous devez également spécifier le `roleArn` rôle, qui accorde AWS IoT Analytics les autorisations d'exécution`iotevents:BatchPutMessage`. Chaque fois que le contenu des ensembles de données est créé, AWS IoT Analytics chaque entrée de contenu du jeu de données est envoyée

sous forme de message à l' AWS IoT Events entrée spécifiée. Par exemple, si votre ensemble de données contient le contenu suivant.

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

AWS IoT Analytics Envoie ensuite des messages contenant des champs tels que les suivants.

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

Vous souhaiterez créer une AWS IoT Events entrée qui reconnaît les champs qui vous intéressent (un ou plusieurs des`what`,`who`,`dt`) et créer un modèle de AWS IoT Events détecteur qui utilise ces champs de saisie lors d'événements pour déclencher des actions ou définir des variables internes.

Bloc-notes Jupyter

Jupyter Notebook est une solution open source permettant d'utiliser des langages de script pour exécuter une exploration de données ad hoc et des analyses avancées. Vous pouvez approfondir et appliquer des analyses plus complexes et utiliser des méthodes d'apprentissage automatique, telles que le clustering k-means et les modèles de régression à des fins de prédiction, sur les données de vos appareils IoT.

AWS IoT Analytics utilise les instances de SageMaker blocs-notes Amazon AI pour héberger ses blocs-notes Jupyter. Avant de créer une instance de bloc-notes, vous devez créer une relation entre Amazon SageMaker AI AWS IoT Analytics et Amazon :

1. Accédez à la console SageMaker AI et créez une instance de bloc-notes :
   a. Renseignez les détails, puis choisissez Create a new role (Créer un rôle). Notez l'ARN du rôle.
   b. Créez une instance de blocs-notes.
2. Accédez à la console IAM et modifiez le rôle SageMaker AI :

a. Ouvrez le rôle. Il doit avoir une stratégie gérée.

b. Choisissez Ajouter une politique intégrée, puis pour Service, choisissez IoTAnalytics. Choisissez Sélectionner des actions, puis entrez **`GetDatasetContent`** dans la zone de recherche et choisissez-la. Choisissez Review Policy (Examiner une stratégie).

c. Vérifiez l'exactitude de la politique, entrez un nom, puis choisissez Créer une politique.

Cela donne au rôle nouvellement créé l'autorisation de lire un ensemble de données AWS IoT Analytics.

1. Revenez au https://console.aws.amazon.com/iotanalytics/, et dans le volet de navigation de gauche, choisissez Notebooks. Sur la page Carnets de notes, choisissez Créer un bloc-notes.
2. Sur la page Sélectionner un modèle, choisissez un modèle vierge IoTA.
3. Sur la page Configurer le bloc-notes, entrez le nom de votre bloc-notes. Dans Sélectionner la source du jeu de données, choisissez puis choisissez le jeu de données que vous avez créé précédemment. Dans Sélectionnez une instance de bloc-notes, choisissez l'instance de bloc-notes que vous avez créée dans SageMaker AI.
4. Après avoir passé en revue vos choix, choisissez Créer un bloc-notes.
5. Sur la page Notebooks, votre instance de bloc-notes s'ouvre dans la console Amazon SageMaker AI.

Modèles de bloc-notes

Les modèles de AWS IoT Analytics blocs-notes contiennent AWS des modèles d'apprentissage automatique et des visualisations créés par des créateurs pour vous aider à démarrer avec des cas AWS IoT Analytics d'utilisation. Vous pouvez utiliser ces modèles de blocs-notes pour en savoir plus ou les réutiliser pour les adapter aux données de vos appareils IoT et apporter une valeur ajoutée immédiate.

Vous trouverez les modèles de bloc-notes suivants dans la AWS IoT Analytics console :

- Détection d'anomalies contextuelles — Application de la détection d'anomalies contextuelles à la vitesse du vent mesurée à l'aide d'un modèle de moyenne mobile pondérée exponentiellement (PEWMA) de Poisson.

- Prévision de la production des panneaux solaires — Application de modèles de séries chronologiques par morceaux, saisonniers et linéaires pour prédire la production des panneaux solaires.
- Maintenance prédictive sur les moteurs à réaction — Application de réseaux neuronaux multivariés à mémoire à court terme (LSTM) et de régression logistique pour prévoir les pannes de moteurs à réaction.
- Segmentation de la clientèle des maisons intelligentes — Application de la méthode k-means et de l'analyse des composants principaux (PCA) pour détecter différents segments de clientèle dans les données relatives à l'utilisation de la maison intelligente.
- Prévision de la congestion dans les villes intelligentes — Application du LSTM pour prévoir les taux d'utilisation des autoroutes urbaines.
- Prévision de la qualité de l'air dans les villes intelligentes — Application du LSTM pour prévoir la pollution par les particules dans les centres-villes.

Commencer avec AWS IoT Analytics

Cette section décrit les commandes de base que vous utilisez pour collecter, stocker, traiter et interroger les données de votre appareil à l'aide de AWS IoT Analytics. Les exemples présentés ici utilisent le AWS Command Line Interface (AWS CLI). Pour plus d'informations sur le AWS CLI, consultez le guide de AWS Command Line Interface l'utilisateur. Pour plus d'informations sur les commandes CLI disponibles pour AWS IoT, consultez iot dans la AWS Command Line Interface référence.

> **Important**
>
> Utilisez la `aws iotanalytics` commande pour interagir avec AWS IoT Analytics AWS CLI. Utilisez la `aws iot` commande pour interagir avec d'autres parties du système IoT à l'aide du AWS CLI.

> **Note**
>
> Lorsque vous entrez les noms des AWS IoT Analytics entités (canal, ensemble de données, magasin de données et pipeline) dans les exemples suivants, sachez que toutes les lettres majuscules que vous utilisez sont automatiquement remplacées par des minuscules par le système. Les noms des entités doivent commencer par une lettre minuscule et ne contenir que des lettres minuscules, des traits de soulignement et des chiffres.

Création d'un canal

Un canal collecte et archive les données de messages brutes et non traitées avant de publier ces données dans un pipeline. Les messages entrants étant envoyés à un canal, la première étape consiste à créer un canal pour vos données.

```
aws iotanalytics create-channel --channel-name mychannel
```

Si vous souhaitez que AWS IoT les messages soient ingérés AWS IoT Analytics, vous pouvez créer une règle du moteur de AWS IoT règles pour envoyer les messages à ce canal. Ceci est montré plus loin dansIngestion de données pour AWS IoT Analytics. Une autre méthode pour transférer les données dans un canal consiste à utiliser la AWS IoT Analytics commande`BatchPutMessage`.

Pour répertorier les canaux que vous avez déjà créés :

```
aws iotanalytics list-channels
```

Pour obtenir plus d'informations sur une chaîne.

```
aws iotanalytics describe-channel --channel-name mychannel
```

Les messages de canal non traités sont stockés dans un compartiment Amazon S3 géré par AWS IoT Analytics vous ou dans un compartiment que vous gérez. Utilisez le paramètre `channelStorage` pour spécifier le compartiment. La valeur par défaut est le compartiment Amazon S3 géré par le service. Si vous choisissez de stocker les messages de canal dans un compartiment Amazon S3 que vous gérez, vous devez AWS IoT Analytics autoriser l'exécution des actions suivantes sur votre compartiment Amazon S3 en votre nom : `s3:GetBucketLocation` (vérifier l'emplacement du compartiment) `s3:PutObject` (magasin), `s3:GetObject` (lecture), `s3:ListBucket` (retraitement).

Example

JSON

```
{
    "Version":"2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetObject",
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:PutObject"
            ],
            "Resource": [
                "arn:aws:s3:*:*:*:your-iot-analytics-bucket",
                "arn:aws:s3:*:*:your-iot-analytics-bucket/*"
```

```
            ]
        }
    ]
}
```

Si vous modifiez les options ou les autorisations de votre stockage de canaux géré par le client, vous devrez peut-être retraiter les données des canaux pour vous assurer que les données précédemment ingérées sont incluses dans le contenu du jeu de données. Voir Retraitement des données de canal.

Création d'un magasin de données

Un magasin de données reçoit et stocke vos messages. Il ne s'agit pas d'une base de données, mais d'un dépôt évolutif et interrogeable de vos messages. Vous pouvez créer plusieurs banques de données pour stocker les messages provenant de différents appareils ou emplacements, ou vous pouvez utiliser une seule banque de données pour recevoir tous vos AWS IoT messages.

```
aws iotanalytics create-datastore --datastore-name mydatastore
```

Pour répertorier les magasins de données que vous avez déjà créés.

```
aws iotanalytics list-datastores
```

Pour obtenir plus d'informations sur un magasin de données.

```
aws iotanalytics describe-datastore --datastore-name mydatastore
```

Politiques Amazon S3 relatives aux AWS IoT Analytics ressources

Vous pouvez stocker les messages de banque de données traités dans un compartiment Amazon S3 géré par AWS IoT Analytics ou dans un compartiment que vous gérez. Lorsque vous créez un magasin de données, sélectionnez le compartiment Amazon S3 de votre choix à l'aide du paramètre `datastoreStorage` API. La valeur par défaut est le compartiment Amazon S3 géré par le service.

Si vous choisissez de stocker les messages du magasin de données dans un compartiment Amazon S3 que vous gérez, vous devez AWS IoT Analytics autoriser l'exécution des actions suivantes sur votre compartiment Amazon S3 à votre place :

- `s3:GetBucketLocation`
- `s3:PutObject`
- `s3:DeleteObject`

Si vous utilisez le magasin de données comme source pour un ensemble de données de requêtes SQL, configurez une politique de compartiment Amazon S3 qui AWS IoT Analytics autorise l'appel de requêtes Amazon Athena sur le contenu de votre compartiment.

> **Note**
>
> Nous vous recommandons de le spécifier `aws:SourceArn` dans votre politique de compartiment afin d'éviter le problème de confusion lié à la sécurité des adjoints. Cela restreint l'accès en autorisant uniquement les demandes provenant d'un compte spécifique. Pour de plus amples informations sur le problème de l'adjoint confus, veuillez consulter the section called "Prévention du cas de figure de l'adjoint désorienté entre services".

Voici un exemple de politique de compartiment qui accorde les autorisations requises.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "MyStatementSid",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
```

```
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastore"
                    ]
                }
            }
        }
    ]
}
```

Pour plus d'informations, consultez la section Accès entre comptes dans le guide de l'utilisateur d'Amazon Athena.

> **Note**
>
> Si vous mettez à jour les options ou les autorisations de votre magasin de données géré par le client, vous devrez peut-être retraiter les données des canaux pour vous assurer que toutes les données précédemment ingérées sont incluses dans le contenu du jeu de données. Pour plus d'informations, consultez la section Retraitement des données de canal.

Formats de fichier

AWS IoT Analytics les magasins de données prennent actuellement en charge les formats de fichier JSON et Parquet. Le format de fichier par défaut est JSON.

- JSON (JavaScript Object Notation) : format de texte qui prend en charge les paires nom-valeur et les listes de valeurs ordonnées.
- Apache Parquet : format de stockage en colonnes utilisé pour stocker et interroger efficacement de gros volumes de données.

Pour configurer le format de fichier du magasin de AWS IoT Analytics données, vous pouvez utiliser l'`FileFormatConfiguration`objet lorsque vous créez le magasin de données.

`fileFormatConfiguration`

Contient les informations de configuration des formats de fichiers. AWS IoT Analytics les magasins de données prennent en charge les formats JSON et Parquet.

Le format de fichier par défaut est JSON. Vous ne pouvez spécifier qu'un seul format. Vous ne pouvez pas modifier le format de fichier après avoir créé le magasin de données.

`jsonConfiguration`

Contient les informations de configuration du format JSON.

`parquetConfiguration`

Contient les informations de configuration du format Parquet.

`schemaDefinition`

Informations nécessaires à la définition d'un schéma.

`columns`

Spécifie une ou plusieurs colonnes de stockage de vos données.

Chaque schéma peut contenir jusqu'à 100 colonnes. Chaque colonne peut contenir jusqu'à 100 types imbriqués.

`name`

Le nom de la colonne.

Contraintes de longueur : 1 à 255 caractères.

`type`

Type de données. Pour plus d'informations sur le type de données pris en charge, consultez la section Types de données courants du Guide du AWS Glue développeur.

Contraintes de longueur : 1-131072 caractères.

AWS IoT Analytics prend en charge tous les types de données répertoriés sur la page Types de données sur Amazon Athena, à l'exception `DECIMAL(`*`precision`*`, `*`scale`*`)` de -. *`precision`*

Création d'un magasin de données (console)

La procédure suivante explique comment créer un magasin de données qui enregistre les données au format Parquet.

Pour créer un magasin de données

1. Connectez-vous à la https://console.aws.amazon.com/iotanalytics/.
2. Dans le volet de navigation, sélectionnez Data stores.
3. Sur la page Stockages de données, choisissez Créer un magasin de données.
4. Sur la page Spécifier les détails du magasin de données, entrez les informations de base concernant votre magasin de données.
   a. Pour ID de banque de données, entrez un ID de banque de données unique. Vous ne pouvez pas modifier cet identifiant une fois que vous l'avez créé.
   b. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une ou plusieurs balises personnalisées (paires clé-valeur) à votre banque de données. Les balises peuvent vous aider à identifier les ressources pour lesquelles vous créez AWS IoT Analytics.
   c. Choisissez Suivant.
5. Sur la page Configurer le type de stockage, spécifiez le mode de stockage de vos données.
   a. Pour Type de stockage, choisissez Stockage géré par le service.
   b. Pour Configurer la durée pendant laquelle vous souhaitez conserver vos données traitées, choisissez Indéfiniment.
   c. Choisissez Suivant.
6. Sur la page Configurer le format des données, définissez la structure et le format de vos enregistrements de données.
   a. Pour la classification, choisissez Parquet. Vous ne pouvez pas modifier ce format après avoir créé le magasin de données.
   b. Pour Source d'inférence, choisissez une chaîne JSON pour votre magasin de données.
   c. Pour String, entrez votre schéma au format JSON, comme dans l'exemple suivant.

```
{
    "device_id": "0001",
    "temperature": 26,
    "humidity": 29,
```

```
    "datetime": "2018-01-26T07:06:01"
}
```

d. Choisissez Infer le schéma.

e. Sous Configurer le schéma Parquet, vérifiez que le format correspond à votre exemple JSON. Si le format ne correspond pas, mettez à jour le schéma Parquet manuellement.

- Si vous souhaitez que votre schéma affiche davantage de colonnes, choisissez Ajouter une nouvelle colonne, entrez un nom de colonne, puis choisissez le type de données.

> **Note**
>
> Par défaut, votre schéma peut contenir 100 colonnes. Pour plus d'informations, consultez AWS IoT Analytics Quotas.

- Vous pouvez modifier le type de données d'une colonne existante. Pour plus d'informations sur les types de données pris en charge, consultez la section Types de données courants dans le Guide du AWS Glue développeur.

> **Note**
>
> Après avoir créé votre banque de données, vous ne pouvez pas modifier le type de données d'une colonne existante.

- Pour supprimer une colonne existante, choisissez Supprimer la colonne.

f. Choisissez Suivant.

7. (Facultatif) AWS IoT Analytics prend en charge les partitions personnalisées dans votre banque de données afin que vous puissiez effectuer des requêtes sur des données élaguées afin d'améliorer la latence. Pour plus d'informations sur les partitions personnalisées prises en charge, consultezCloisons personnalisées.

   Choisissez Suivant.

8. Sur la page Réviser et créer, passez en revue vos choix, puis choisissez Créer un magasin de données.

> **Important**
>
> Vous ne pouvez pas modifier l'ID du magasin de données, le format de fichier ou le type de données d'une colonne après avoir créé le magasin de données.

9. Vérifiez que votre nouveau magasin de données apparaît sur la page Stockages de données.

Cloisons personnalisées

AWS IoT Analytics prend en charge le partitionnement des données afin que vous puissiez organiser les données dans votre magasin de données. Lorsque vous utilisez le partitionnement des données pour organiser les données, vous pouvez effectuer des requêtes sur les données élaguées. Cela permet de réduire le volume de données scannées par requête et d'améliorer le temps de latence.

Vous pouvez partitionner vos données en fonction des attributs des données des messages ou des attributs ajoutés par le biais des activités du pipeline.

Pour commencer, activez le partitionnement des données dans un magasin de données. Spécifiez une ou plusieurs dimensions de partition de données et connectez votre magasin de données partitionné à un AWS IoT Analytics pipeline. Rédigez ensuite des requêtes qui exploitent la `WHERE` clause pour optimiser les performances.

Création d'un magasin de données (console)

La procédure suivante explique comment créer un magasin de données avec une partition personnalisée.

Pour créer un magasin de données

1. Connectez-vous à la console AWS IoT Analytics.
2. Dans le volet de navigation, sélectionnez Data stores.
3. Sur la page Stockages de données, choisissez Créer un magasin de données.
4. Sur la page Spécifier les détails du magasin de données, entrez les informations de base concernant votre magasin de données.

   a. Pour ID de banque de données, entrez un ID de banque de données unique. Vous ne pouvez pas modifier cet identifiant une fois que vous l'avez créé.

b. (Facultatif) Pour les balises, choisissez Ajouter une nouvelle balise pour ajouter une ou plusieurs balises personnalisées (paires clé-valeur) à votre banque de données. Les balises peuvent vous aider à identifier les ressources pour lesquelles vous créez AWS IoT Analytics.

c. Choisissez Suivant.

5. Sur la page Configurer le type de stockage, spécifiez le mode de stockage de vos données.

   a. Pour Type de stockage, choisissez Stockage géré par le service.

   b. Pour Configurer la durée pendant laquelle vous souhaitez conserver vos données traitées, choisissez Indéfiniment.

   c. Choisissez Suivant.

6. Sur la page Configurer le format des données, définissez la structure et le format de vos enregistrements de données.

   a. Pour la classification du format de données de votre magasin de données, choisissez JSON ou Parquet. Pour plus d'informations sur les types de fichiers AWS IoT Analytics pris en charge, consultezFormats de fichier.

   > **Note**
   >
   > Vous ne pouvez pas modifier ce format après avoir créé le magasin de données.

   b. Choisissez Suivant.

7. Créez des partitions personnalisées pour ce magasin de données.

   a. Pour Ajouter des partitions de données, sélectionnez Activer.

   b. Pour Source de partition de données, spécifiez les informations de base concernant la source de votre partition.

      Choisissez Exemple de source, puis sélectionnez le AWS IoT Analytics canal qui collecte les messages pour cette banque de données.

   c. Pour les exemples d'attributs de message, sélectionnez les attributs de message que vous souhaitez utiliser pour partitionner votre banque de données. Ajoutez ensuite vos sélections sous forme de dimensions de partition d'attributs ou de dimensions de partition d'horodatage sous Actions.

> **Note**
>
> Vous ne pouvez ajouter qu'une seule partition d'horodatage à votre banque de données.

d. Pour les dimensions de partition de stockage de données personnalisées, définissez les informations de base concernant les dimensions de votre partition. Chaque attribut d'échantillon de message que vous avez sélectionné à l'étape précédente deviendra les dimensions de votre partition. Personnalisez chaque dimension à l'aide des options suivantes :

- Type de partition : spécifiez si cette dimension de partition est un attribut ou un type de partition Timestamp.
- Nom de l'attribut et nom de la dimension : par défaut, le nom de l'attribut d'exemple de message que vous avez sélectionné AWS IoT Analytics sera utilisé comme identifiant pour la dimension de votre partition d'attributs. Modifiez le nom de l'attribut pour personnaliser le nom de la dimension de votre partition. Vous pouvez utiliser le nom de la dimension dans la `WHERE` clause pour optimiser les performances des requêtes.
  - Le nom de toute dimension d'attribut de partition est préfixé par`__partition_`.
  - Pour les types de partitions d'horodatage, AWS IoT Analytics crée les quatre dimensions suivantes avec les noms`__year`,,`__month`,`__day`. `__hour`
- Commande : réorganisez les dimensions de votre partition pour améliorer la latence de vos requêtes.

Pour le format d'horodatage, spécifiez le format de votre partition d'horodatage en faisant correspondre l'horodatage ingéré à partir des données de votre message. Vous pouvez choisir l'une des options de format AWS IoT Analytics répertoriées ou en spécifier une qui correspond au format de vos données. En savoir plus sur la spécification des formateurs de date et d'heure.

Pour ajouter une nouvelle dimension qui n'est pas un attribut de message, choisissez Ajouter de nouvelles partitions.

e. Choisissez Suivant.

8. Sur la page Réviser et créer, passez en revue vos choix, puis choisissez Créer un magasin de données.

> **Important**
>
> - Vous ne pouvez pas modifier l'ID du magasin de données après avoir créé le magasin de données.
> - Pour modifier des partitions existantes, vous devez créer un autre magasin de données et retraiter les données via un pipeline.

9. Vérifiez que votre nouveau magasin de données apparaît sur la page Stockages de données.

Création d'un pipeline

Un pipeline consomme les messages d'un canal et vous permet de traiter et de filtrer les messages avant de les stocker dans un magasin de données. Pour connecter un canal à un magasin de données, vous devez créer un pipeline. Le plus simple des pipelines ne fait rien d'autre que spécifier le canal qui collecte les données et identifier le magasin de données auquel les messages seront envoyés. Pour plus d'informations sur les pipelines plus complexes, consultez la section Activités des pipelines.

Au début, nous vous recommandons de créer un pipeline qui ne fait rien d'autre que connecter un canal à un magasin de données. Ensuite, une fois que vous avez vérifié que les flux de données brutes transitaient vers le magasin de données, vous pouvez introduire des activités de pipeline supplémentaires pour traiter ces données.

Exécutez la commande suivante pour créer un pipeline.

```
aws iotanalytics create-pipeline --cli-input-json file://mypipeline.json
```

Le fichier `mypipeline.json` contient le contenu suivant.

```
{
    "pipelineName": "mypipeline",
    "pipelineActivities": [
        {
            "channel": {
                "name": "mychannelactivity",
                "channelName": "mychannel",
                "next": "mystoreactivity"
```

```
            }
        },
        {
            "datastore": {
                "name": "mystoreactivity",
                "datastoreName": "mydatastore"
            }
        }
    ]
}
```

Exécutez la commande suivante pour répertorier vos pipelines existants.

```
aws iotanalytics list-pipelines
```

Exécutez la commande suivante pour afficher la configuration d'un pipeline individuel.

```
aws iotanalytics describe-pipeline  --pipeline-name mypipeline
```

Ingestion de données pour AWS IoT Analytics

Si vous disposez d'un canal qui achemine les données vers un pipeline qui les stocke dans un magasin de données où elles peuvent être consultées, vous êtes prêt à y envoyer des données de message. AWS IoT Analytics Nous montrons ici deux méthodes pour introduire des données AWS IoT Analytics. Vous pouvez envoyer un message à l'aide du courtier de AWS IoT messages ou de l' AWS IoT Analytics `BatchPutMessage`API.

Rubriques

- Utilisation du courtier de AWS IoT messages
- Utilisation de l' BatchPutMessage API

Utilisation du courtier de AWS IoT messages

Pour utiliser le courtier de AWS IoT messages, vous devez créer une règle à l'aide du moteur de AWS IoT règles. La règle achemine les messages portant sur un sujet spécifique vers AWS IoT Analytics. Mais avant, cette règle nécessite la création d'un rôle qui accorde les autorisations requises.

Création d'un rôle IAM

Pour que AWS IoT les messages soient acheminés vers un AWS IoT Analytics canal, vous devez définir une règle. Mais vous devez d'abord créer un rôle IAM qui accorde à cette règle l'autorisation d'envoyer des données de message à un AWS IoT Analytics canal.

Exécutez la commande suivante pour créer le rôle.

```
aws iam create-role --role-name myAnalyticsRole --assume-role-policy-document file://
arpd.json
```

Le contenu du `arpd.json` fichier doit ressembler à ce qui suit.

JSON

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

Joignez ensuite un document de politique au rôle.

```
aws iam put-role-policy --role-name myAnalyticsRole --policy-name myAnalyticsPolicy --
policy-document file://pd.json
```

Le contenu du `pd.json` fichier doit ressembler à ce qui suit.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
                "arn:aws:iotanalytics:us-east-1:111122223333:channel/your-
channel"
            ]
        }
    ]
}
```

Création d'une AWS IoT règle

Créez une AWS IoT règle qui envoie des messages à votre chaîne.

```
aws iot create-topic-rule --rule-name analyticsTestRule --topic-rule-payload file://
rule.json
```

Le contenu du `rule.json` fichier doit ressembler à ce qui suit.

```
{
    "sql": "SELECT * FROM 'iot/test'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [ {
        "iotAnalytics": {
            "channelName": "mychannel",
            "roleArn": "arn:aws:iam::your-account-number:role/myAnalyticsRole"
        }
    } ]
}
```

Remplacez `iot/test` par la rubrique MQTT des messages qui doivent être acheminés. Remplacez le nom du canal et le rôle par ceux que vous avez créés dans les sections précédentes.

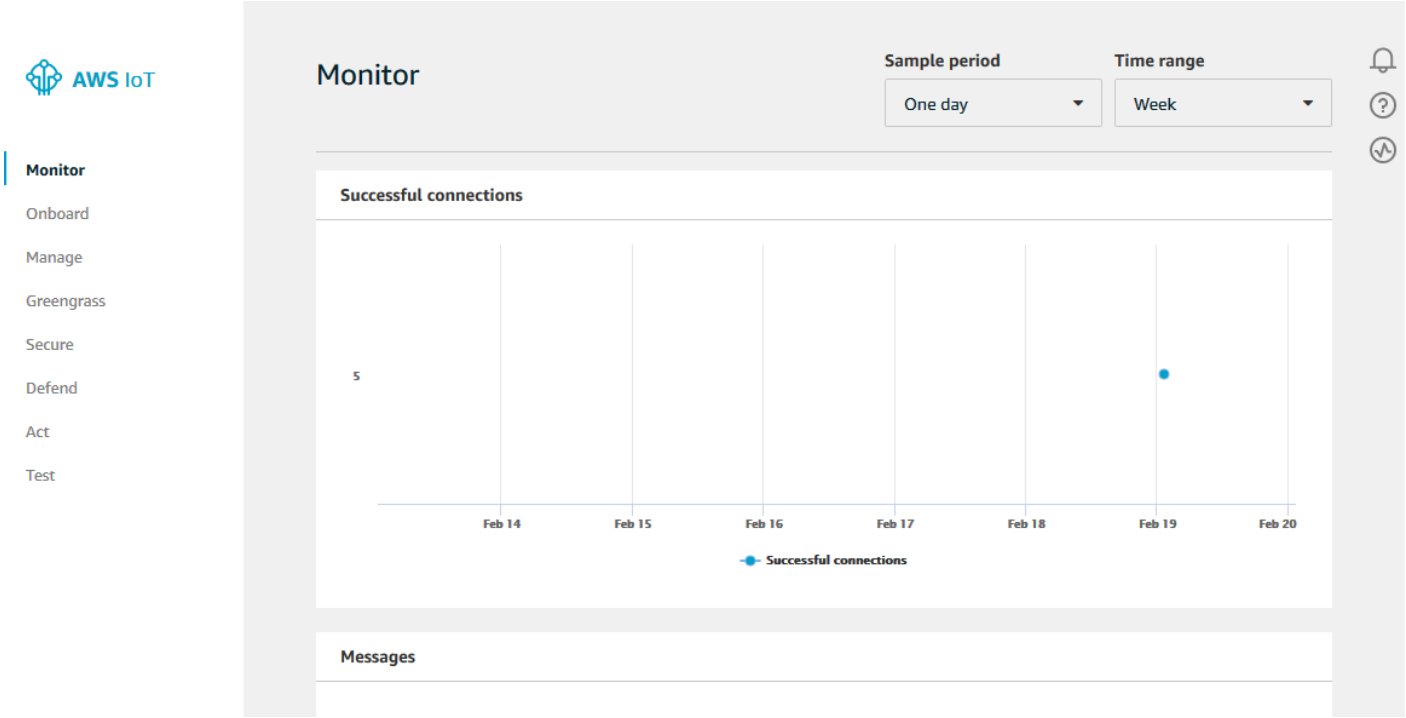
Envoi de messages MQTT à AWS IoT Analytics

Une fois que vous avez joint une règle à un canal, un canal à un pipeline et un pipeline à un magasin de données, toutes les données correspondant à la règle sont désormais transmises AWS IoT

Analytics au magasin de données, prêtes à être consultées. Pour tester cela, vous pouvez utiliser la AWS IoT console pour envoyer un message.

> **Note**
>
> Les noms de champs des charges utiles (données) des messages auxquels vous envoyez. AWS IoT Analytics
>
> - Peuvent uniquement contenir des caractères alphanumériques et des traits de soulignement (_). Les autres caractères spéciaux ne sont pas autorisés.
> - Ils doivent commencer par un caractère alphabétique ou un trait de soulignement (_).
> - Ils ne peuvent pas contenir de tirets (-).
> - En termes d'expressions régulières : « `^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$` ».
> - Ne peut pas comporter plus de 255 caractères
> - Ils ne sont pas sensibles à la casse. Les champs nommés `foo` et appartenant `F00` à la même charge utile sont considérés comme des doublons.
>
> Par exemple, `{"temp_01": 29}` ou `{"_temp_01": 29}` sont valides, mais `{"temp-01": 29}`, `{"01_temp": 29}` ou `{"__temp_01": 29}` ne sont pas valides dans les charges utiles de messages.

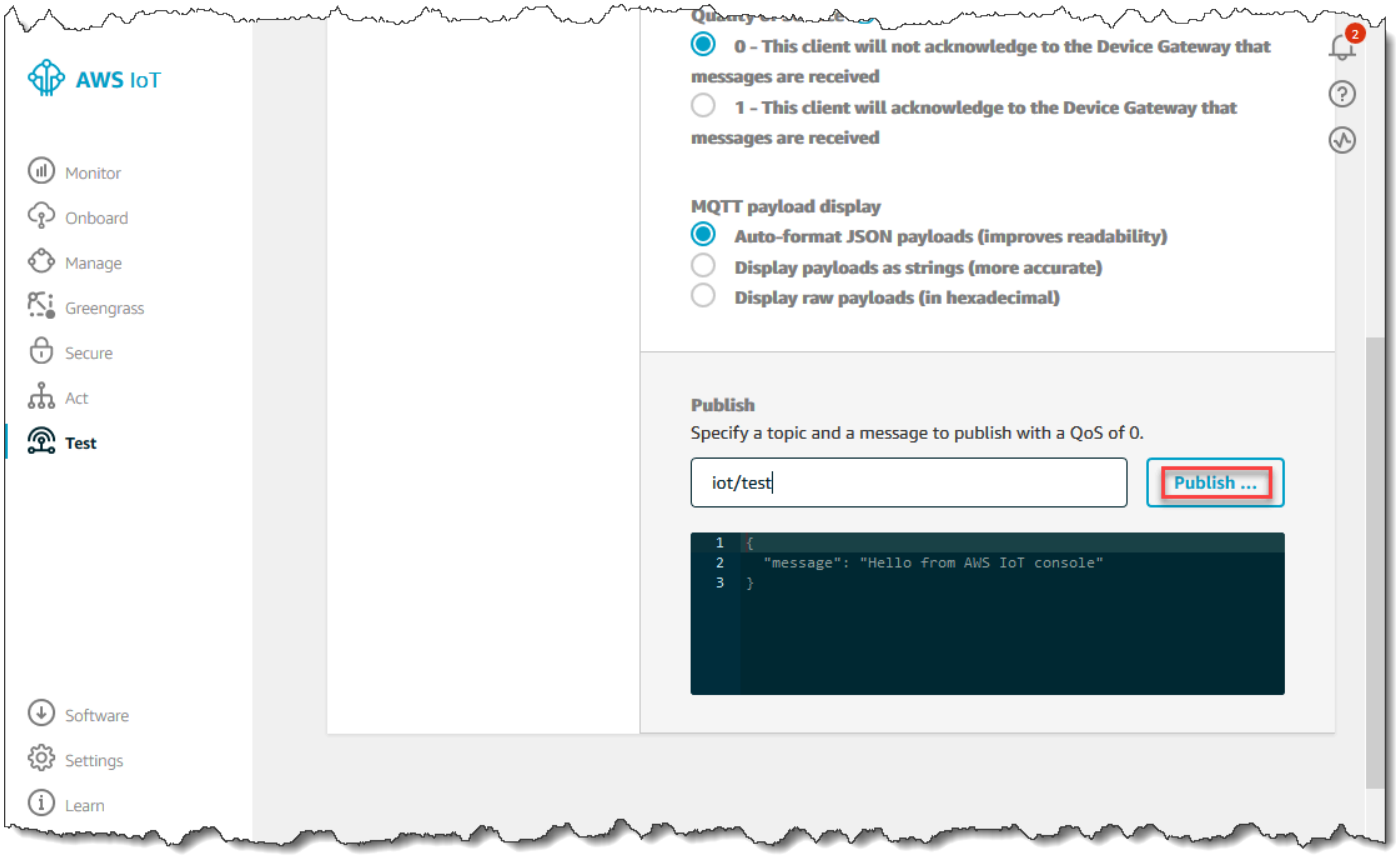
1. Dans la console AWS IoT, dans le volet gauche de navigation, choisissez Test.



2. Sur la page du client MQTT, dans la section Publish, sous Specify a topic, entrez **`iot/test`**. Dans la section charge utile du message, vérifiez que le contenu JSON suivant est présent, ou saisissez-le dans le cas contraire.

```
{
    "message": "Hello from the IoT console"
}
```

3. Choisissez Publish to topic (Publier dans la rubrique.



Cela publie un message qui est acheminé vers le magasin de données que vous avez créé précédemment.

Utilisation de l' BatchPutMessage API

Une autre méthode pour faire entrer les données des messages AWS IoT Analytics consiste à utiliser la commande `BatchPutMessage` API. Cette méthode ne nécessite pas que vous définissiez une AWS IoT règle pour acheminer les messages portant sur un sujet spécifique vers votre chaîne. Mais cela nécessite que l'appareil qui envoie le son data/messages au canal soit capable d'exécuter un logiciel créé avec le AWS SDK ou d'utiliser le AWS CLI to call`BatchPutMessage`.

1. Créez un fichier `messages.json` contenant les messages à envoyer (dans cet exemple, un seul message est envoyé).

```
[
    { "messageId": "message01", "payload": "{ \"message\": \"Hello from the CLI
\" }" }
```

```
]
```

2. Exécutez la commande `batch-put-message`.

```
aws iotanalytics batch-put-message  --channel-name mychannel  --messages file://
messages.json --cli-binary-format raw-in-base64-out
```

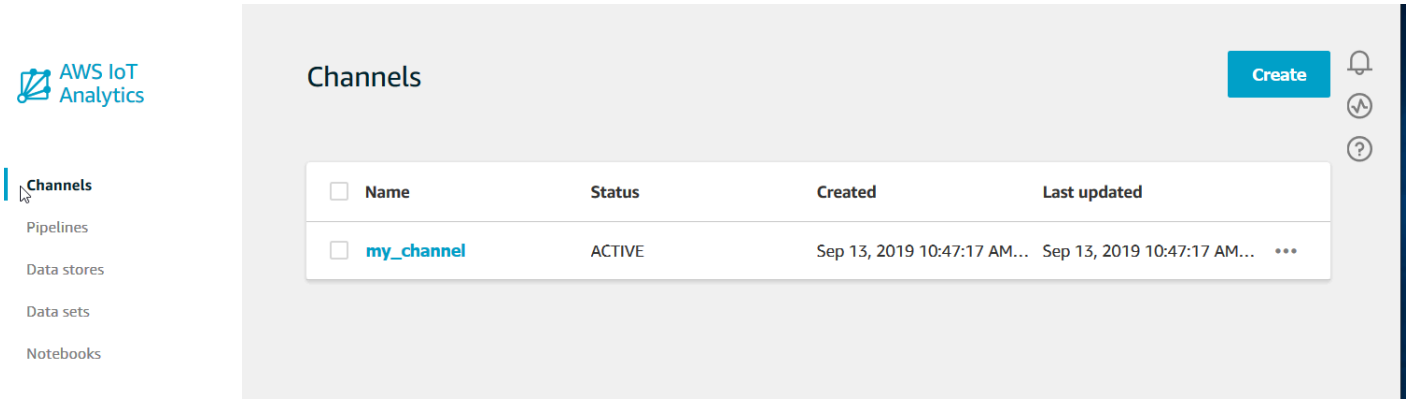
S'il n'y a aucune erreur, le résultat suivant s'affiche.

```
{
    "batchPutMessageErrorEntries": []
}
```

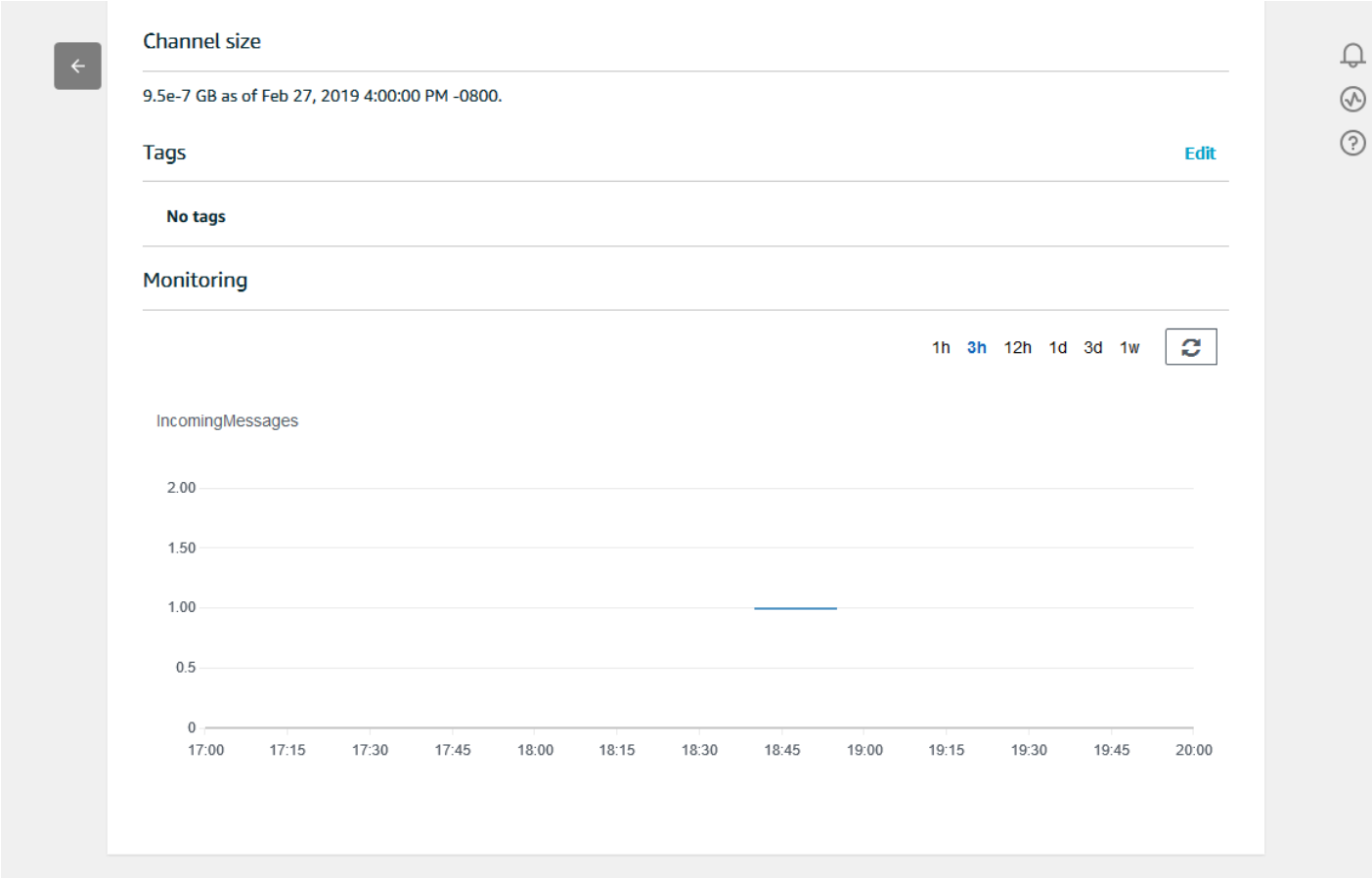
Surveillance des données ingérées

Vous pouvez vérifier que les messages que vous avez envoyés sont bien ingérés dans votre chaîne à l'aide de la AWS IoT Analytics console.

1. Dans le volet de navigation de gauche de la AWS IoT Analytics console, choisissez Prepare et (si nécessaire) Channel, puis choisissez le nom du canal que vous avez créé précédemment.

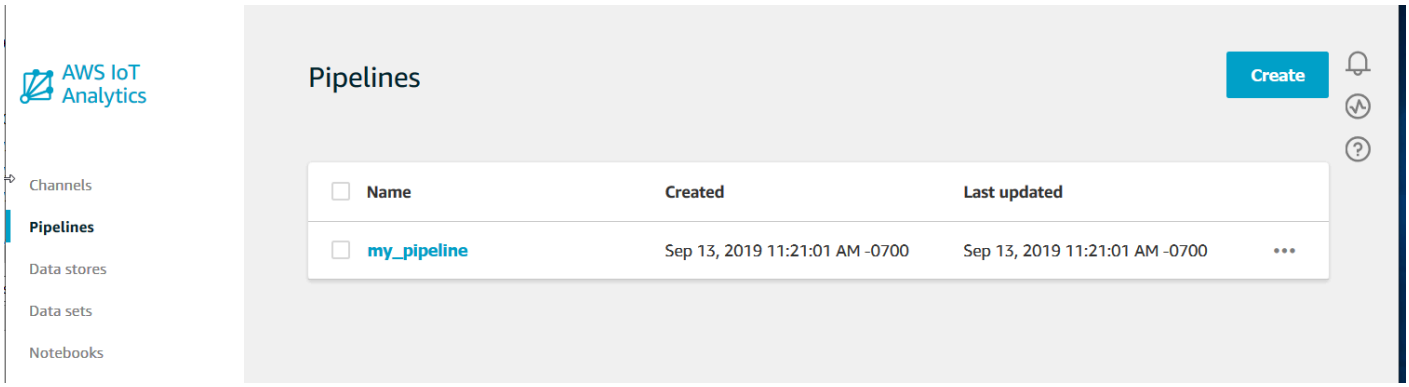


2. Sur la page de détails du canal, faites défiler vers le bas jusqu'à la section Monitoring (Surveillance). Ajustez la période affichée, le cas échéant, en choisissant l'un des indicateurs de temps (1h 3h 12h 1j 3j 1sem). Vous devriez voir une ligne graphique indiquant le nombre de messages ingérés dans ce canal pendant la période spécifiée.

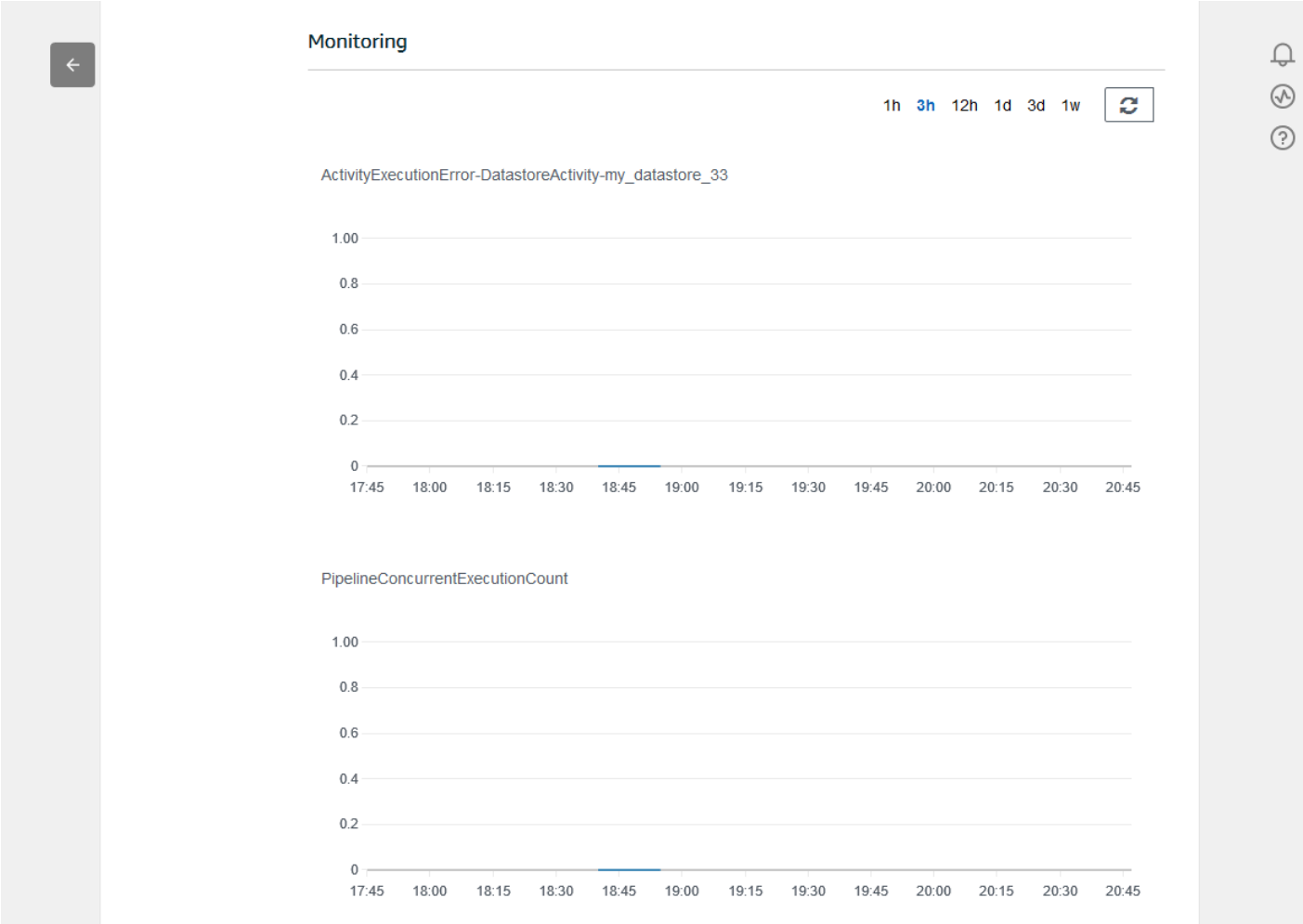


Une fonctionnalité de surveillance similaire existe pour vérifier l'exécution des activités du pipeline. Vous pouvez surveiller les erreurs d'exécution des activités sur la page des détails du pipeline. Si vous n'avez pas spécifié d'activités dans le cadre de votre pipeline, aucune erreur d'exécution ne devrait s'afficher.

1. Dans le volet de navigation de gauche de la AWS IoT Analytics console, choisissez Prepare, puis Pipelines, puis choisissez le nom d'un pipeline que vous avez créé précédemment.



2. Sur la page de détails du pipeline, faites défiler vers le bas jusqu'à la section Monitoring (Surveillance). Ajustez la période affichée, le cas échéant, en choisissant l'un des indicateurs de temps (1h 3h 12h 1j 3j 1sem). Vous devriez voir une ligne graphique indiquant le nombre d'erreurs d'exécution des activités du pipeline au cours de la période spécifiée.



Création d'un jeu de données

Vous extrayez des données d'un magasin de données en créant un ensemble de données SQL ou un ensemble de données conteneur. AWS IoT Analytics peut interroger les données pour répondre à des questions analytiques. Bien qu'un magasin de données ne soit pas une base de données, vous utilisez des expressions SQL pour interroger les données et produire des résultats qui sont stockés dans un ensemble de données.

Rubriques

- Interrogation de données

- Accès aux données demandées

Interrogation de données

Pour interroger les données, vous créez un ensemble de données. Un ensemble de données contient le code SQL que vous utilisez pour interroger le magasin de données ainsi qu'un calendrier facultatif qui répète la requête au jour et à l'heure de votre choix. Vous créez les plannings facultatifs à l'aide d'expressions similaires aux expressions CloudWatch de planning Amazon.

Exécutez la commande suivante pour créer un ensemble de données.

```
aws iotanalytics create-dataset --cli-input-json file://mydataset.json
```

Où le `mydataset.json` fichier contient le contenu suivant.

```
{
    "datasetName": "mydataset",
    "actions": [
        {
            "actionName":"myaction",
            "queryAction": {
                "sqlQuery": "select * from mydatastore"
            }
        }
    ]
}
```

Exécutez la commande suivante pour créer le contenu de l'ensemble de données en exécutant la requête.

```
aws iotanalytics create-dataset-content --dataset-name mydataset
```

Attendez quelques minutes que le contenu du jeu de données soit créé avant de continuer.

Accès aux données demandées

Le résultat de la requête est le contenu de votre ensemble de données, stocké sous forme de fichier, au format CSV. Le fichier est mis à votre disposition par le biais d'Amazon S3. L'exemple suivant

montre comment vous pouvez vérifier que vos résultats sont prêts et que vous pouvez télécharger le fichier.

Exécutez la commande suivante `get-dataset-content`.

```
aws iotanalytics get-dataset-content --dataset-name mydataset
```

Si votre ensemble de données contient des données, la sortie de `get-dataset-content` contient `"state": "SUCCEEDED"` dans le `status` champ, comme dans l'exemple suivant.

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
          "entryName": "someEntry",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

`dataURI` est une URL signée menant aux résultats. Elle est valide pendant une brève période (quelques heures). En fonction de votre flux de travail, vous souhaiterez peut-être toujours appeler `get-dataset-content` avant d'accéder au contenu, car l'appel de cette commande génère une nouvelle URL signée.

Exploration AWS IoT Analytics des données

Plusieurs options s'offrent à vous pour stocker, analyser et visualiser vos AWS IoT Analytics données.

Rubriques de cette page :

- Amazon S3
- AWS IoT Events

- Suite rapide
- Bloc-notes Jupyter

Amazon S3

Vous pouvez envoyer le contenu d'un ensemble de données vers un bucket Amazon Simple Storage Service (Amazon S3), ce qui permet l'intégration à vos lacs de données existants ou l'accès à partir d'applications internes et d'outils de visualisation. Voir le champ `contentDeliveryRules::destination::s3DestinationConfiguration` dans CreateDataset.

AWS IoT Events

Vous pouvez envoyer le contenu du jeu de données en entrée à AWS IoT Events un service qui vous permet de surveiller les appareils ou les processus pour détecter les défaillances ou les changements de fonctionnement, et de déclencher des actions supplémentaires lorsque de tels événements se produisent.

Pour ce faire, créez un ensemble de données en utilisant CreateDatasetet spécifiez une AWS IoT Events entrée dans le champ`contentDeliveryRules :: destination :: iotEventsDestinationConfiguration :: inputName`. Vous devez également spécifier le `roleArn` rôle qui AWS IoT Analytics autorise l'exécution de « iotevents : BatchPutMessage ». Chaque fois que le contenu de l'ensemble de données est créé, AWS IoT Analytics chaque entrée de contenu de l'ensemble de données est envoyée sous forme de message à l' AWS IoT Events entrée spécifiée. Par exemple, si votre ensemble de données contient :

```
"what","who","dt"
"overflow","sensor01","2019-09-16 09:04:00.000"
"overflow","sensor02","2019-09-16 09:07:00.000"
"underflow","sensor01","2019-09-16 11:09:00.000"
...
```

puis AWS IoT Analytics enverra des messages contenant des champs comme celui-ci :

```
{ "what": "overflow", "who": "sensor01", "dt": "2019-09-16 09:04:00.000" }
```

```
{ "what": "overflow", "who": "sensor02", "dt": "2019-09-16 09:07:00.000" }
```

et vous souhaiterez créer une AWS IoT Events entrée qui reconnaisse les champs qui vous intéressent (un ou plusieurs des`what`,`who`,`dt`) et créer un modèle de AWS IoT Events détecteur qui utilise ces champs de saisie lors d'événements pour déclencher des actions ou définir des variables internes.

Suite rapide

AWS IoT Analytics fournit une intégration directe avec Quick Suite. Quick Suite est un service d'analyse commerciale rapide que vous pouvez utiliser pour créer des visualisations, effectuer des analyses ad hoc et obtenir rapidement des informations commerciales à partir de vos données. Quick Suite permet aux entreprises de s'adapter à des centaines de milliers d'utilisateurs et fournit des performances réactives grâce à un puissant moteur en mémoire (SPICE). Quick Suite est disponible dans ces régions.

Bloc-notes Jupyter

AWS IoT Analytics les ensembles de données peuvent également être directement consommés par Jupyter Notebook afin d'effectuer des analyses avancées et une exploration de données. Jupyter Notebook est une solution open source. Vous pouvez les télécharger et installer depuis http://jupyter.org/install.html. Une intégration supplémentaire avec SageMaker AI, une solution de bloc-notes hébergée par Amazon, est également disponible.

Conservation de plusieurs versions d'ensembles de données

Vous pouvez choisir le nombre de versions du contenu de votre ensemble de données à conserver, et pendant combien de temps, en spécifiant les valeurs des `retentionPeriod` and `versioningConfiguration` champs de l'ensemble de données lorsque vous invoquez le CreateDatasetet UpdateDataset APIs:

```
...
"retentionPeriod": {
  "unlimited": "boolean",
  "numberOfDays": "integer"
},
"versioningConfiguration": {
  "unlimited": "boolean",
  "maxVersions": "integer"
},
```

```
...
```

Les paramètres de ces deux paramètres fonctionnent ensemble pour déterminer le nombre de versions du contenu des ensembles de données conservées, ainsi que leur durée, de la manière suivante.

	retentionPeriod [non spécifié]	retentionPeriod : illimité = VRAI, numberOfDays = non défini	retentionPeriod : illimité = FAUX, numberOfDays = X
versioningConfiguration : [non spécifié]	Seule la dernière version ainsi que la dernière version valide (si elles sont différentes) sont conservées pendant 90 jours.	Seule la dernière version ainsi que la dernière version valide (si elles sont différentes) sont conservées pendant une durée illimitée.	Seule la dernière version ainsi que la dernière version valide (si elles sont différentes) sont conservées pendant X jours.
versioningConfiguration : unlimited = TRUE, maxVersions non défini	Toutes les versions des 90 derniers jours seront conservées, quel qu'en soit le nombre.	Il n'y a aucune limite pour le nombre de versions conservées.	Toutes les versions des X derniers jours seront conservées, quel qu'en soit le nombre.
versioningConfiguration : unlimited = FALSE, maxVersions = Y	Pas plus de Y versions des 90 derniers jours seront conservées.	Jusqu'à Y versions seront conservées, quelle que soit leur ancienneté.	Pas plus de Y versions des X derniers jours seront conservées.

Syntaxe de la charge utile des messages

Les noms de champs des charges utiles (données) des messages que vous envoyez à AWS IoT Analytics :

- Ne doit contenir que des caractères alphanumériques et des traits de soulignement (_) ; aucun autre caractère spécial n'est autorisé
- Ils doivent commencer par un caractère alphabétique ou un trait de soulignement (_).
- Ils ne peuvent pas contenir de tirets (-).
- En termes d'expressions régulières : « `^[A-Za-z_]([A-Za-z0-9]*|[A-Za-z0-9][A-Za-z0-9_]*)$` ».
- Ils ne peuvent pas contenir plus de 255 caractères
- Ils ne sont pas sensibles à la casse. Les champs nommés « foo » et « FOO » dans la même charge utile sont considérés comme des doublons.

Par exemple, `{"temp_01": 29}` ou `{"_temp_01": 29}` sont valides, mais `{"temp-01": 29}`, `{"01_temp": 29}` ou `{"__temp_01": 29}` ne sont pas valides dans les charges utiles de messages.

Travailler avec des AWS IoT SiteWise données

AWS IoT SiteWise est un service géré que vous pouvez utiliser pour collecter, modéliser, analyser et visualiser des données provenant d'équipements industriels à grande échelle. Le service fournit un cadre de modélisation des actifs pour créer des représentations de vos appareils, processus et installations industriels.

Grâce aux modèles AWS IoT SiteWise d'actifs, vous pouvez définir les données d'équipement industriel à consommer et la manière de traiter vos données en indicateurs complexes. Vous pouvez configurer des modèles d'actifs pour collecter et traiter des données dans le AWS cloud. Pour plus d'informations, consultez le guide de AWS IoT SiteWiseel'utilisateur.

AWS IoT Analytics s'intègre à AWS IoT SiteWise ce qui vous permet d'exécuter et de planifier des requêtes SQL sur AWS IoT SiteWise les données. Pour commencer à interroger vos AWS IoT SiteWise données, créez un magasin de données en suivant les procédures décrites dans Configurer les paramètres de stockage dans le Guide de l'AWS IoT SiteWise utilisateur. Suivez ensuite les étapes indiquées dans Création d'un ensemble de AWS IoT SiteWise données avec des données (console) ou dans Création d'un ensemble de données avec AWS IoT SiteWise data (AWS CLI) pour créer un AWS IoT Analytics ensemble de données et exécuter une requête SQL sur vos données industrielles.

Rubriques

- Création d'un AWS IoT Analytics ensemble de données avec AWS IoT SiteWise des données
- Accéder au contenu du jeu de données
- Tutoriel : Interroger AWS IoT SiteWise des données dans AWS IoT Analytics

Création d'un AWS IoT Analytics ensemble de données avec AWS IoT SiteWise des données

Un AWS IoT Analytics ensemble de données contient des instructions et des expressions SQL que vous utilisez pour interroger des données dans votre magasin de données, ainsi qu'un calendrier facultatif qui répète la requête au jour et à l'heure que vous spécifiez. Vous pouvez utiliser des expressions similaires aux expressions CloudWatch de planification Amazon pour créer les plannings facultatifs.

> **Note**
>
> Un ensemble de données est généralement un ensemble de données qui peuvent ou non être organisées sous forme de tableau. En revanche, AWS IoT Analytics crée votre ensemble de données en appliquant une requête SQL aux données de votre magasin de données.

Suivez ces étapes pour commencer à créer un ensemble de données pour vos AWS IoT SiteWise données.

Rubriques

- Création d'un ensemble de AWS IoT SiteWise données avec des données (console)
- Création d'un ensemble de données avec AWS IoT SiteWise data (AWS CLI)

Création d'un ensemble de AWS IoT SiteWise données avec des données (console)

Suivez ces étapes pour créer un ensemble de données dans la AWS IoT Analytics console pour vos AWS IoT SiteWise données.

Pour créer un jeu de données

1. Dans le https://console.aws.amazon.com/iotanalytics/volet de navigation de gauche, sélectionnez Datasets.
2. Sur la page Créer un jeu de données, choisissez Create SQL.

3. Sur la page Spécifier les détails du jeu de données, spécifiez les détails de votre ensemble de données.

   a. Entrez un nom pour votre jeu de données.

   b. Pour Source du magasin de données, choisissez l'ID unique qui identifie votre magasin de AWS IoT SiteWise données.

   c. (Facultatif) Pour les balises, ajoutez une ou plusieurs balises personnalisées (paires clé-valeur) à votre ensemble de données.

4. Utilisez des expressions SQL pour interroger vos données et répondre à des questions analytiques.

   a. Dans le champ de requête Auteur, entrez une requête SQL qui utilise un caractère générique pour afficher jusqu'à cinq lignes de données.

   ```
   SELECT * FROM my_iotsitewise_datastore.asset_metadata LIMIT 5
   ```

   Pour plus d'informations sur les fonctionnalités SQL prises en charge dans AWS IoT Analytics, consultezExpressions SQL dans AWS IoT Analytics. Vous pouvez également consulter Tutoriel : Interroger AWS IoT SiteWise des données dans AWS IoT Analytics des exemples de requêtes statistiques qui peuvent fournir un aperçu de vos données.

   b. Vous pouvez choisir Tester la requête pour vérifier que votre saisie est correcte et pour afficher les résultats dans un tableau à la suite de la requête.

   > **Note**
   >
   > Étant donné que Amazon Athena le nombre maximum de requêtes en cours d'exécution est limité, vous devez limiter votre requête SQL à une taille raisonnable afin qu'elle ne s'exécute pas pendant une période prolongée.

5. (Facultatif) Lorsque vous créez le contenu d'un ensemble de données à partir de données issues d'une période spécifiée, certaines données peuvent ne pas arriver à temps pour être traitées. Pour autoriser un délai, vous pouvez spécifier un décalage, ou delta. Pour de plus amples informations, veuillez consulter Recevoir des notifications de données en retard via Amazon CloudWatch Events.

   Après avoir configuré un filtre de sélection de données sur la page Configurer le filtre de sélection de données, choisissez Next.

6. (Facultatif) Sur la page Définir le calendrier des requêtes, vous pouvez planifier l'exécution régulière de cette requête afin d'actualiser le jeu de données. Les plannings des ensembles de données peuvent être créés et modifiés à tout moment.

   > **Note**
   >
   > Données provenant d' AWS IoT SiteWise ingestations effectuées AWS IoT Analytics toutes les six heures. Nous vous recommandons de sélectionner une fréquence de six heures ou plus.

   Choisissez une option pour Fréquence, puis cliquez sur Suivant.

7. AWS IoT Analytics créera des versions du contenu de cet ensemble de données et stockera les résultats de vos analyses pour la période spécifiée. Nous recommandons 90 jours, mais vous pouvez choisir de définir votre politique de conservation personnalisée. Vous pouvez également limiter le nombre de versions stockées du contenu de votre ensemble de données.

   Après avoir sélectionné vos options sur la page Configurer les résultats de votre jeu de données, choisissez Next.

8. (Facultatif) Vous pouvez configurer les règles de livraison des résultats de votre jeu de données vers une destination spécifique, telle que AWS IoT Events.

   Après avoir sélectionné vos options sur la page Configurer les règles de diffusion du contenu du jeu de données, choisissez Next.

9. Passez en revue vos choix, puis choisissez Créer un ensemble de données.
10. Vérifiez que votre nouveau jeu de données apparaît sur la page Ensembles de données.

Création d'un ensemble de données avec AWS IoT SiteWise data (AWS CLI)

Exécutez les AWS CLI commandes suivantes pour commencer à interroger vos AWS IoT SiteWise données.

Les exemples présentés ici utilisent le AWS Command Line Interface (AWS CLI). Pour plus d'informations sur le AWS CLI, consultez le guide de AWS Command Line Interface l'utilisateur. Pour plus d'informations sur les commandes CLI disponibles pour AWS IoT Analytics, consultez iotanalytics dans la AWS Command Line Interface référence.

Pour créer un jeu de données

1. Exécutez la `create-dataset` commande suivante pour créer un ensemble de données.

```
aws iotanalytics create-dataset --cli-input-json file://my_dataset.json
```

Où le `my_dataset.json` fichier contient le contenu suivant.

```
{
    "datasetName": "my_dataset",
    "actions": [
        {
            "actionName":"my_action",
            "queryAction": {
                "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
 LIMIT 5"
            }
        }
    ]
}
```

Pour plus d'informations sur les fonctionnalités SQL prises en charge dans AWS IoT Analytics, consultezExpressions SQL dans AWS IoT Analytics. Vous pouvez également consulter Tutoriel : Interroger AWS IoT SiteWise des données dans AWS IoT Analytics des exemples de requêtes statistiques qui peuvent fournir un aperçu de vos données.

2. Exécutez la `create-dataset-content` commande suivante pour créer le contenu de votre ensemble de données en exécutant votre requête.

```
aws iotanalytics create-dataset-content --dataset-name my_dataset
```

Accéder au contenu du jeu de données

Le résultat de la requête SQL est le contenu de votre ensemble de données, stocké sous forme de fichier au format CSV. Le fichier est mis à votre disposition par le biais d'Amazon S3. Les étapes suivantes indiquent comment vérifier que vos résultats sont prêts et comment télécharger le fichier.

Rubriques

- Accédez au contenu du jeu de données dans AWS IoT Analytics (Console)

- Accédez au contenu du jeu de données dans AWS IoT Analytics (AWS CLI)

Accédez au contenu du jeu de données dans AWS IoT Analytics (Console)

Si votre ensemble de données contient des données, vous pouvez prévisualiser et télécharger les résultats de vos requêtes SQL dans AWS IoT Analytics la console.

Pour accéder aux résultats de votre AWS IoT Analytics jeu de données

1. Dans la console, sur la page Ensembles de données, choisissez le nom du jeu de données auquel vous souhaitez accéder.
2. Sur la page récapitulative du jeu de données, choisissez l'onglet Contenu.
3. Dans le tableau de contenu du jeu de données, choisissez le nom de la requête dont vous souhaitez prévisualiser les résultats ou téléchargez un fichier csv contenant les résultats.

Accédez au contenu du jeu de données dans AWS IoT Analytics (AWS CLI)

Si votre ensemble de données contient des données, vous pouvez prévisualiser et télécharger les résultats de vos requêtes SQL.

Les exemples présentés ici utilisent le AWS Command Line Interface (AWS CLI). Pour plus d'informations sur le AWS CLI, consultez le guide de AWS Command Line Interface l'utilisateur. Pour plus d'informations sur les commandes CLI disponibles pour AWS IoT Analytics, consultez iotanalytics dans la AWS Command Line Interface référence.

Pour accéder aux résultats AWS IoT Analytics de votre jeu de données (AWS CLI)

1. Exécutez la `get-dataset-content` commande suivante pour afficher le résultat de votre requête.

```
aws iotanalytics get-dataset-content --dataset-name my_iotsitewise_dataset
```

2. Si votre ensemble de données contient des données, la sortie de`get-dataset-content`, se `"state": "SUCCEEDED"` trouve dans le `status` champ, comme dans l'exemple suivant.

```
{
    "timestamp": 1508189965.746,
    "entries": [
        {
```

```
          "entryName": "my_entry_name",
          "dataURI": "https://aws-iot-analytics-datasets-f7253800-859a-472c-aa33-
e23998b31261.s3.amazonaws.com/results/f881f855-c873-49ce-abd9-b50e9611b71f.csv?X-
Amz-"

        }
    ],
    "status": {
      "state": "SUCCEEDED",
      "reason": "A useful comment."
    }
}
```

3. La sortie de `get-dataset-content` inclut un`dataURI`, qui est une URL signée vers les résultats de sortie. Elle est valide pendant une brève période (quelques heures). Accédez à l'`dataURI`URL pour accéder aux résultats de vos requêtes SQL.

> **Note**
>
> En fonction de votre flux de travail, vous souhaiterez peut-être toujours appeler `get-dataset-content` avant d'accéder au contenu, car l'appel de cette commande génère une nouvelle URL signée.

Tutoriel : Interroger AWS IoT SiteWise des données dans AWS IoT Analytics

Ce didacticiel explique comment interroger AWS IoT SiteWise des données dans AWS IoT Analytics. Le didacticiel utilise les données d'une démonstration AWS IoT SiteWise qui fournit un exemple de jeu de données pour un parc éolien.

> **Important**
>
> Vous serez facturé pour les ressources que cette démonstration crée et consomme.

Rubriques

- Prérequis
- Charger et vérifier les données

- Exploration des données
- Exécuter des requêtes statistiques
- Nettoyer les ressources de votre didacticiel

Prérequis

Pour ce didacticiel, vous avez besoin des ressources suivantes :

- Vous devez avoir un AWS compte pour commencer à utiliser AWS IoT SiteWise et AWS IoT Analytics. Si vous n'en avez pas, suivez les procédures décrites dans Pour créer un AWS compte.
- Ordinateur de développement exécutant Windows, macOS, Linux ou Unix pour accéder à la console AWS Management Console. Pour plus d'informations, consultez Démarrer avec le AWS Management Console.
- AWS IoT SiteWise des données qui définissent AWS IoT SiteWise des modèles et des actifs et diffusent des données représentant des données provenant d'équipements de parcs éoliens. Pour créer vos données, suivez les étapes décrites dans la section Création de la AWS IoT SiteWise démo dans le guide de AWS IoT SiteWise l'utilisateur.
- Les données de votre équipement de parc éolien de AWS IoT SiteWise démonstration dans un magasin de données existant que vous gérez. Pour plus d'informations sur la création d'un magasin de données pour vos AWS IoT SiteWise données, voir Configurer les paramètres de stockage dans le Guide de AWS IoT SiteWise l'utilisateur.

> **Note**
>
> Vos AWS IoT SiteWise métadonnées apparaissent dans votre AWS IoT SiteWise banque de données peu après leur création ; toutefois, l'affichage de vos données brutes peut prendre jusqu'à six heures. En attendant, vous pouvez créer un AWS IoT Analytics ensemble de données et exécuter des requêtes sur vos métadonnées.

Étape suivante

Charger et vérifier les données

Charger et vérifier les données

Les données que vous interrogez dans ce didacticiel sont un exemple de jeu de AWS IoT SiteWise données qui modélise les éoliennes d'un parc éolien.

> **Note**
>
> Vous allez interroger trois tables de votre magasin de données tout au long de ce didacticiel :
>
> - `raw`- Contient des données brutes non traitées pour chaque actif.
> - `asset_metadata`- Contient des informations générales sur chaque actif.
> - `asset_hierarchy_metadata`- Contient des informations sur les relations entre les actifs.

Pour exécuter les requêtes SQL de ce didacticiel

1. Suivez les étapes décrites dans Création d'un ensemble de AWS IoT SiteWise données avec des données (console) ou Création d'un ensemble de données avec AWS IoT SiteWise data (AWS CLI) pour créer un AWS IoT Analytics ensemble de données pour vos AWS IoT SiteWise données.
2. Pour mettre à jour votre requête de jeu de données tout au long de ce didacticiel, procédez comme suit.

   a. Dans la AWS IoT Analytics console, sur la page Ensembles de données, choisissez le nom du jeu de données que vous avez créé sur la page précédente.

   b. Sur la page de résumé du jeu de données, choisissez Modifier pour modifier votre requête SQL.

   c. Pour afficher les résultats dans un tableau à la suite de la requête, choisissez Tester la requête.

   Vous pouvez également exécuter la `update-dataset` commande suivante pour modifier la requête SQL à l'aide du AWS CLI.

   ```
   aws iotanalytics update-dataset --cli-input-json file://update-query.json
   ```

   Contenu de `update-query.json` :

```
{
    "datasetName": "my_dataset",
    "actions": [
        {
            "actionName": "myDatasetUpdateAction",
            "queryAction": {
                "sqlQuery": "SELECT * FROM my_iotsitewise_datastore.asset_metadata
 LIMIT 3"
            }
        }
    ]
}
```

3. Dans la AWS IoT Analytics console ou à l'aide du AWS CLI, exécutez la requête suivante sur vos données pour vérifier que votre `asset_metadata` table s'est correctement chargée.

```
SELECT COUNT(*) FROM my_iotsitewise_datastore.asset_metadata
```

De même, vous pouvez vérifier que vos `raw` tables `asset_hierarchy_metadata` et ne sont pas vides.

Étape suivante

Exploration des données

Exploration des données

Une fois vos AWS IoT SiteWise données créées et chargées dans un magasin de données, vous pouvez créer un AWS IoT Analytics ensemble de données et exécuter des requêtes SQL AWS IoT Analytics pour obtenir des informations sur vos actifs. Les requêtes suivantes montrent comment explorer vos données avant d'exécuter des requêtes statistiques.

Pour explorer vos données à l'aide de requêtes SQL

1. Affichez un échantillon de colonnes et de valeurs dans chaque tableau, comme dans le tableau brut.

```
SELECT * FROM my_iotsitewise_datastore.raw LIMIT 5
```

2. `SELECT DISTINCT`Utilisez-le pour interroger votre `asset_metadata` table et répertorier les noms (uniques) de vos AWS IoT SiteWise actifs.

```
SELECT DISTINCT assetname FROM my_iotsitewise_datastore.asset_metadata ORDER BY
 assetname
```

3. Pour répertorier les informations relatives aux propriétés d'un AWS IoT SiteWise actif en particulier, utilisez la `WHERE` clause.

```
SELECT assetpropertyname,
     assetpropertyunit,
     assetpropertydatatype
FROM my_iotsitewise_datastore.asset_metadata
WHERE assetname = 'Demo Turbine Asset 2'
```

4. Avec AWS IoT Analytics, vous pouvez joindre des données provenant de deux ou plusieurs tables de votre magasin de données, comme dans l'exemple suivant.

```
SELECT * FROM my_iotsitewise_datastore.raw AS raw
JOIN my_iotsitewise_datastore.asset_metadata AS asset_metadata
ON raw.seriesId = asset_metadata.timeseriesId
```

Pour afficher toutes les relations entre vos actifs, utilisez les `JOIN` fonctionnalités de la requête suivante.

```
SELECT DISTINCT parent.assetName as "Parent name",
    child.assetName AS "Child name"
FROM (
    SELECT sourceAssetId AS parent,
            targetAssetId AS child
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS relations
JOIN my_iotsitewise_datastore.asset_metadata AS child
    ON relations.child = child.assetId
JOIN my_iotsitewise_datastore.asset_metadata AS parent
    ON relations.parent = parent.assetId
```

Étape suivante

Exécuter des requêtes statistiques

Exécuter des requêtes statistiques

Maintenant que vous avez exploré vos AWS IoT SiteWise données, vous pouvez exécuter des requêtes statistiques qui fournissent des informations précieuses sur votre équipement industriel. Les requêtes suivantes illustrent certaines des informations que vous pouvez récupérer.

Pour exécuter des requêtes statistiques sur les données de AWS IoT SiteWise démonstration d'un parc éolien

1. Exécutez la commande SQL suivante pour rechercher les dernières valeurs de toutes les propriétés avec des valeurs numériques pour un actif particulier (Demo Turbine Asset 4).

```
SELECT assetName,
    assetPropertyName,
    assetPropertyUnit,
    max_by(value, timeInSeconds) AS Latest
FROM (
    SELECT *,
        CASE assetPropertyDataType
        WHEN 'DOUBLE' THEN
        cast(doubleValue AS varchar)
        WHEN 'INTEGER' THEN
        cast(integerValue AS varchar)
        WHEN 'STRING' THEN
        stringValue
        WHEN 'BOOLEAN' THEN
        cast(booleanValue AS varchar)
        ELSE NULL
        END AS value
    FROM my_iotsitewise_datastore.asset_metadata AS asset_metadata
    JOIN my_iotsitewise_datastore.raw AS raw
        ON raw.seriesId = asset_metadata.timeSeriesId
    WHERE startYear=2021
        AND startMonth=7
        AND startDay=8
        AND assetName='Demo Turbine Asset 4'
)
GROUP BY assetName, assetPropertyName, assetPropertyUnit
```

2. Joignez les tables de métadonnées et votre table brute pour identifier les propriétés de vitesse maximale du vent pour tous les actifs, en plus de leurs actifs parents.

```
SELECT child_assets_data_set.parentAssetId,
        child_assets_data_set.childAssetId,
        asset_metadata.assetPropertyId,
        asset_metadata.assetPropertyName,
        asset_metadata.timeSeriesId,
        raw_data_set.max_speed
FROM (
   SELECT sourceAssetId AS parentAssetId,
        targetAssetId AS childAssetId
    FROM my_iotsitewise_datastore.asset_hierarchy_metadata
    WHERE associationType = 'CHILD'
)
AS child_assets_data_set
JOIN mls_demo.asset_metadata AS asset_metadata
    ON asset_metadata.assetId = child_assets_data_set.childAssetId
JOIN (
    SELECT seriesId, MAX(doubleValue) AS max_speed
    FROM my_iotsitewise_datastore.raw
    GROUP BY seriesId
)
AS raw_data_set
ON raw_data_set.seriesId = asset_metadata.timeseriesid
WHERE assetPropertyName = 'Wind Speed'
ORDER BY max_speed DESC
```

3. Pour trouver la valeur moyenne d'une propriété particulière (Vitesse du vent) pour un actif (Demo Turbine Asset 2), exécutez la commande SQL suivante. Vous devez le `my_bucket_id` remplacer par l'ID de votre bucket.

```
SELECT AVG(doubleValue) as "Average wind speed"
FROM my_iotsitewise_datastore.raw
WHERE seriesId =
    (SELECT timeseriesId
    FROM my_iotsitewise_datastore.asset_metadata as asset_metadata
    WHERE asset_metadata.assetname = 'Demo Turbine Asset 2'
            AND asset_metadata.assetpropertyname = 'Wind Speed')
```

Étape suivante

Nettoyer les ressources de votre didacticiel

Nettoyer les ressources de votre didacticiel

Une fois le didacticiel terminé, nettoyez vos ressources pour éviter d'encourir des frais.

Pour supprimer votre AWS IoT SiteWise démo

La AWS IoT SiteWise démo se supprime d'elle-même au bout d'une semaine. Si vous avez terminé d'utiliser les ressources de démonstration, vous pouvez supprimer la démo plus tôt. Pour supprimer la démo manuellement, procédez comme suit.

1. Accédez à la console CloudFormation.
2. Choisissez IoTSiteWiseDemoAssets dans la liste Stacks (Piles).
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, saisissez 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 ne parvient pas à être supprimée à nouveau, suivez les étapes de la CloudFormation console pour ignorer les ressources qui n'ont pas pu être supprimées, puis réessayez.

Pour supprimer votre banque de données

- Pour supprimer votre magasin de données géré, exécutez la commande CLI`delete-datastore`, comme dans l'exemple suivant.

```
aws iotanalytics delete-datastore --datastore-name my_IotSiteWise_datastore
```

Pour supprimer votre AWS IoT Analytics jeu de données

- Pour supprimer votre ensemble de données, exécutez la commande CLI`delete-dataset`, comme dans l'exemple suivant. Il n'est pas nécessaire de supprimer le contenu de l'ensemble de données avant d'effectuer cette opération.

```
aws iotanalytics delete-dataset --dataset-name my_dataset
```

> **Note**
>
> Cette commande ne produit aucun résultat.

Activités liées au pipeline

Le pipeline fonctionnel le plus simple connecte un canal à un magasin de données, ce qui fait de lui un pipeline avec deux activités : une activité `channel` et une activité `datastore`. Vous pouvez obtenir une traitement plus puissant des messages en ajoutant d'autres activités à votre pipeline.

Vous pouvez utiliser cette RunPipelineActivityopération pour simuler les résultats de l'exécution d'une activité de pipeline sur une charge utile de message que vous fournissez. Cela peut vous être utile lorsque vous développez et débugez les activités de votre pipeline. RunPipelineActivity L'exemple montre comment il est utilisé.

Activité de la chaîne

La première activité d'un pipeline doit être celle `channel` qui détermine la source des messages à traiter.

```
{
    "channel": {
        "name": "MyChannelActivity",
        "channelName": "mychannel",
        "next": "MyLambdaActivity"
    }
}
```

Activité de la banque de données

L'activité `datastore`, qui spécifie l'emplacement où stocker les données traitées, est la dernière activité.

```
{
    "datastore": {
        "name": "MyDatastoreActivity",
        "datastoreName": "mydatastore"
    }
}
```

AWS Lambda activité

Vous pouvez utiliser une **lambda**activité pour effectuer des traitements complexes sur des messages. Par exemple, vous pouvez enrichir les messages avec des données issues d'opérations d'API externes ou filtrer les messages en fonction de la logique d'Amazon DynamoDB. Toutefois, vous ne pouvez pas utiliser cette activité de pipeline pour ajouter des messages supplémentaires ou supprimer des messages existants avant d'accéder à un magasin de données.

La AWS Lambda fonction utilisée dans une **lambda**activité doit recevoir et renvoyer un tableau d'objets JSON. Pour obtenir un exemple, consultez the section called "Exemple de fonction Lambda 1".

Pour AWS IoT Analytics autoriser l'appel de votre fonction Lambda, vous devez ajouter une politique. Par exemple, exécutez la commande CLI suivante et *exampleFunctionName* remplacez-la par le nom de votre fonction Lambda, *123456789012* remplacez-la par votre ID de AWS compte et utilisez le Amazon Resource Name (ARN) du pipeline qui appelle la fonction Lambda donnée.

```
aws lambda add-permission --function-name exampleFunctionName --
action lambda:InvokeFunction --statement-id iotanalytics --principal
 iotanalytics.amazonaws.com --source-account 123456789012 --source-arn
 arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline
```

La commande renvoie ce qui suit :

```
{
    "Statement": "{\"Sid\":\"iotanalyticsa\",\"Effect\":\"Allow\",
\"Principal\":{\"Service\":\"iotanalytics.amazonaws.com\"},\"Action\":
\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:aws-region:aws-
account:function:exampleFunctionName\",\"Condition\":{\"StringEquals\":
{\"AWS:SourceAccount\":\"123456789012\"},\"ArnLike\":{\"AWS:SourceArn\":
\"arn:aws:iotanalytics:us-east-1:123456789012:pipeline/examplePipeline\"}}}"
}
```

Pour plus d'informations, consultez Utilisation des stratégies fondées sur les ressources pour AWS Lambda dans le Guide de l'utilisateur AWS Lambda .

Exemple de fonction Lambda 1

Dans cet exemple, la fonction Lambda ajoute des informations basées sur les données du message d'origine. Un appareil publie un message avec une charge utile similaire à l'exemple suivant.

```
{
  "thingid": "00001234abcd",
  "temperature": 26,
  "humidity": 29,
  "location": {
    "lat": 52.4332935,
    "lon": 13.231694
  },
  "ip": "192.168.178.54",
  "datetime": "2018-02-15T07:06:01"
}
```

Et le périphérique a la définition de pipeline suivante.

```
{
    "pipeline": {
        "activities": [
            {
                "channel": {
                    "channelName": "foobar_channel",
                    "name": "foobar_channel_activity",
                    "next": "lambda_foobar_activity"
                }
            },
            {
                "lambda": {
                    "lambdaName": "MyAnalyticsLambdaFunction",
                    "batchSize": 5,
                    "name": "lambda_foobar_activity",
                    "next": "foobar_store_activity"
                }
            },
            {
                "datastore": {
                    "datastoreName": "foobar_datastore",
                    "name": "foobar_store_activity"
                }
            }
        ],
        "name": "foobar_pipeline",
        "arn": "arn:aws:iotanalytics:eu-west-1:123456789012:pipeline/foobar_pipeline"
    }
```

```
}
```

La fonction Lambda Python (`MyAnalyticsLambdaFunction`) suivante ajoute l' GMaps URL et la température, en degrés Fahrenheit, au message.

```
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def c_to_f(c):
    return 9.0/5.0 * c + 32

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))
    maps_url = 'N/A'

    for e in event:
        #e['foo'] = 'addedByLambda'
        if 'location' in e:
            lat = e['location']['lat']
            lon = e['location']['lon']
            maps_url = "http://maps.google.com/maps?q={},{}".format(lat,lon)

        if 'temperature' in e:
            e['temperature_f'] = c_to_f(e['temperature'])

        logger.info("maps_url: {}".format(maps_url))
        e['maps_url'] = maps_url

    logger.info("event after processing: {}".format(event))

    return event
```

Exemple de fonction Lambda 2

Une technique utile consiste à compresser et sérialiser les charges utiles de messages afin de réduire les coûts de transport et de stockage. Dans ce deuxième exemple, la fonction Lambda suppose que la charge utile du message représente un original JSON, qui a été compressé puis codé en base64 (sérialisé) sous forme de chaîne. Elle renvoie le JSON d'origine.

```
import base64
import gzip
import json
import logging
import sys

# Configure logging
logger = logging.getLogger()
logger.setLevel(logging.INFO)
streamHandler = logging.StreamHandler(stream=sys.stdout)
formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
streamHandler.setFormatter(formatter)
logger.addHandler(streamHandler)

def decode_to_bytes(e):
    return base64.b64decode(e)

def decompress_to_string(binary_data):
    return gzip.decompress(binary_data).decode('utf-8')

def lambda_handler(event, context):
    logger.info("event before processing: {}".format(event))

    decompressed_data = []

    for e in event:
        binary_data = decode_to_bytes(e)
        decompressed_string = decompress_to_string(binary_data)

        decompressed_data.append(json.loads(decompressed_string))

    logger.info("event after processing: {}".format(decompressed_data))

    return decompressed_data
```

AddAttributes activité

Une activité `addAttributes` ajoute des attributs en fonction des attributs existants dans le message. Cela vous permet de modifier la forme du message avant qu'il ne soit enregistré. Par exemple, vous pouvez utiliser `addAttributes` pour normaliser des données provenant de différentes générations de microprogrammes.

Tenez compte du message d'entrée suivant.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ]
    }
}
```

L'`addAttributes`activité ressemble à ce qui suit.

```
{
    "addAttributes": {
        "name": "MyAddAttributesActivity",
        "attributes": {
            "device.id": "id",
            "device.coord[0]": "lat",
            "device.coord[1]": "lon"
        },
        "next": "MyRemoveAttributesActivity"
    }
}
```

Cette activité déplace l'ID de l'appareil vers le niveau racine et extrait la valeur du `coord` tableau, en la promouvant au rang d'attributs de niveau supérieur appelés `lat` et`lon`. À la suite de cette activité, le message d'entrée est transformé selon l'exemple suivant.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
```

```
    "lat": 47.6,
    "lon": -122.3
}
```

L'attribut d'appareil d'origine est toujours présent. Si vous souhaitez le supprimer, vous pouvez utiliser l'activité `removeAttributes`.

RemoveAttributes activité

Une activité `removeAttributes` permet de supprimer des attributs d'un message. Par exemple, compte tenu du message issu de l'`addAttributes`activité.

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6, -122.3 ]
    },
    "id": "device-123",
    "lat": 47.6,
    "lon": -122.3
}
```

Pour normaliser ce message afin qu'il n'inclue que les données requises au niveau racine, effectuez l'`removeAttributes`activité suivante.

```
{
    "removeAttributes": {
        "name": "MyRemoveAttributesActivity",
        "attributes": [
            "device"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

Il en résulte que le message suivant circule le long du pipeline.

```
{
    "id": "device-123",
    "lat": 47.6,
```

```
    "lon": -122.3
}
```

SelectAttributes activité

L'activité `selectAttributes` crée un message en utilisant uniquement les attributs spécifiés du message d'origine. Tous les autres attribut sont abandonnés. `selectAttributes` crée de nouveaux attributs sous la racine du message uniquement. Donc, avec le message suivant :

```
{
    "device": {
        "id": "device-123",
        "coord": [ 47.6152543, -122.3354883 ],
        "temp": 50,
        "hum": 40
    },
    "light": 90
}
```

et cette activité :

```
{
    "selectAttributes": {
        "name": "MySelectAttributesActivity",
        "attributes": [
            "device.temp",
            "device.hum",
            "light"
        ],
        "next": "MyDatastoreActivity"
    }
}
```

Le résultat est le message suivant qui circule dans le pipeline.

```
{
    "temp": 50,
    "hum": 40,
    "light": 90
}
```

Là encore, `selectAttributes` ne peut créer que des objets à la racine.

Activité du filtre

Une activité `filter` filtre un message en fonction de ses attributs. L'expression utilisée dans cette activité ressemble à une `WHERE` clause SQL, qui doit renvoyer une valeur booléenne.

```
{
    "filter": {
        "name": "MyFilterActivity",
        "filter": "temp > 40 AND hum < 20",
        "next": "MyDatastoreActivity"
    }
}
```

DeviceRegistryEnrich activité

Cette `deviceRegistryEnrich` activité vous permet d'ajouter des données du registre de l' AWS IoT appareil à la charge utile de vos messages. Par exemple, avec le message suivant :

```
{
    "temp": 50,
    "hum": 40,
    "device" {
        "thingName": "my-thing"
    }
}
```

et une activité `deviceRegistryEnrich` qui se présente sous la forme suivante :

```
{
    "deviceRegistryEnrich": {
        "name": "MyDeviceRegistryEnrichActivity",
        "attribute": "metadata",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

Le message de sortie ressemble maintenant à cet exemple.

```
{
    "temp" : 50,
    "hum" : 40,
    "device" {
        "thingName" : "my-thing"
    },
    "metadata" : {
        "defaultClientId": "my-thing",
        "thingTypeName": "my-thing",
        "thingArn": "arn:aws:iot:us-east-1:<your-account-number>:thing/my-thing",
        "version": 1,
        "thingName": "my-thing",
        "attributes": {},
        "thingId": "aaabbbccc-dddeeef-gghh-jjkk-llmmnnoopp"
    }
}
```

Vous devez spécifier un rôle dans le champ `roleArn` de l'activité. Les autorisations appropriées doivent lui être associées. Le rôle doit disposer d'une politique d'autorisation similaire à celle décrite dans l'exemple suivant.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeThing"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

et une stratégie d'approbation qui se présente sous la forme suivante :

JSON

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

DeviceShadowEnrich activité

Une `deviceShadowEnrich` activité ajoute des informations du service AWS IoT Device Shadow à un message. Par exemple, avec le message suivant :

```
{
    "temp": 50,
    "hum": 40,
    "device": { "thingName": "my-thing" }
}
```

et l'activité `deviceShadowEnrich` suivante :

```
{
    "deviceShadowEnrich": {
        "name": "MyDeviceShadowEnrichActivity",
        "attribute": "shadow",
        "thingName": "device.thingName",
        "roleArn": "arn:aws:iam::<your-account-number>:role:MyEnrichRole",
        "next": "MyDatastoreActivity"
    }
}
```

Le résultat est un message qui ressemble à l'exemple suivant.

```
{
    "temp": 50,
    "hum": 40,
    "device": {
        "thingName": "my-thing"
    },
    "shadow": {
        "state": {
            "desired": {
                "attributeX": valueX, ...
            },
            "reported": {
                "attributeX": valueX, ...
            },
            "delta": {
                "attributeX": valueX, ...
            }
        },
        "metadata": {
            "desired": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            },
            "reported": ": {
                "attribute1": {
                    "timestamp": timestamp
                }, ...
            }
        },
        "timestamp": timestamp,
        "clientToken": "token",
        "version": version
    }
}
```

Vous devez spécifier un rôle dans le champ `roleArn` de l'activité. Les autorisations appropriées doivent lui être associées. Le rôle doit avoir une politique d'autorisation semblable à la suivante.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetThingShadow"
            ],
            "Resource": [
            "arn:aws:iot:us-east-1:123456789012:thing/your-thingName"
            ]
        }
    ]
}
```

et une stratégie d'approbation qui se présente sous la forme suivante :

JSON

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

Activité mathématique

Une activité `math` calcule une expression arithmétique à l'aide des attributs du message. L'expression doit renvoyer un nombre. Par exemple, avec le message d'entrée suivant :

```
{
    "tempF": 50,
}
```

après traitement par l'activité `math` suivante :

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "(tempF - 32) / 2",
        "attribute": "tempC",
        "next": "MyDatastoreActivity"
    }
}
```

le message résultant ressemble à :

```
{
    "tempF" : 50,
    "tempC": 9
}
```

Opérateurs et fonctions d'activités mathématiques

Vous pouvez utiliser les opérateurs suivants dans une activité `math` :

+	addition
-	soustraction
*	multiplication
/	division

%	modulo

Vous pouvez utiliser les fonctions suivantes dans une fonction activité `math` :

- abs(Decimal)
- acos(Decimal)
- asin(Decimal)
- atan(Decimal)
- atan2(Decimal, Decimal)
- ceil(Decimal)
- cos(Decimal)
- cosh(Decimal)
- exp(Decimal)
- ln(Decimal)
- log(Decimal)
- mod(Decimal, Decimal)
- power(Decimal, Decimal)
- round(Decimal)
- sign(Decimal)
- sin(Decimal)
- sinh(Decimal)
- sqrt(Decimal)
- tan(Decimal)
- tanh(Decimal)
- trunc (décimal, entier)

abs(Decimal)

Il renvoie la valeur absolue d'un nombre.

Exemples : `abs(-5)` renvoie 5.

Type d'argument	Résultat
`Int`	`Int`, la valeur absolue de l'argument.
`Decimal`	`Decimal`, la valeur absolue de l'argument
`Boolean`	`Undefined` .
`String`	`Decimal`. Le résultat est la valeur absolue de l'argument. Si la chaîne ne peut être pas convertie, le résultat est `Undefined` .
Tableau	`Undefined` .
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

acos(Decimal)

Renvoie le cosinus inverse d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `acos(0)` = 1.5707963267948966

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le cosinus inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Decimal`	`Decimal` (avec double précision), le cosinus inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Boolean`	`Undefined` .

Type d'argument	Résultat
`String`	`Decimal`(avec une double précision) le cosinus inverse de l'argument. Si la chaîne ne peut être pas convertie, le résultat est `Undefined` . Des résultats imaginaires sont retournés sous la forme `Undefined` .
Tableau	`Undefined` .
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

asin(Decimal)

Renvoie le sinus inverse d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `asin(0)` = 0.0

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le sinus inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Decimal`	`Decimal` (avec double précision), le sinus inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), le sinus inverse de l'argument. Si la chaîne ne peut être pas convertie, le résultat est `Undefined` . Des

Type d'argument	Résultat
	résultats imaginaires sont retournés sous la forme `Undefined` .
Tableau	`Undefined` .
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

atan(Decimal)

Renvoie la tangente inverse d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `atan(0)` = 0.0

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), la tangente inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Decimal`	`Decimal` (avec double précision), la tangente inverse de l'argument. Des résultats imaginair es sont retournés sous la forme `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), la tangente inverse de l'argument. Si la chaîne ne peut être pas convertie, le résultat est `Undefined` . Des résultats imaginaires sont retournés sous la forme `Undefined` .
Tableau	`Undefined` .

Type d'argument	Résultat
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

atan2(Decimal, Decimal)

Il renvoie l'angle en radians, entre l'axe des X positifs et le point (x, y) défini dans les deux arguments. L'angle est positif pour les angles dans le sens antihoraire (demi-plan supérieur, y > 0) et négatif pour les angles dans le sens des aiguilles d'une montre. Les `Decimal` arguments sont arrondis avec une double précision avant l'application de la fonction.

Exemples : `atan(1, 0)` = 1.5707963267948966

Type d'argument	Type d'argument	Résultat
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal`(avec double précision), l'angle entre l'axe X et le point (x, y) spécifié
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	`Decimal`, la tangente inverse du point décrit. Si une chaîne ne peut pas être convertie, le résultat est `Undefined` .
Autre valeur	Autre valeur	`Undefined` .

ceil(Decimal)

Arrondit la valeur `Decimal` donnée à la valeur `Int` supérieure la plus proche.

Exemples :

`ceil(1.2)` = 2

`ceil(11.2)` = -1

Type d'argument	Résultat
`Int`	`Int`, la valeur d'argument.
`Decimal`	`Int`, la chaîne est convertie `Decimal` et arrondie à la valeur la plus proche`Int`. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
Autre valeur	`Undefined` .

cos(Decimal)

Renvoie le cosinus d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `cos(0)` = 1

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le cosinus de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined` .
`Decimal`	`Decimal` (avec double précision), le cosinus de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), le cosinus de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` . Des résultats imaginaires sont retournés sous la forme `Undefined` .
Tableau	`Undefined` .

Type d'argument	Résultat
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

cosh(Decimal)

Renvoie le cosinus hyperbolique d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `cosh(2.3)` = 5.037220649268761

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le cosinus hyperbolique de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined` .
`Decimal`	`Decimal` (avec double précision), le cosinus hyperbolique de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined` .
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), le cosinus hyperbolique de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` . Des résultats imaginaires sont retournés sous la forme `Undefined` .
Tableau	`Undefined` .

Type d'argument	Résultat
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

exp(Decimal)

Renvoie la valeur e augmentée à l'argument décimal. `Decimal`les arguments sont arrondis à une double précision avant l'application de la fonction.

Exemples : `exp(1)` = 1

Type d'argument	Résultat
`Int`	`Decimal`(avec double précision), e^argument.
`Decimal`	`Decimal`(avec double précision), e^argument
`String`	`Decimal`(avec double précision), e^argumen t. Si le `String` ne peut pas être converti en a`Decimal`, le résultat est`Undefined` .
Autre valeur	`Undefined` .

ln(Decimal)

Renvoie le logarithme naturel de l'argument Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `ln(e)` = 1

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le logarithme naturel de l'argument.

Type d'argument	Résultat
`Decimal`	`Decimal`(avec double précision), le logarithme naturel de l'argument
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), le logarithm e naturel de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
Tableau	`Undefined` .
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

log(Decimal)

Renvoie le logarithme 10 de base de l'argument Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `log(100)` = 2.0

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le logarithme de base 10 de l'argument.
`Decimal`	`Decimal` (avec double précision), le logarithme de base 10 de l'argument.
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), le logarithme de base 10 de l'argument. Si la valeur `String`

Type d'argument	Résultat
	ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
Tableau	`Undefined` .
Objet	`Undefined` .
Null	`Undefined` .
Non défini	`Undefined` .

mod(Decimal, Decimal)

Renvoie le reste de la division entre le premier argument et le deuxième argument. Vous pouvez également l'utiliser `%` comme opérateur infixe pour la même fonctionnalité modulo.

Exemples : `mod(8, 3)` = 2

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	`Int`, le premier argument modulo du second argument.
`Int` / `Decimal`	`Int` / `Decimal`	`Decimal`, le premier argument modulo du second argument.
`String` / `Int` / `Decimal`	`String` / `Int` / `Decimal`	Si toutes les chaînes sont converties en`Decimals`, le résultat est le premier argument modulo le second argument. Sinon la valeur est renvoy, `Undefined` .
Autre valeur	Autre valeur	`Undefined` .

power(Decimal, Decimal)

Renvoie le premier argument augmenté vers le deuxième argument. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `power(2, 5)` = 32,0

Type d'argument 1	Type d'argument 2	Sortie
`Int` / `Decimal`	`Int` / `Decimal`	Une valeur `Decimal` (avec double précision), le premier argument renvoyé à la puissance du deuxième argument.
`Int` / `Decimal` / `String`	`Int` / `Decimal` / `String`	Une valeur `Decimal` (avec double précision), le premier argument renvoyé à la puissance du deuxième argument. Toutes les chaînes sont converties en`Decimals`. Si tout valeur `String` échoue à être convertie en `Decimal`, le résultat est `Undefined` .
Autre valeur	Autre valeur	`Undefined` .

round(Decimal)

Arrondit la valeur `Decimal` donnée à la valeur `Int` la plus proche. Si la valeur `Decimal` se situe à équidistance entre deux valeurs `Int` (par exemple, 0,5), la valeur `Decimal` est arrondie à la valeur supérieure.

Exemples :

`Round(1.2)` = 1

`Round(1.5)` = 2

`Round(1.7)` = 2

`Round(-1.1)` = -1

`Round(-1.5)` = -2

Type d'argument	Résultat
`Int`	L'argument
`Decimal`	La valeur `Decimal` est arrondie à la valeur `Int` inférieure la plus proche.
`String`	La valeur `Decimal` est arrondie à la valeur `Int` inférieure la plus proche. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
Autre valeur	`Undefined` .

sign(Decimal)

Renvoie le signe d'un chiffre donné. Lorsque le signe de l'argument est positif, la valeur 1 et renvoyée. Lorsque le signe de l'argument est négatif, la valeur -1 et renvoyée. Si l'argument est 0, la valeur 0 est renvoyée.

Exemples :

`sign(-7)` = -1

`sign(0)`= 0

`sign(13)` = 1

Type d'argument	Résultat
`Int`	`Int`, le signe de la valeur `Int`.
`Decimal`	`Int`, le signe de la valeur `Decimal`.

Type d'argument	Résultat
`String`	`Int`, le signe de la valeur `Decimal`. La chaîne est convertie en `Decimal` valeur et le signe de la `Decimal` valeur est renvoyé. Si la valeur `String` ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
Autre valeur	`Undefined` .

sin(Decimal)

Renvoie le sinus d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `sin(0)` = 0.0

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le sinus de l'argument.
`Decimal`	`Decimal` (avec double précision), le sinus de l'argument.
`Boolean`	`Undefined` .
`String`	`Decimal`, le sinus de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sinh(Decimal)

Renvoie le sinus hyperbolique d'un nombre. Les valeurs `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction. Le résultat est une valeur `Decimal` de double précision.

Exemples : `sinh(2.3)` = 4.936961805545957

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le sinus hyperbolique de l'argument.
`Decimal`	`Decimal` (avec double précision), le sinus hyperbolique de l'argument.
`Boolean`	`Undefined` .
`String`	`Decimal`, le sinus hyperbolique de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

sqrt(Decimal)

Renvoie la racine carrée d'un nombre en radians. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `sqrt(9)` = 3.0

Type d'argument	Résultat
`Int`	La racine carrée de l'argument.

Type d'argument	Résultat
`Decimal`	La racine carrée de l'argument.
`Boolean`	`Undefined` .
`String`	La racine carrée de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tan(Decimal)

Renvoie la tangente d'un nombre en radians. Les valeurs `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `tan(3)` = -0,1425465430742778

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), la tangente de l'argument.
`Decimal`	`Decimal` (avec double précision), la tangente de l'argument.
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), la tangente de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .

Type d'argument	Résultat
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

tanh(Decimal)

Renvoie la tangente hyperbolique d'un nombre en radians. Les valeurs `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction.

Exemples : `tanh(2.3)` = 0.9800963962661914

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), la tangente hyperbolique de l'argument.
`Decimal`	`Decimal` (avec double précision), la tangente hyperbolique de l'argument.
`Boolean`	`Undefined` .
`String`	`Decimal` (avec double précision), la tangente hyperbolique de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined` .
`Array`	`Undefined` .
`Object`	`Undefined` .
`Null`	`Undefined` .
`Undefined`	`Undefined` .

trunc (décimal, entier)

Tronque le premier argument du nombre de `Decimal`, spécifié par le deuxième argument. Si le deuxième argument est inférieur à zéro, il sera défini sur zéro. Si le deuxième argument est supérieur à 34, il sera défini sur 34. Les zéros suivants sont supprimés du résultat.

Exemples :

`trunc(2.3, 0)` = 2

`trunc(2.3123, 2)`= 2,31

`trunc(2.888, 2)`= 2,88

`trunc(2.00, 5)` = 2

Type d'argument 1	Type d'argument 2	Résultat
`Int`	`Int`	La valeur source.
`Int` / `Decimal` / `String`	`Int` / `Decimal`	Le premier argument est tronqué jusqu'à la longueur décrite par le deuxième argument. Le deuxième argument, s'il ne s'agit pas d'un `Int`, sera arrondi à la valeur `Int` inférieure la plus proche. Les chaînes sont converties en `Decimal` valeurs. Si la chaîne ne peut être pas convertie, le résultat est `Undefined` .
Autre valeur		Non défini.

RunPipelineActivity

Voici un exemple de la façon dont vous utiliseriez la `RunPipelineActivity` commande pour tester l'activité d'un pipeline. Dans cet exemple, nous testons une activité mathématique.

1. Créez un `maths.json` fichier contenant la définition de l'activité du pipeline que vous souhaitez tester.

```
{
    "math": {
        "name": "MyMathActivity",
        "math": "((temp - 32) * 5.0) / 9.0",
        "attribute": "tempC"
    }
}
```

2. Créez un `payloads.json` fichier contenant les exemples de charges utiles utilisées pour tester l'activité du pipeline.

```
[
    "{\"humidity\": 52, \"temp\": 68 }",
    "{\"humidity\": 52, \"temp\": 32 }"
]
```

3. Appelez l'`RunPipelineActivities`opération depuis la ligne de commande.

```
aws iotanalytics run-pipeline-activity  --pipeline-activity file://maths.json  --
payloads file://payloads.json --cli-binary-format raw-in-base64-out
```

Cela produit les résultats suivants.

```
{
    "logResult": "",
    "payloads": [
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjY4LCJ0ZW1wQyI6MjB9",
        "eyJodW1pZGl0eSI6NTIsInRlbXAiOjMyLCJ0ZW1wQyI6MH0="
    ]
}
```

Les charges utiles répertoriées dans les résultats sont des chaînes codées en Base64. Lorsque ces chaînes sont décodées, vous obtenez les résultats suivants.

```
{"humidity":52,"temp":68,"tempC":20}
{"humidity":52,"temp":32,"tempC":0}
```

Retraitement des messages du canal

AWS IoT Analytics vous permet de retraiter les données du canal. Cela peut être utile dans les cas suivants :

- Vous souhaitez relire des ingérées données existantes et non recommencer à zéro.
- Vous mettez à jour un pipeline et souhaitez intégrer les données existantes up-to-date avec les modifications.
- Vous souhaitez inclure les données qui ont été ingérées avant de modifier les options de stockage gérées par le client, les autorisations relatives aux canaux ou le magasin de données.

Paramètres

Lorsque vous retraitez des messages de canal via le pipeline avec AWS IoT Analytics, vous devez spécifier les informations suivantes :

`StartPipelineReprocessing`

Lance le retraitement des messages du canal via le pipeline.

`ChannelMessages`

Spécifie un ou plusieurs ensembles de messages de canal que vous souhaitez retraiter.

Si vous utilisez l'`channelMessages`objet, vous ne devez pas spécifier de valeur pour `startTime` et`endTime`.

`s3Paths`

Spécifie une ou plusieurs clés qui identifient les objets Amazon Simple Storage Service (Amazon S3) qui enregistrent les messages de votre chaîne. Vous devez utiliser le chemin complet de la clé.

Exemple de chemin : `00:00:00/1582940490000_1582940520000_123456789012_mychannel_0_2118.0.json`

Type : tableau de chaînes

Contraintes relatives aux membres du tableau : 1 à 100 éléments.

Contraintes de longueur : 1 à 1024 caractères.

`endTime`

Heure de fin (exclusive) des données de canal retraitées.

Si vous spécifiez une valeur pour le `endTime` paramètre, vous ne devez pas utiliser l'`channelMessages`objet.

Type : Timestamp

`startTime`

Heure de début (incluse) du retraitement des données de message brutes.

Si vous spécifiez une valeur pour le `startTime` paramètre, vous ne devez pas utiliser l'`channelMessages`objet.

Type : Timestamp

`pipelineName`

Nom du pipeline sur lequel démarrer le retraitement.

Type : String

Contraintes de longueur : 1 à 128 caractères.

Retraitement des messages du canal (console)

Ce didacticiel explique comment retraiter les données de canal stockées dans l'objet Amazon S3 spécifié dans la AWS IoT Analytics console.

Avant de commencer, assurez-vous que les messages du canal que vous souhaitez retraiter sont enregistrés dans un compartiment Amazon S3 géré par le client.

1. Connectez-vous à la console AWS IoT Analytics.
2. Dans le volet de navigation, sélectionnez Pipelines.
3. Choisissez votre pipeline cible.
4. Choisissez Retraiter les messages dans Actions.
5. Sur la page de retraitement du pipeline, choisissez des objets S3 pour les messages de retraitement.

La AWS IoT Analytics console propose également les options suivantes :

- Toute la plage disponible : retraitez toutes les données valides dans le canal.
- 120 derniers jours : retraitez les données arrivées au cours des 120 derniers jours.
- 90 derniers jours : retraitez les données arrivées au cours des 90 derniers jours.
- 30 derniers jours : retraitez les données arrivées au cours des 30 derniers jours.
- Plage personnalisée : retraitez les données arrivées dans la plage de temps spécifiée. Vous pouvez choisir n'importe quel intervalle de temps.

6. Entrez la clé de l'objet Amazon S3 qui stocke les messages de votre chaîne.

   Pour trouver la clé, procédez comme suit :

   a. Accédez à la console Amazon S3.
   b. Choisissez l'objet Amazon S3 cible.
   c. Sous Propriétés, dans la section Vue d'ensemble de l'objet, copiez la clé.

7. Choisissez Démarrer le retraitement.

Retraitement des messages du canal (API)

Lorsque vous utilisez l'`StartPipelineReprocessing`API, tenez compte des points suivants :

- Les `endTime` paramètres `startTime` et spécifient le moment où les données brutes ont été ingérées, mais il s'agit d'estimations approximatives. Vous pouvez les arrondir à l'heure la plus proche. `startTime`C'est inclusif, mais `endTime` c'est exclusif.
- La commande lance le retraitement de manière asynchrone et est renvoyée immédiatement.
- Il n'est pas garanti que les messages retraités soient traitées dans l'ordre dans lequel ils ont été initialement reçus. Il s'agit à peu près du même ordre, mais pas exactement.
- Vous pouvez effectuer jusqu'à 1 000 demandes d'`StartPipelineReprocessing`API toutes les 24 heures pour retraiter les messages du même canal via un pipeline.
- Le retraitement de vos données brutes entraîne des coûts supplémentaires.

Pour plus d'informations, consultez l'StartPipelineReprocessingAPI dans la section Référence des AWS IoT Analytics API.

Annulation des activités de retraitement des canaux

Pour annuler une activité de retraitement du pipeline, utilisez l'CancelPipelineReprocessingAPI ou choisissez Annuler le retraitement sur la page Activités de la AWS IoT Analytics console. Si vous annulez le retraitement, les données restantes ne seront pas traitées à nouveau. Vous devez lancer une autre demande de retraitement.

Utilisez l'DescribePipelineAPI pour vérifier l'état du retraitement. Voir le `reprocessingSummaries` champ dans la réponse.

Automatiser votre flux de travail

AWS IoT Analytics fournit une analyse avancée des données pour AWS IoT. Vous pouvez collecter automatiquement les données IoT, les traiter, les stocker et les analyser à l'aide des outils d'analyse et d'apprentissage machine. Vous pouvez exécuter des conteneurs hébergeant votre propre code analytique personnalisé ou Jupyter Notebook ou utiliser des conteneurs de code personnalisés tiers afin de ne pas avoir à recréer les outils d'analyse existants. Vous pouvez exploiter les fonctionnalités suivantes pour récupérer les données d'entrée d'un magasin de données et les intégrer dans un flux de travail automatisé :

Création du contenu d'un jeu de données selon un calendrier récurrent

Planifiez la création automatique du contenu du jeu de données en spécifiant un déclencheur lorsque vous appelez `CreateDataset` (`triggers:schedule:expression`). Les données présentes dans un magasin de données sont utilisées pour créer le contenu du jeu de données. Vous sélectionnez les champs souhaités à l'aide d'une requête SQL (`actions:queryAction:sqlQuery`).

Définissez un intervalle de temps contigu et sans chevauchement pour vous assurer que le nouveau contenu du jeu de données contient uniquement les données arrivées depuis la dernière fois. Utilisez les `:offsetSeconds` champs `actions:queryAction:filters:deltaTime` et pour spécifier l'intervalle de temps delta. Spécifiez ensuite un déclencheur pour créer le contenu de l'ensemble de données une fois l'intervalle de temps écoulé. Consultez the section called "Exemple 6 : création d'un jeu de données SQL avec une fenêtre delta (CLI)".

Création du contenu d'un jeu de données à la fin d'un autre jeu de données

Déclenchez la création d'un nouveau contenu de jeu de données lorsque la création du contenu d'un autre ensemble de données est terminée`triggers:dataset:name`.

Exécutez automatiquement vos applications d'analyse

Conteneurisez vos propres applications d'analyse de données personnalisées et déclenchez-les pour qu'elles s'exécutent lorsque le contenu d'un autre ensemble de données est créé. Ainsi, vous pouvez alimenter votre application avec des données issues du contenu d'un ensemble de données créé selon un calendrier récurrent. Vous pouvez agir automatiquement sur les résultats de votre analyse depuis votre application. (`actions:containerAction`)

Création du contenu d'un jeu de données à la fin d'un autre jeu de données

Déclenchez la création d'un nouveau contenu de jeu de données lorsque la création du contenu d'un autre ensemble de données est terminée`triggers:dataset:name`.

Exécutez automatiquement vos applications d'analyse

Conteneurisez vos propres applications d'analyse de données personnalisées et déclenchez-les pour qu'elles s'exécutent lorsque le contenu d'un autre ensemble de données est créé. Ainsi, vous pouvez alimenter votre application avec des données issues du contenu d'un ensemble de données créé selon un calendrier récurrent. Vous pouvez agir automatiquement sur les résultats de votre analyse depuis votre application. (`actions:containerAction`)

Cas d'utilisation

Automatisez la mesure de la qualité des produits pour réduire OpEx

Un système avec une valve intelligente permet de mesurer la température, l'humidité et la pression. Le système rassemble les événements périodiquement et également lorsque certains événements se produisent, par exemple lorsqu'une valeur s'ouvre et se ferme. Vous pouvez automatiser une analyse qui agrège les données non superposées provenant de ces fenêtres périodiques et crée des rapports KPI sur la qualité du produit final. AWS IoT Analytics Après avoir traité chaque lot, vous mesurez la qualité globale du produit et réduisez vos dépenses opérationnelles en maximisant le volume de production.

Automatisation de l'analyse d'une flotte d'appareils

Vous exécutez des analyses (algorithme, science des données ou ML pour KPI) toutes les 15 minutes sur des données générées par des centaines d'appareils. Chaque cycle d'analyse génère et stocke l'état pour la prochaine exécution d'analyse. Pour chacune de vos analyses, vous souhaitez utiliser uniquement les données reçues au sein d'une période de temps spécifique. AWS IoT Analytics Vous pouvez ainsi orchestrer vos analyses et créer le KPI et le rapport pour chaque exécution, puis stocker les données pour les analyses futures.

Automatisation de la détection des anomalies

AWS IoT Analytics vous permet d'automatiser votre flux de travail de détection des anomalies que vous devez exécuter manuellement toutes les 15 minutes sur les nouvelles données arrivées dans un magasin de données. Vous pouvez également automatiser la création d'un tableau de bord qui indique l'utilisation des appareils et les utilisateurs les plus actifs au cours d'un laps de temps spécifié.

Prévision des résultats des processus industriels

Vous avez des chaînes de production industrielles. À l'aide des données envoyées à AWS IoT Analytics, y compris les mesures de processus disponibles, vous pouvez rendre opérationnels les flux de travail analytiques afin de prévoir les résultats des processus. Les données du modèle peuvent être organisées dans une matrice M x N où chaque ligne contient des données provenant de différents moments où des échantillons de laboratoire sont prélevés. AWS IoT Analytics vous aide à opérationnaliser votre flux de travail analytique en créant des fenêtres delta et en utilisant vos outils de science des données pour créer KPIs et enregistrer l'état des appareils de mesure.

Utilisation d'un conteneur Docker

Cette section contient des informations sur la création de votre propre conteneur Docker. Un risque de sécurité existe si vous réutilisez des conteneurs Docker créés par des tiers : ces conteneurs peuvent exécuter du code arbitraire avec vos autorisations utilisateur. Assurez-vous que vous faites confiance à l'auteur de tout conteneur tiers avant de l'utiliser.

Voici les étapes à suivre pour configurer l'analyse périodique des données parvenues depuis la dernière analyse effectuée :

1. Créez un conteneur Docker qui contient vos données d'application et toutes les bibliothèques requises ou autres dépendances.

   L'extension IotAnalytics Jupyter fournit une API de conteneurisation pour faciliter le processus de conteneurisation. Vous pouvez également exécuter des images de votre propre création dans lesquelles vous créez ou assemblez le jeu d'outils de votre application pour effectuer l'analyse de données ou le calcul souhaités. AWS IoT Analytics vous permet de définir la source des données d'entrée de l'application conteneurisée et la destination des données de sortie du conteneur Docker au moyen de variables. (Les Input/Output variables de conteneur Docker personnalisées contiennent plus d'informations sur l'utilisation de variables avec un conteneur personnalisé.)
2. Chargez le conteneur dans un registre Amazon ECR.
3. Créez un magasin de données pour recevoir et stocker des messages (données) provenant d'appareils (`iotanalytics: CreateDatastore`)
4. Créez un canal où les messages sont envoyés (`iotanalytics: CreateChannel`).
5. Créez un pipeline pour connecter le canal au magasin de données (`iotanalytics: CreatePipeline`).

6. Créez un rôle IAM qui autorise l'envoi de données de message à un AWS IoT Analytics canal () `iam: CreateRole`.
7. Créez une règle IoT qui utilise une requête SQL pour connecter un canal à la source des données du message (`iot: CreateTopicRule`champ`topicRulePayload:actions:iotAnalytics`). Lorsqu'un appareil envoie un message avec le sujet approprié via MQTT, il est acheminé vers votre chaîne. Vous pouvez également utiliser `iotanalytics: BatchPutMessage` pour envoyer des messages directement dans un canal à partir d'un appareil capable d'utiliser le AWS SDK ou AWS CLI.
8. Créez un jeu de données SQL dont la création est déclenchée par un calendrier (`iotanalytics: CreateDataset,`champ`actions: queryAction:sqlQuery`).

   Vous spécifiez également un filtre préalable à appliquer aux données de message pour contribuer à limiter les messages à ceux qui sont arrivés depuis la dernière exécution de l'action. (Le champ `actions:queryAction:filters:deltaTime:timeExpression` donne une expression permettant de déterminer l'heure d'un message, tandis que le champ `actions:queryAction:filters:deltaTime:offsetSeconds` indique la latence possible lors de l'arrivée d'un message.)

   Le préfiltre, ainsi que le calendrier de déclenchement, déterminent votre fenêtre delta. Chaque nouveau jeu de données SQL est créé à partir des messages reçus depuis la dernière création du jeu de données SQL. (Qu'en est-il de la première fois que le jeu de données SQL est créé ? Une estimation de la date à laquelle le jeu de données aurait été créé pour la dernière fois est établie sur la base du calendrier et du préfiltre.)
9. Créez un autre jeu de données déclenché par la création du premier (CreateDatasetchamp`trigger:dataset`). Pour cet ensemble de données, vous spécifiez une action de conteneur (`actions:containerAction`file) qui pointe vers le conteneur Docker que vous avez créé lors de la première étape et fournit les informations nécessaires à son exécution. Voici également les éléments que vous devez spécifier :

   - L'ARN du conteneur docker stocké dans votre compte (`image`.)
   - ARN du rôle qui autorise le système à accéder aux ressources nécessaires pour exécuter l'action de conteneur (`executionRoleArn`).
   - Configuration de la ressource qui exécute l'action du conteneur (`resourceConfiguration`.)

- Type de ressource de calcul utilisée pour exécuter l'action du conteneur (`computeType`avec les valeurs possibles :`ACU_1 [vCPU=4, memory=16GiB] or ACU_2 [vCPU=8, memory=32GiB]`).
- Taille (Go) du stockage persistant disponible pour l'instance de ressource utilisée pour exécuter l'action du conteneur (`volumeSizeInGB`).
- Les valeurs des variables utilisées dans le contexte de l'exécution de l'application (essentiellement, les paramètres transmis à l'application) (`variables`).

  Ces variables sont remplacées au moment où un conteneur est exécuté. Cela vous permet d'exécuter le même conteneur avec différentes variables (paramètres) fournies au moment de la création du contenu du jeu de données. L'extension IotAnalytics Jupyter simplifie ce processus en reconnaissant automatiquement les variables d'un bloc-notes et en les rendant disponibles dans le cadre du processus de conteneurisation. Vous pouvez choisir les variables reconnues ou ajouter vos propres variables personnalisées. Avant d'exécuter un conteneur, le système remplace chacune de ces variables par la valeur active au moment de l'exécution.
- L'une des variables est le nom du jeu de données dont le dernier contenu est utilisé comme entrée dans l'application (il s'agit du nom du jeu de données que vous avez créé à l'étape précédente) (`datasetContentVersionValue:datasetName`).

À l'aide de la requête SQL et de la fenêtre delta pour générer le jeu de données, et du conteneur associé à votre application, vous AWS IoT Analytics créez un ensemble de données de production planifié qui s'exécute à l'intervalle que vous spécifiez sur les données de la fenêtre delta, produisant le résultat souhaité et envoyant des notifications.

Vous pouvez suspendre votre application de jeu de données de production et la reprendre à tout moment. Lorsque vous reprenez votre application de jeu de données de production, elle récupère par défaut toutes les données arrivées depuis la dernière exécution, mais qui n'ont pas encore été analysées. AWS IoT Analytics Vous pouvez également configurer la manière dont vous souhaitez reprendre votre jeu de données de production (durée de la fenêtre de travail) en effectuant une série d'exécutions consécutives. Vous pouvez également reprendre votre application de jeu de données de production en capturant uniquement les données nouvellement arrivées qui correspondent à la taille spécifiée de votre fenêtre delta.

Veuillez noter les limites suivantes lors de la création ou de la définition d'un ensemble de données déclenché par la création d'un autre ensemble de données :

- Seuls les ensembles de données de conteneurs peuvent être déclenchés par des ensembles de données SQL.
- Un jeu de données SQL peut déclencher au maximum 10 ensembles de données de conteneurs.

Les erreurs suivantes peuvent être renvoyées lors de la création d'un ensemble de données conteneur déclenché par un ensemble de données SQL :

- « Triggering dataset can only be added on a container dataset »
- « There can only be one triggering dataset »

  Cette erreur se produit si vous tentez de définir un ensemble de données conteneur déclenché par deux ensembles de données SQL différents.
- « L'ensemble de données déclencheur <dataset-name>ne peut pas être déclenché par un ensemble de données conteneur »

  Cette erreur se produit si vous tentez de définir un autre jeu de données conteneur déclenché par un autre jeu de données conteneur.
- « <N>les ensembles de données dépendent déjà de l'<dataset-name>ensemble de données. »

  Cette erreur se produit si vous tentez de définir un autre jeu de données conteneur déclenché par un ensemble de données SQL qui déclenche déjà 10 ensembles de données de conteneurs.
- « Exactly one trigger type should be provided »

  Cette erreur se produit lorsque vous tentez de définir un ensemble de données qui est déclenché à la fois par un déclencheur de planification et par un déclencheur de jeu de données.

Variables de conteneur input/output Docker personnalisées

Cette section illustre la façon dont le programme qui est exécuté par votre image Docker personnalisée peut lire les variables d'entrée et charger sa sortie.

Fichier de paramètres

Les variables d'entrée et les destinations dans lesquelles vous souhaitez charger la sortie sont stockées dans un fichier JSON situé sous `/opt/ml/input/data/iotanalytics/params` sur l'instance qui exécute l'image Docker. Voici un exemple du contenu de ce fichier.

```
{
```

```
   "Context": {
       "OutputUris": {
           "html": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.html",
           "ipynb": "s3://aws-iot-analytics-dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.ipynb"
       }
   },
   "Variables": {
       "source_dataset_name": "mydataset",
       "source_dataset_version_id": "xxxx",
       "example_var": "hello world!",
       "custom_output": "s3://aws-iot-analytics/dataset-xxxxxxx/notebook/results/
iotanalytics-xxxxxxx/output.txt"
   }
}
```

Outre le nom et l'ID de version de votre ensemble de données, la section `Variables` contient les variables spécifiées dans l'appel `iotanalytics:CreateDataset`. Dans cet exemple, une variable `example_var` a la valeur `hello world!`. Un URI de sortie personnalisé a également été fourni dans la variable `custom_output`. Le `OutputUris` champ contient les emplacements par défaut vers lesquels le conteneur peut télécharger sa sortie. Dans cet exemple, une sortie par défaut a URIs été fournie pour les sorties ipynb et html.

Variables d'entrée

Le programme lancée par votre image Docker peut lire des variables dans le fichier `params`. Voici un exemple de programme qui ouvre le `params` fichier, l'analyse et imprime la valeur de la `example_var` variable.

```
import json

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]
print(example_var)
```

Téléchargement de la sortie

Le programme lancé par votre image Docker peut également stocker sa sortie dans un emplacement Amazon S3. La sortie doit être chargée avec une liste de contrôle d'accès bucket-owner-full-control

« ». La liste d'accès permet au AWS IoT Analytics service de contrôler la sortie téléchargée. Dans cet exemple, nous étendons le précédent pour télécharger le contenu de `example_var` vers l'emplacement Amazon S3 défini par `custom_output` dans le `params` fichier.

```
import boto3
import json
from urllib.parse import urlparse

ACCESS_CONTROL_LIST = "bucket-owner-full-control"

with open("/opt/ml/input/data/iotanalytics/params") as param_file:
    params = json.loads(param_file.read())
example_var = params["Variables"]["example_var"]

outputUri = params["Variables"]["custom_output"]
# break the S3 path into a bucket and key
bucket = urlparse(outputUri).netloc
key = urlparse(outputUri).path.lstrip("/")

s3_client = boto3.client("s3")
s3_client.put_object(Bucket=bucket, Key=key, Body=example_var, ACL=ACCESS_CONTROL_LIST)
```

Permissions

Vous devez créer deux rôles . Un rôle autorise le lancement d'une instance d' SageMaker IA afin de conteneuriser un bloc-notes. Un autre rôle est nécessaire pour exécuter un conteneur.

Le rôle peut être créé automatiquement par ou manuellement. Si vous créez votre nouvelle instance d' SageMaker IA avec la AWS IoT Analytics console, vous avez la possibilité de créer automatiquement un nouveau rôle qui accorde tous les privilèges nécessaires pour exécuter des instances d' SageMaker IA et conteneuriser des blocs-notes. Vous pouvez également créer manuellement un rôle avec ces privilèges. Pour ce faire, créez un rôle auquel est attachée la `AmazonSageMakerFullAccess` politique et ajoutez la stratégie suivante.

JSON

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

```
      "Effect": "Allow",
      "Action": [
        "ecr:BatchDeleteImage",
        "ecr:BatchGetImage",
        "ecr:CompleteLayerUpload",
        "ecr:CreateRepository",
        "ecr:DescribeRepositories",
        "ecr:GetAuthorizationToken",
        "ecr:InitiateLayerUpload",
        "ecr:PutImage",
        "ecr:UploadLayerPart"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": "arn:aws:s3:*:*:iotanalytics-notebook-containers/*"
    }
  ]
}
```

Vous devez créer manuellement le second rôle qui accorde l'autorisation d'exécuter un conteneur. Vous devez le faire même si vous avez utilisé la AWS IoT Analytics console pour créer automatiquement le premier rôle. Créez un rôle associé à la politique et à la politique de confiance suivantes.

JSON

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

```
            ],
            "Resource": "arn:aws:s3:*:*:aws-*-dataset-*/*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotanalytics:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

Voici un exemple de politique de confiance.

JSON

```
{
```

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

Utilisation de CreateDataset l'API via Java et AWS CLI

Crée un jeu de données. Un ensemble de données stocke les données extraites d'un magasin de données en appliquant une `queryAction` (requête SQL) ou a `containerAction` (exécution d'une application conteneurisée). Cette opération crée le squelette d'un ensemble de données. Le jeu de données peut être rempli manuellement par appel `CreateDatasetContent` ou automatiquement selon un paramètre `trigger` que vous spécifiez. Pour plus d'informations, consultez CreateDataset et `CreateDatasetContent`.

Rubriques

- Exemple 1 : création d'un jeu de données SQL (Java)
- Exemple 2 : création d'un jeu de données SQL avec une fenêtre delta (Java)
- Exemple 3 : création d'un jeu de données conteneur avec son propre déclencheur de planification (Java)
- Exemple 4 : création d'un ensemble de données conteneur avec un ensemble de données SQL comme déclencheur (Java)
- Exemple 5 : création d'un jeu de données SQL (CLI)
- Exemple 6 : création d'un jeu de données SQL avec une fenêtre delta (CLI)

Exemple 1 : création d'un jeu de données SQL (Java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();
```

```
//Create Action
action.setActionName("SQLAction1");
action.setQueryAction(new SqlQueryDatasetAction().withSqlQuery("select * from
 DataStoreName"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Sortie en cas de réussite :

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemple 2 : création d'un jeu de données SQL avec une fenêtre delta (Java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Filter for DeltaTime
QueryFilter deltaTimeFilter = new QueryFilter();
deltaTimeFilter.withDeltaTime(
                new DeltaTime()
```

```
                .withOffsetSeconds(-1 * EstimatedDataDelayInSeconds)
                .withTimeExpression("from_unixtime(timestamp)"));

//Create Action
action.setActionName("SQLActionWithDeltaTime");
action.setQueryAction(new SqlQueryDatasetAction()
                .withSqlQuery("SELECT * from DataStoreName")
                .withFilters(deltaTimeFilter));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Sortie en cas de réussite :

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemple 3 : création d'un jeu de données conteneur avec son propre déclencheur de planification (Java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
DatasetAction action = new DatasetAction();

//Create Action
```

```
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger();
trigger.setSchedule(new Schedule().withExpression("cron(0 12 * * ? *)"));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);

// Add RetentionPeriod to CreateDatasetRequest object
request.setRetentionPeriod(new RetentionPeriod().withNumberOfDays(10));
final CreateDatasetResult result = iot.createDataset(request);
```

Sortie en cas de réussite :

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>, RetentionPeriod: {unlimited:
 true} or {numberOfDays: 10, unlimited: false}}
```

Exemple 4 : création d'un ensemble de données conteneur avec un ensemble de données SQL comme déclencheur (Java)

```
CreateDatasetRequest request = new CreateDatasetRequest();
request.setDatasetName(dataSetName);
```

```
DatasetAction action = new DatasetAction();

//Create Action
action.setActionName("ContainerActionDataset");
action.setContainerAction(new ContainerDatasetAction()
        .withImage(ImageURI)
        .withExecutionRoleArn(ExecutionRoleArn)
        .withResourceConfiguration(
                new ResourceConfiguration()
                .withComputeType(new ComputeType().withAcu(1))
                .withVolumeSizeInGB(1))
        .withVariables(new Variable()
        .withName("VariableName")
        .withStringValue("VariableValue"));

// Add Action to Actions List
List<DatasetAction> actions = new ArrayList<DatasetAction>();
actions.add(action);

//Create Trigger
DatasetTrigger trigger = new DatasetTrigger()
        .withDataset(new TriggeringDataset()
                .withName(TriggeringSQLDataSetName));

//Add Trigger to Triggers List
List<DatasetTrigger> triggers = new ArrayList<DatasetTrigger>();
triggers.add(trigger);

// Add Triggers and Actions to CreateDatasetRequest object
request.setActions(actions);
request.setTriggers(triggers);
final CreateDatasetResult result = iot.createDataset(request);
```

Sortie en cas de réussite :

```
{DatasetName: <datatsetName>, DatasetArn: <datatsetARN>}
```

Exemple 5 : création d'un jeu de données SQL (CLI)

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --dataset-
name="<dataSetName>" --actions="[{\"actionName\":\"<ActionName>\", \"queryAction\":
{\"sqlQuery\":\"<SQLQuery>\"}}]" --retentionPeriod numberOfDays=10
```

Sortie en cas de réussite :

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
    "retentionPeriod": {unlimited: true} or {numberOfDays: 10, unlimited: false}
}
```

Exemple 6 : création d'un jeu de données SQL avec une fenêtre delta (CLI)

Les fenêtres delta sont une série d'intervalles de temps continus définis par l'utilisateur et ne se chevauchant pas. Les fenêtres Delta vous permettent de créer le contenu d'un ensemble de données à partir de nouvelles données arrivées dans le magasin de données depuis la dernière analyse et d'effectuer une analyse sur celles-ci. Vous créez une fenêtre delta `deltaTime` en définissant la `filters` partie d'un ensemble `queryAction` de données (CreateDataset). En général, vous souhaiterez créer le contenu de l'ensemble de données automatiquement en configurant également un déclencheur d'intervalle de temps (`triggers:schedule:expression`). En gros, cela vous permet de filtrer les messages arrivés pendant une période donnée, afin que les données contenues dans les messages des fenêtres temporelles précédentes ne soient pas comptées deux fois.

Dans cet exemple, nous créons un nouvel ensemble de données qui crée automatiquement le nouveau contenu de l'ensemble de données toutes les 15 minutes en utilisant uniquement les données arrivées depuis la dernière fois. Nous indiquons un décalage `deltaTime` de 3 minutes (180 secondes) qui permet un délai de 3 minutes pour que les messages arrivent dans le magasin de données spécifié. Ainsi, si le contenu du jeu de données est créé à 10 h 30, les données utilisées (incluses dans le contenu du jeu de données) seront celles dont l'heure est comprise entre 10 h 12 et 10 h 27 (soit 10 h 30 - 15 minutes - 3 minutes et 10 h 30 - 3 minutes).

```
aws iotanalytics --endpoint <EndPoint>  --region <Region> create-dataset --cli-input-
json file://delta-window.json
```

Où le fichier `delta-window.json` contient les éléments suivants.

```
{
  "datasetName": "delta_window_example",
  "actions": [
    {
      "actionName": "delta_window_action",
      "queryAction": {
```

```
        "sqlQuery": "SELECT temperature, humidity, timestamp FROM my_datastore",
        "filters": [
          {
            "deltaTime": {
              "offsetSeconds": -180,
              "timeExpression": "from_unixtime(timestamp)"
            }
          }
        ]
      }
    }
  ],
  "triggers": [
    {
      "schedule": {
        "expression": "cron(0/15 * * * ? *)"
      }
    }
  ]
}
```

Sortie en cas de réussite :

```
{
    "datasetName": "<datasetName>",
    "datasetArn": "<datatsetARN>",
}
```

Conteneurisation d'un bloc-notes

Cette section contient des informations sur la création d'un conteneur Docker à l'aide d'un bloc-notes Jupyter. Un risque de sécurité existe si vous réutilisez des blocs-notes créés par des tiers : ces conteneurs peuvent exécuter du code arbitraire avec vos autorisations utilisateur. En outre, le code HTML généré par le bloc-notes peut être affiché dans la AWS IoT Analytics console, fournissant ainsi un vecteur d'attaque potentiel sur l'ordinateur qui affiche le code HTML. Assurez-vous que vous faites confiance à l'auteur de tout bloc-notes tiers avant de l'utiliser.

L'une des options avancées permettant d'effectuer des fonctions analytiques consiste à utiliser un bloc-notes Jupyter. Jupyter Notebook fournit de puissants outils de science des données capables d'effectuer de l'apprentissage automatique et de nombreuses analyses statistiques. Pour plus

d'informations, consultez la section Modèles de bloc-notes. (Notez que nous ne prenons actuellement pas en charge la conteneurisation interne JupyterLab.) Vous pouvez empaqueter votre bloc-notes Jupyter et vos bibliothèques dans un conteneur qui s'exécute périodiquement sur un nouveau lot de données au fur et à mesure de leur réception AWS IoT Analytics pendant une fenêtre de temps delta que vous définissez. Vous pouvez planifier une tâche d'analyse qui utilise le conteneur et les nouvelles données segmentées capturées dans le délai spécifié, puis qui stocke le résultat de la tâche pour de futures analyses planifiées.

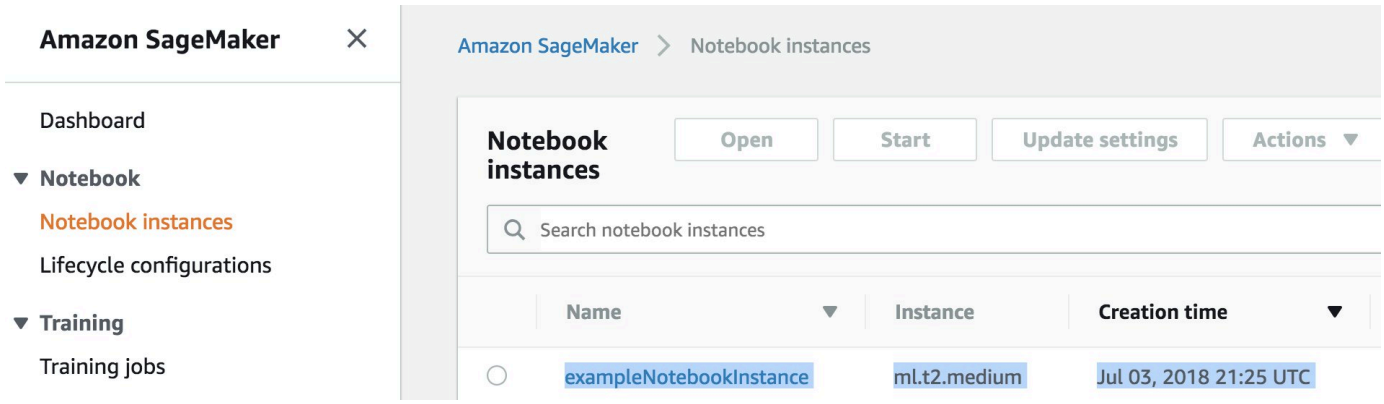
Si vous avez créé une instance SageMaker AI à l'aide de la AWS IoT Analytics console après le 23 août 2018, l'installation de l'extension de conteneurisation a été effectuée automatiquement et vous pouvez commencer à créer une image conteneurisée. Sinon, suivez les étapes répertoriées dans cette section pour activer la conteneurisation des blocs-notes sur votre instance SageMaker AI. Dans ce qui suit, vous modifiez votre rôle d'exécution SageMaker AI pour vous permettre de télécharger l'image du conteneur sur Amazon EC2 et vous installez l'extension de conteneurisation.

Activer la conteneurisation des instances de bloc-notes non créées via la console AWS IoT Analytics

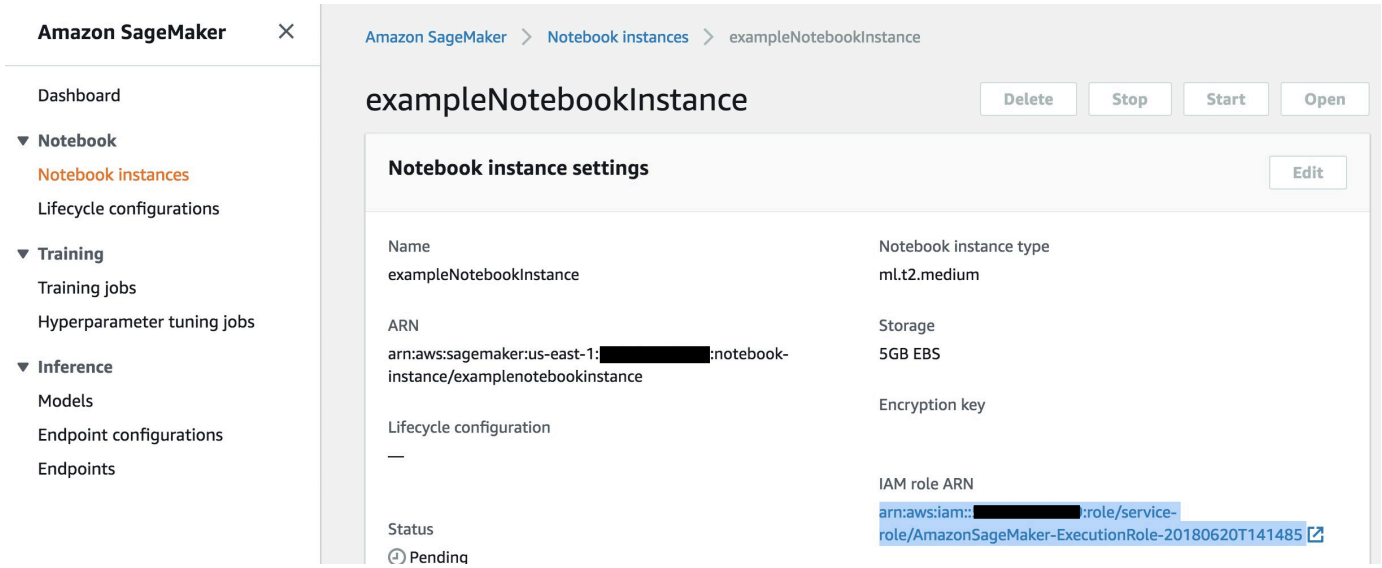
Nous vous recommandons de créer une nouvelle instance d' SageMaker IA via la AWS IoT Analytics console au lieu de suivre ces étapes. Les nouvelles instances prennent automatiquement en charge la conteneurisation.

Si vous redémarrez votre instance SageMaker AI après avoir activé la conteneurisation comme indiqué ici, vous n'aurez pas à ajouter à nouveau les rôles et politiques IAM, mais vous devrez réinstaller l'extension, comme indiqué à l'étape finale.

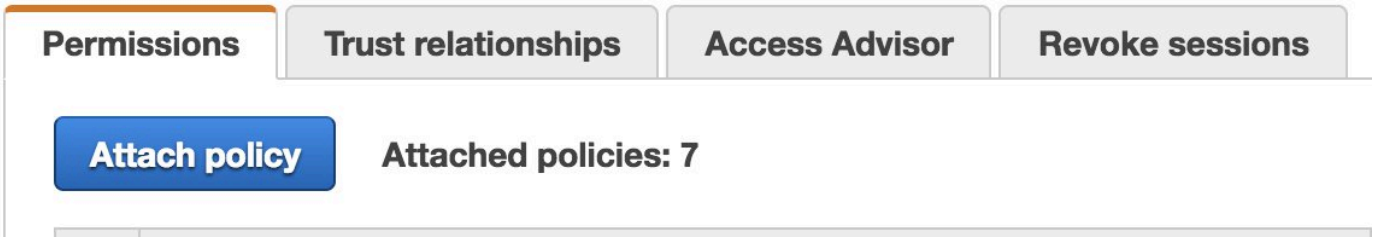
1. Pour accorder à votre instance de bloc-notes l'accès à Amazon ECS, sélectionnez votre instance SageMaker AI sur la page SageMaker AI :



2. Sous ARN du rôle IAM, choisissez le rôle d'exécution SageMaker AI.

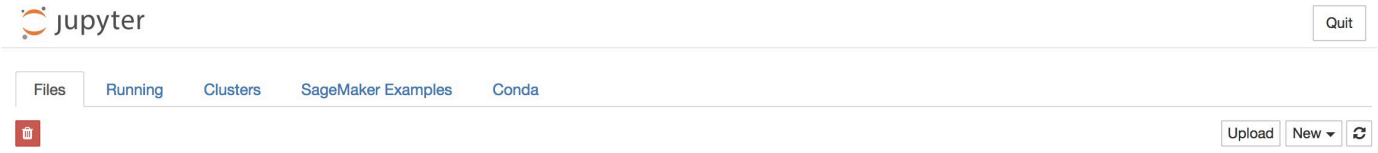


3. Choisissez Attach Policy, puis définissez et attachez la stratégie illustrée dans Permissions. Si la `AmazonSageMakerFullAccess` politique n'est pas déjà jointe, joignez-la également.



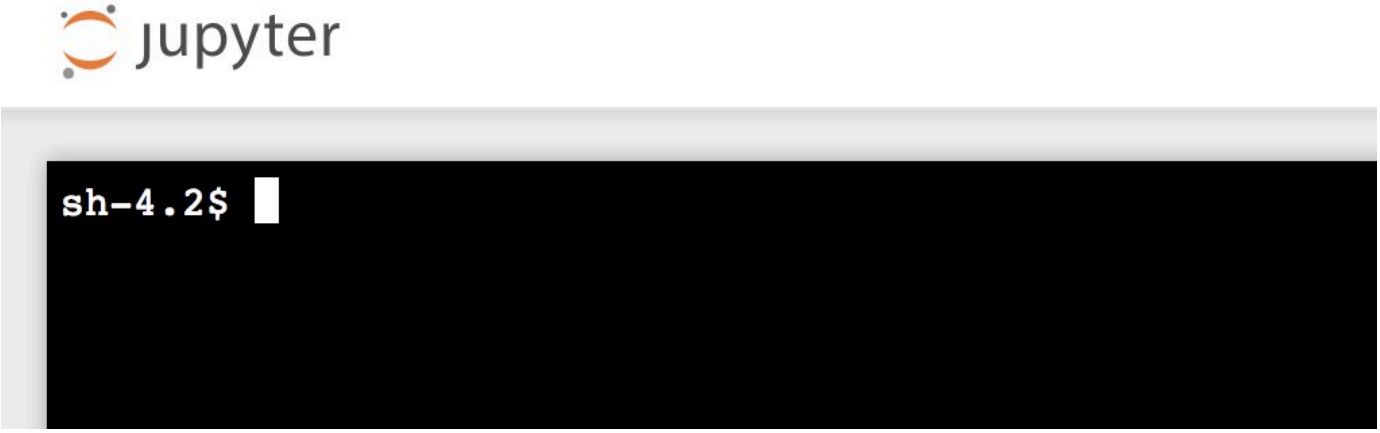
Vous devez également télécharger le code de conteneurisation depuis Amazon S3 et l'installer sur votre instance de bloc-notes. La première étape consiste à accéder au terminal de l'instance SageMaker AI.

1. Dans Jupyter, choisissez Nouveau.



2. Dans le menu qui apparaît, choisissez Terminal.

3. Dans le terminal, saisissez les commandes suivantes pour télécharger le code, décompressez-le et installez-le. Notez que ces commandes tuent tous les processus exécutés par vos blocs-notes sur cette instance d' SageMaker IA.




```
cd /tmp

aws s3 cp s3://iotanalytics-notebook-containers/iota_notebook_containers.zip /tmp

unzip iota_notebook_containers.zip

cd iota_notebook_containers

chmod u+x install.sh

./install.sh
```

Patientez pendant une minute ou deux que l'extension soit validée et installée.

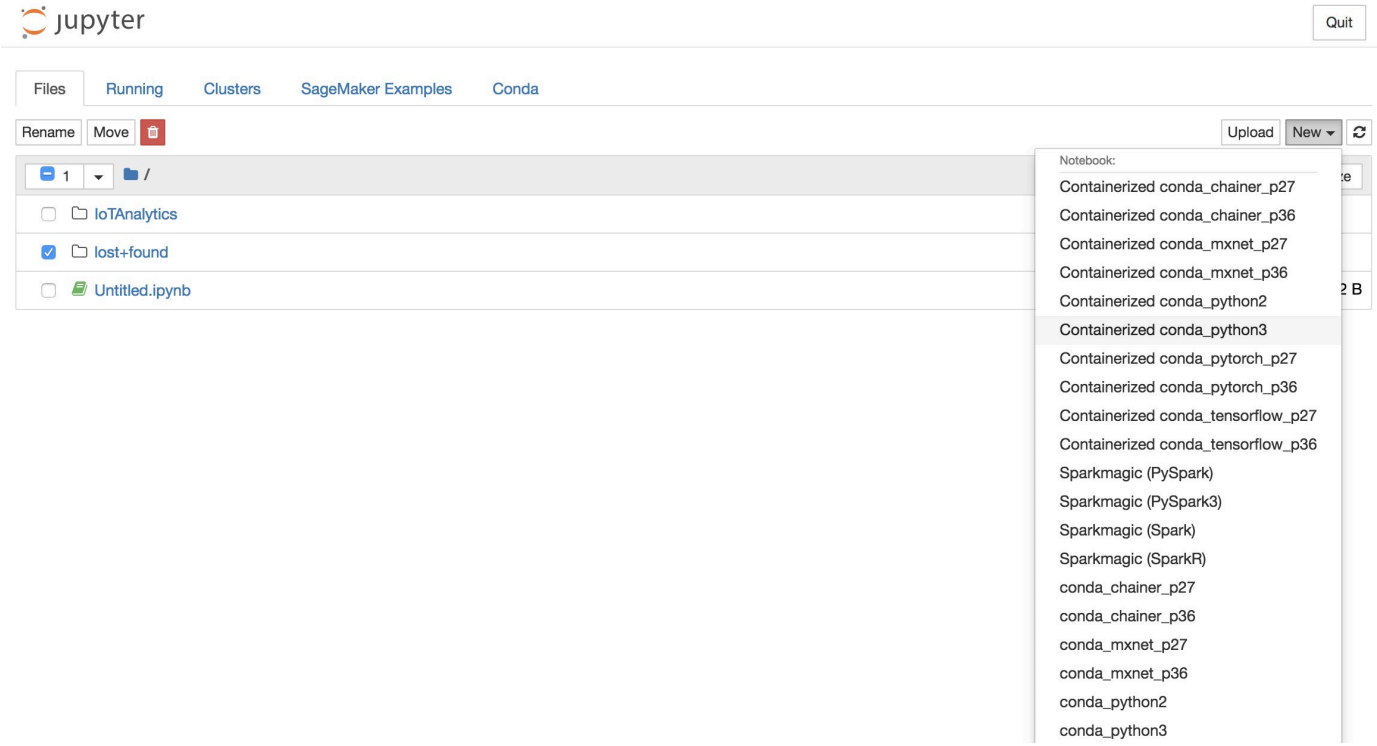
Mettez à jour l'extension de conteneurisation de votre bloc-notes

Si vous avez créé votre instance SageMaker AI via la AWS IoT Analytics console après le 23 août 2018, l'extension de conteneurisation a été installée automatiquement. Vous pouvez mettre à jour l'extension en redémarrant votre instance depuis SageMaker AI Console. Si vous avez installé l'extension manuellement, vous pouvez la mettre à jour en réexécutant les commandes de terminal répertoriées dans Activer la conteneurisation des instances de bloc-notes non créées via la console. AWS IoT Analytics

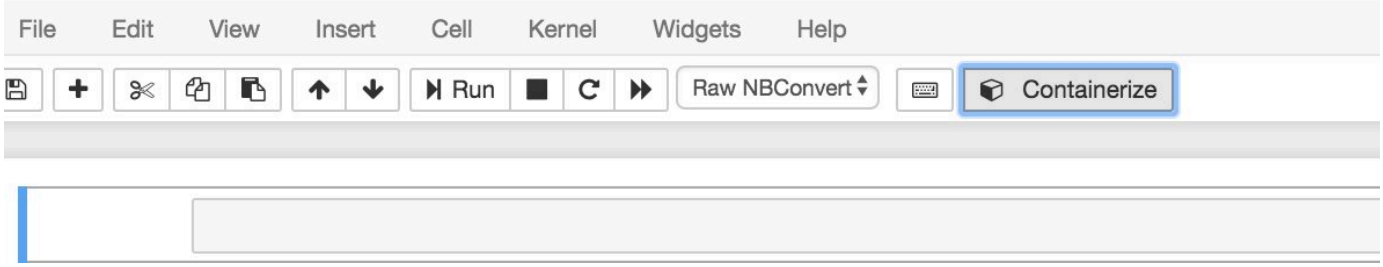
Création d'une image conteneurisée

Dans cette section, nous présentons les étapes nécessaires pour conteneuriser un bloc-notes. Pour commencer, rendez-vous sur Jupyter Notebook pour créer un bloc-notes avec un kernel conteneurisé.

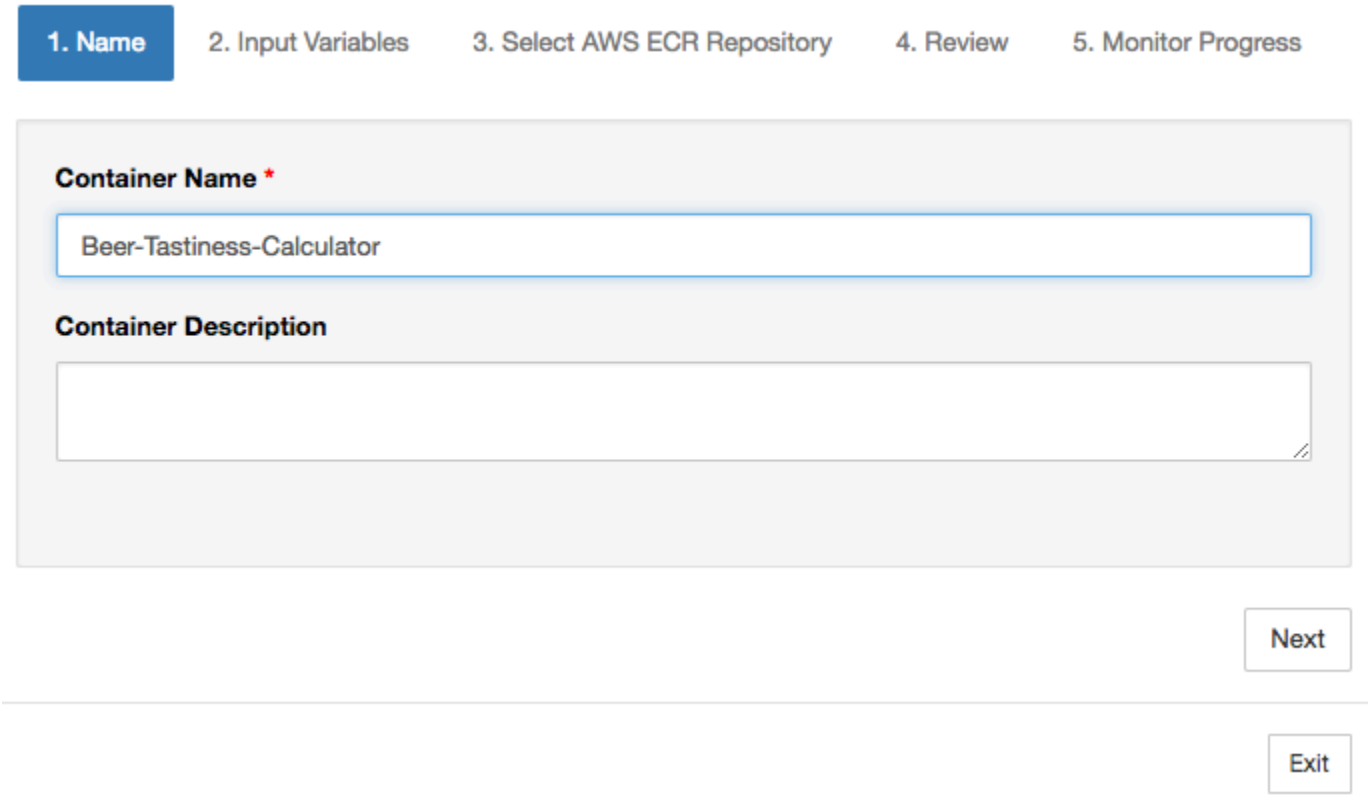
1. Dans votre instance Jupyter Notebook, choisissez New, puis choisissez le type de kernel souhaité dans la liste déroulante. (Le type de noyau doit commencer par « conteneurisé » et se terminer par le noyau que vous auriez autrement sélectionné. Par exemple, si vous voulez simplement un environnement Python 3.0 simple tel que « conda_python3 », choisissez « Containerized conda_python3 »).



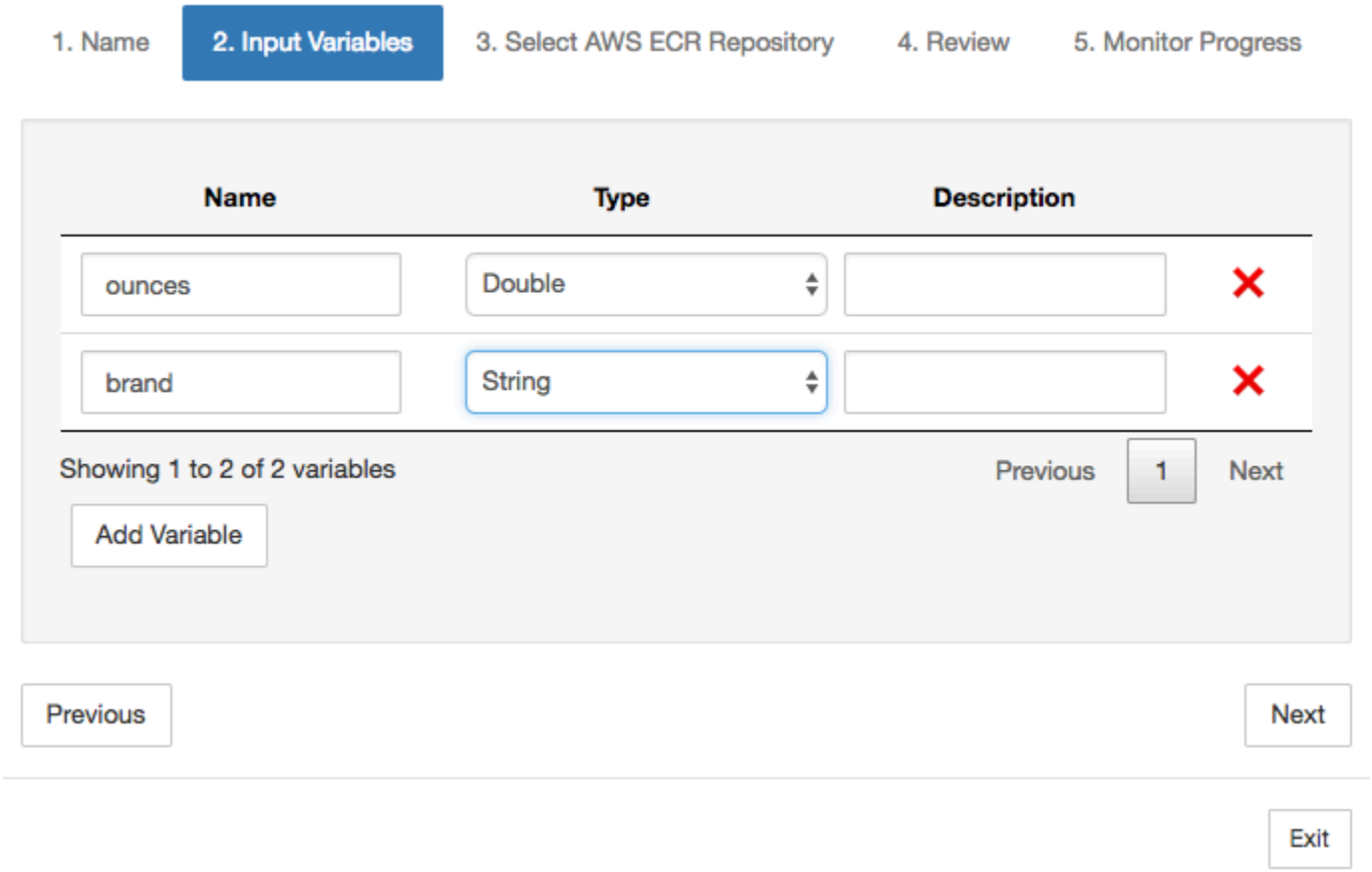
2. Une fois que vous avez terminé de travailler sur votre bloc-notes et que vous souhaitez le conteneuriser, choisissez Containerize.



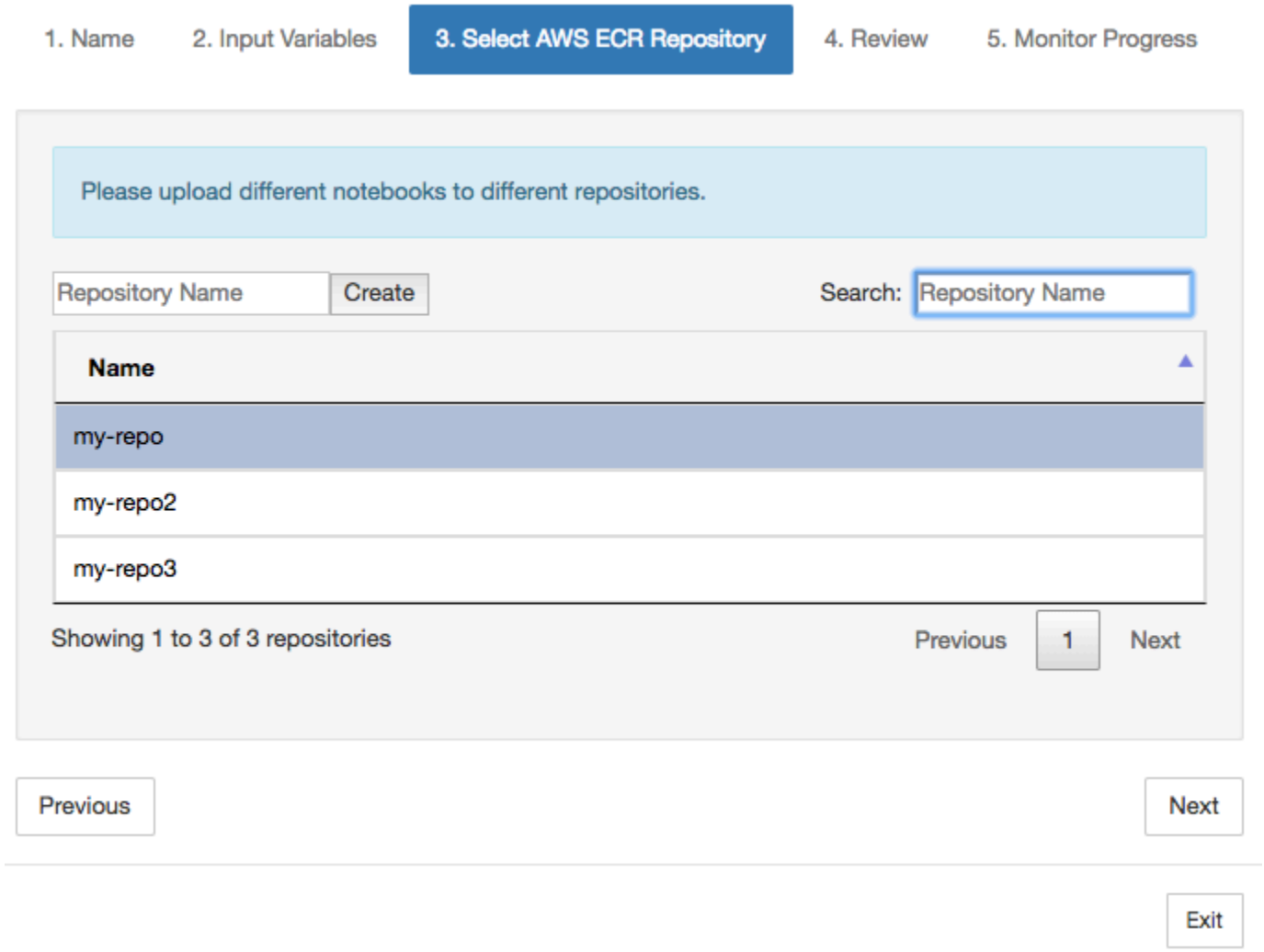
3. Attribuez un nom au bloc-notes conteneurisé. Vous pouvez également saisir une description facultative.



4. Spécifiez les variables d'entrée (paramètres) avec lesquels votre bloc-notes doit être appelé. Vous pouvez sélectionner les variables d'entrée qui sont automatiquement détectées à partir du bloc-notes ou définir des variables personnalisées. (Notez que les variables d'entrée sont uniquement détectées si vous avez déjà exécuté votre bloc-notes.) Pour chaque variable d'entrée, choisissez un type. Vous pouvez également saisir une description facultative de la variable d'entrée.

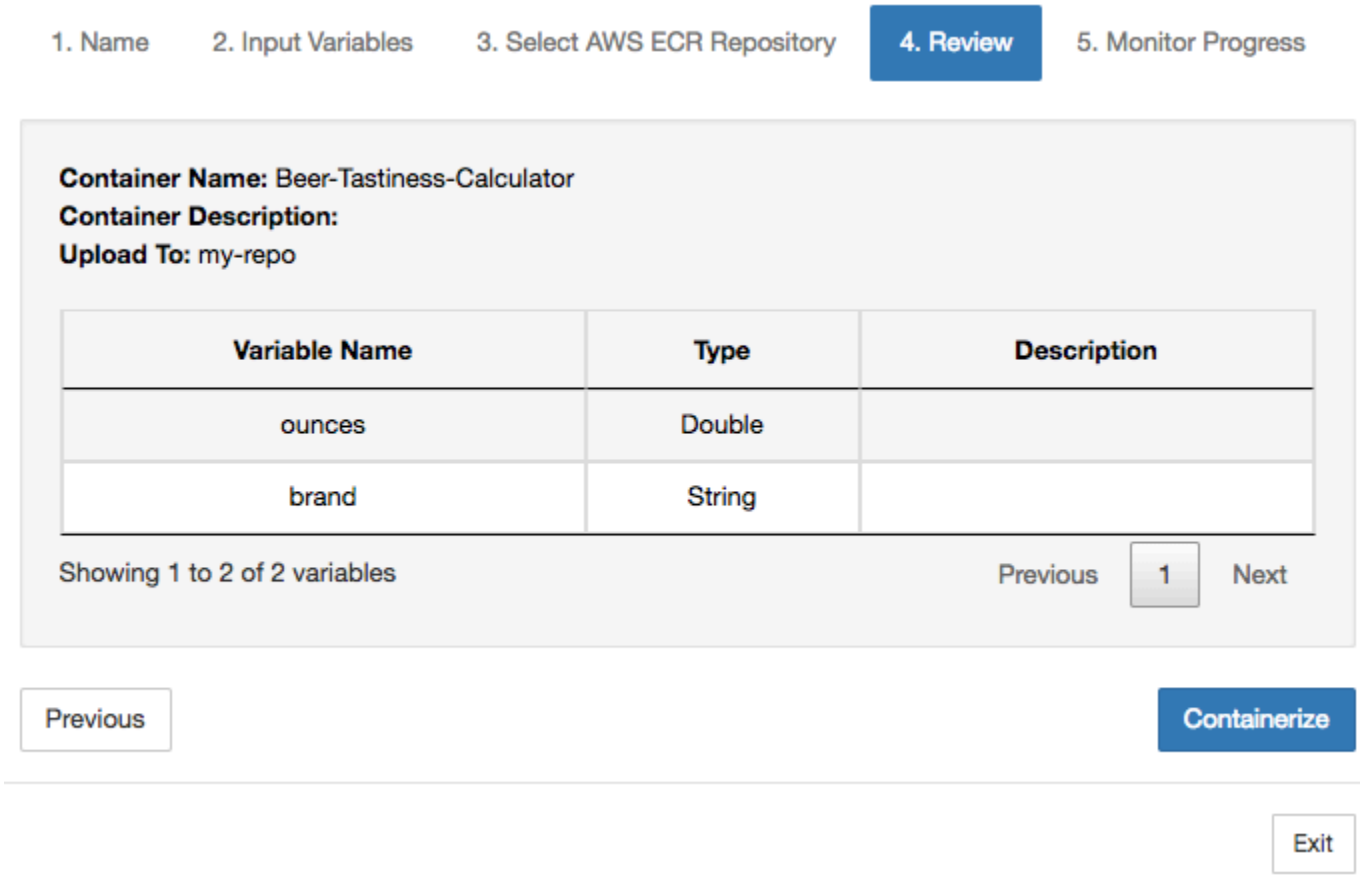


5. Choisissez le référentiel Amazon ECR dans lequel l'image créée à partir du bloc-notes doit être téléchargée.

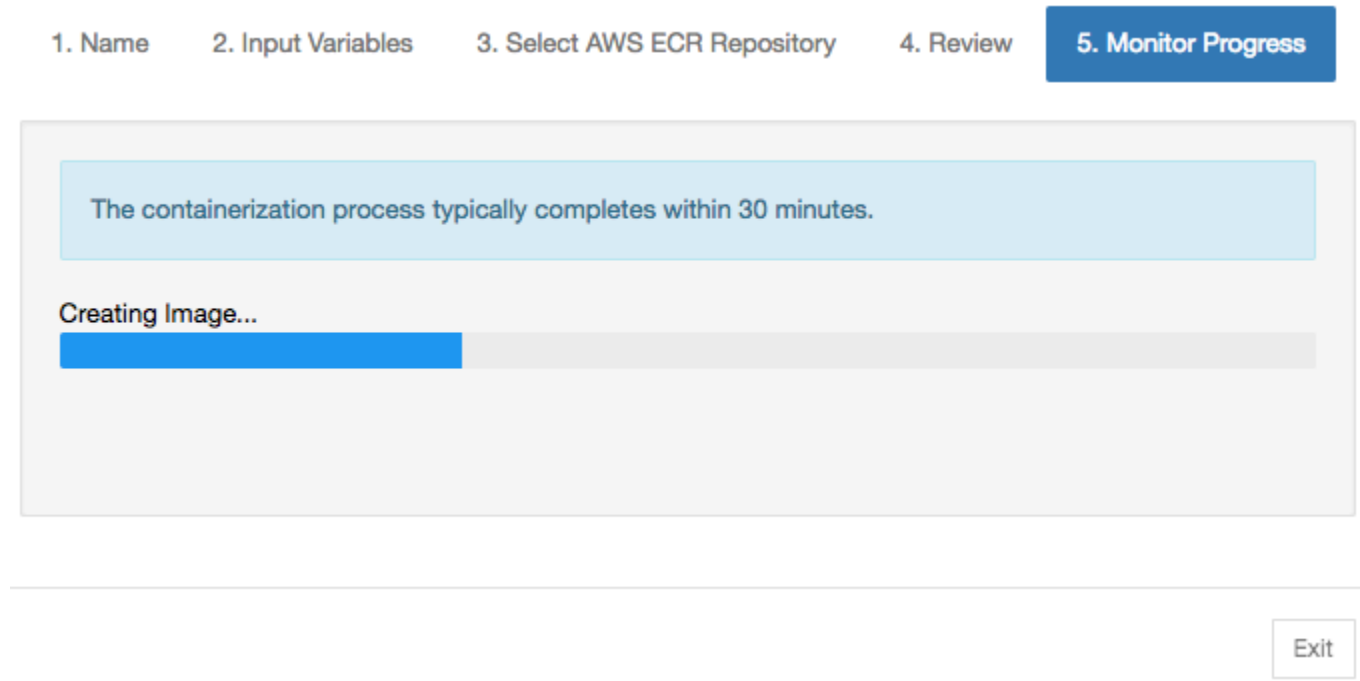


6. Choisissez Containerize pour démarrer le processus.

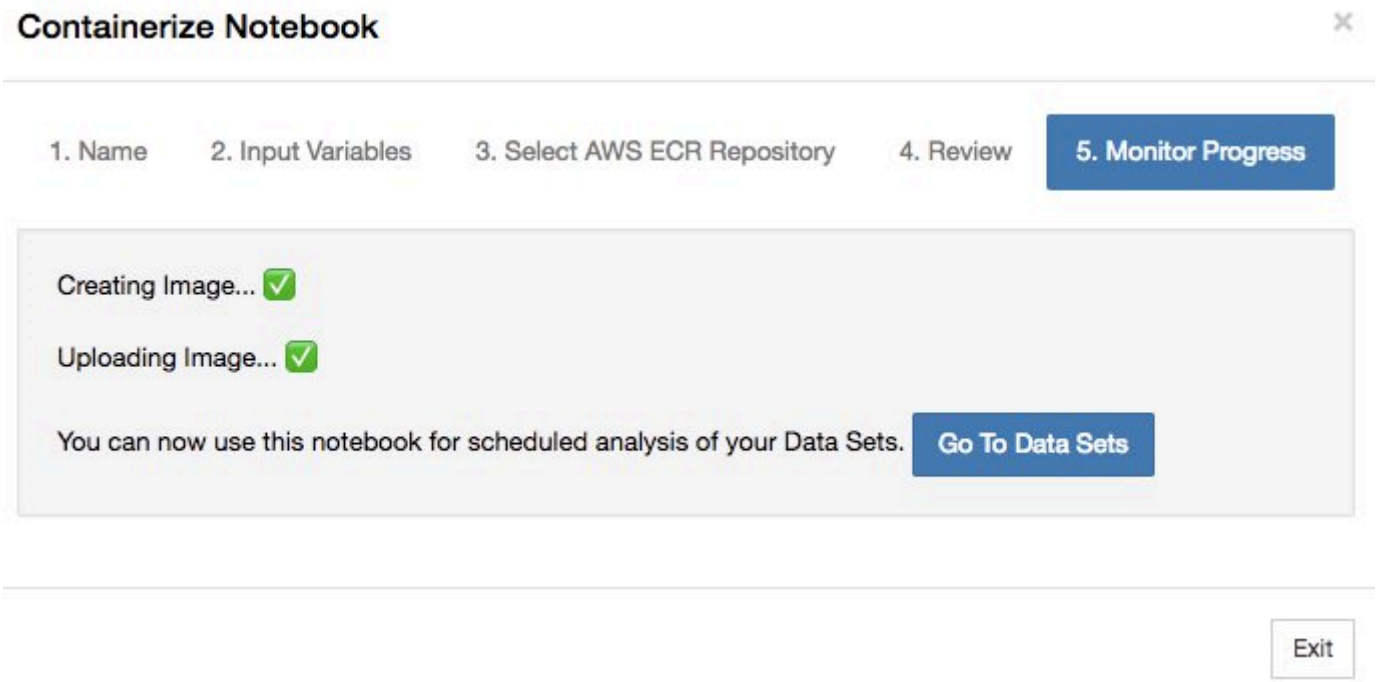
   Une vue d'ensemble vous est présentée résumant vos contributions. Notez qu'une fois le processus lancé, vous ne pouvez pas l'annuler. Le processus peut durer jusqu'à une heure.



7. La page suivante montre la progression.



8. Si vous fermez accidentellement votre navigateur, vous pouvez suivre l'état du processus de conteneurisation dans la section Carnets de notes de la console. AWS IoT Analytics
9. Une fois le processus terminé, l'image conteneurisée est stockée sur Amazon ECR, prête à être utilisée.



Utilisation d'un conteneur personnalisé pour l'analyse

Cette section contient des informations sur la création d'un conteneur Docker à l'aide d'un bloc-notes Jupyter. Un risque de sécurité existe si vous réutilisez des blocs-notes créés par des tiers : ces conteneurs peuvent exécuter du code arbitraire avec vos autorisations utilisateur. En outre, le code HTML généré par le bloc-notes peut être affiché dans la AWS IoT Analytics console, fournissant ainsi un vecteur d'attaque potentiel sur l'ordinateur qui affiche le code HTML. Assurez-vous que vous faites confiance à l'auteur de tout bloc-notes tiers avant de l'utiliser.

Vous pouvez créer votre propre conteneur personnalisé et l'exécuter avec le AWS IoT Analytics service. Pour ce faire, vous configurez une image Docker et vous la chargez sur Amazon ECR, puis vous configurez un ensemble de données pour exécuter une action de conteneur. Cette section fournit un exemple de cette procédure avec Octave.

Ce didacticiel présume également que les actions suivantes ont été effectuées :

- Octave a été installé sur votre ordinateur local.

- Un compte Docker configuré sur votre ordinateur local
- Un AWS compte Amazon ECR ou un accès AWS IoT Analytics

Étape 1 : Configurer une image Docker

Il existe trois fichiers principaux dont vous avez besoin pour ce didacticiel. Voici leur nom et leur contenu :

- `Dockerfile`— Configuration initiale du processus de conteneurisation de Docker.

```
FROM ubuntu:16.04

# Get required set of software
RUN apt-get update
RUN apt-get install -y software-properties-common
RUN apt-get install -y octave
RUN apt-get install -y python3-pip

# Get boto3 for S3 and other libraries
RUN pip3 install --upgrade pip
RUN pip3 install boto3
RUN pip3 install urllib3

# Move scripts over
ADD moment moment
ADD run-octave.py run-octave.py

# Start python script
ENTRYPOINT ["python3", "run-octave.py"]
```

- `run-octave.py`— Analyse le code JSON depuis AWS IoT Analytics, exécute le script Octave et télécharge les artefacts sur Amazon S3.

```
import boto3
import json
import os
import sys
from urllib.parse import urlparse

# Parse the JSON from IoT Analytics
with open('/opt/ml/input/data/iotanalytics/params') as params_file:
    params = json.load(params_file)
```

```
variables = params['Variables']

order = variables['order']
input_s3_bucket = variables['inputDataS3BucketName']
input_s3_key = variables['inputDataS3Key']
output_s3_uri = variables['octaveResultS3URI']

local_input_filename = "input.txt"
local_output_filename = "output.mat"

# Pull input data from S3...
s3 = boto3.resource('s3')
s3.Bucket(input_s3_bucket).download_file(input_s3_key, local_input_filename)

# Run Octave Script
os.system("octave moment {} {} {}".format(local_input_filename,
 local_output_filename, order))

# # Upload the artifacts to S3
output_s3_url = urlparse(output_s3_uri)
output_s3_bucket = output_s3_url.netloc
output_s3_key = output_s3_url.path[1:]

s3.Object(output_s3_bucket, output_s3_key).put(Body=open(local_output_filename,
 'rb'), ACL='bucket-owner-full-control')
```

- `moment`— Un script Octave simple qui calcule le moment en fonction d'un fichier d'entrée ou de sortie et d'un ordre spécifié.

```
#!/usr/bin/octave -qf

arg_list = argv ();
input_filename = arg_list{1};
output_filename = arg_list{2};
order = str2num(arg_list{3});

[D,delimiterOut]=importdata(input_filename)
M = moment(D, order)

save(output_filename,'M')
```

1. Téléchargez le contenu de chaque fichier. Créez un nouveau répertoire et placez-y tous les fichiers, puis `cd` dans ce répertoire.
2. Exécutez la commande suivante.

```
docker build -t octave-moment .
```

3. Vous devriez voir une nouvelle image dans votre dépôt Docker. Vérifiez-le en exécutant la commande suivante.

```
docker image ls | grep octave-moment
```

Étape 2 : télécharger l'image Docker dans un référentiel Amazon ECR

1. Créez un référentiel dans Amazon ECR.

```
aws ecr create-repository --repository-name octave-moment
```

2. Connectez-vous à votre environnement Docker.

```
aws ecr get-login
```

3. Copiez la sortie et exécutez-la. Le résultat devrait ressembler à ce qui suit.

```
docker login -u AWS -p password -e none https://your-aws-account-
id.dkr.ecr..amazonaws.com
```

4. Marquez l'image que vous avez créée avec le tag du référentiel Amazon ECR.

```
docker tag your-image-id  your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-
moment
```

5. Poussez l'image vers Amazon ECR.

```
docker push your-aws-account-id.dkr.ecr.region.amazonaws.com/octave-moment
```

Étape 3 : Chargez vos exemples de données dans un compartiment Amazon S3

1. Téléchargez ce qui suit dans un fichier`input.txt`.

```
0.857549  -0.987565  -0.467288  -0.252233  -2.298007
 0.030077  -1.243324  -0.692745   0.563276   0.772901
-0.508862  -0.404303  -1.363477  -1.812281  -0.296744
-0.203897   0.746533   0.048276   0.075284   0.125395
 0.829358   1.246402  -1.310275  -2.737117   0.024629
 1.206120   0.895101   1.075549   1.897416   1.383577
```

2. Créez un compartiment Amazon S3 appelé`octave-sample-data-`*`your-aws-account-id`*.
3. Téléchargez le fichier dans `input.txt` le compartiment Amazon S3 que vous venez de créer. Vous devriez maintenant avoir un bucket nommé `octave-sample-data-`*`your-aws-account-id`* contenant le `input.txt` fichier.

Étape 4 : Créer un rôle d'exécution de conteneur

1. Copiez ce qui suit dans un fichier nommé`role1.json`. *`your-aws-account-id`*Remplacez-le par votre numéro de AWS compte et *`aws-region`* par la AWS région de vos AWS ressources.

> **Note**
>
> Cet exemple inclut une clé de contexte de condition globale pour se protéger contre le problème confus de sécurité adjoint. Pour de plus amples informations, veuillez consulter the section called "Prévention du cas de figure de l'adjoint désorienté entre services".

JSON

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

```
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset"
                }
            }
        }
    ]
}
```

2. Créez un rôle qui donne des autorisations d'accès à SageMaker AI et AWS IoT Analyticsà l'aide du fichier `role1.json` que vous avez téléchargé.

```
aws iam create-role --role-name container-execution-role --assume-role-policy-
document file://role1.json
```

3. Téléchargez ce qui suit dans un fichier nommé `policy1.json` et remplacez-le *your-account-id* par votre identifiant de compte (voir le deuxième ARN ci-dessous`Statement:Resource`).

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:PutObject",
                "s3:GetObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:*:*:dataset/*",
                "arn:aws:s3:*:*:octave-sample-data-123456789012/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iotanalytics:*"
```

```
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:GetLogEvents",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetBucketLocation",
                "s3:ListBucket",
                "s3:ListAllMyBuckets"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Créez une politique IAM à l'aide du `policy.json` fichier que vous venez de télécharger.

```
aws iam create-policy --policy-name ContainerExecutionPolicy --policy-document
 file://policy1.json
```

5. Attachez la stratégie au rôle.

```
aws iam attach-role-policy --role-name container-execution-role --policy-arn
 arn:aws:iam::your-account-id:policy/ContainerExecutionPolicy
```

Étape 5 : Création d'un ensemble de données avec une action de conteneur

1. Téléchargez ce qui suit dans un fichier nommé `cli-input.json` et remplacez toutes les instances de *`your-account-id`* et *`region`* par les valeurs appropriées.

```
{
    "datasetName": "octave_dataset",
    "actions": [
        {
            "actionName": "octave",
            "containerAction": {
                "image": "your-account-id.dkr.ecr.region.amazonaws.com/octave-
moment",
                "executionRoleArn": "arn:aws:iam::your-account-id:role/container-
execution-role",
                "resourceConfiguration": {
                    "computeType": "ACU_1",
                    "volumeSizeInGB": 1
                },
                "variables": [
                    {
                        "name": "octaveResultS3URI",
                        "outputFileUriValue": {
                            "fileName": "output.mat"
                        }
                    },
                    {
                        "name": "inputDataS3BucketName",
                        "stringValue": "octave-sample-data-your-account-id"
                    },
                    {
                        "name": "inputDataS3Key",
                        "stringValue": "input.txt"
                    },
                    {
                        "name": "order",
                        "stringValue": "3"
                    }
                ]
            }
        }
    ]
}
```

2. Créez un jeu de données à l'aide du fichier `cli-input.json` que vous venez de télécharger et de modifier.

```
aws iotanalytics create-dataset –cli-input-json file://cli-input.json
```

Étape 6 : Invoquer la génération de contenu du jeu de données

1. Exécutez la commande suivante.

```
aws iotanalytics create-dataset-content --dataset-name octave-dataset
```

Étape 7 : Obtenir le contenu du jeu de données

1. Exécutez la commande suivante.

```
aws iotanalytics get-dataset-content --dataset-name octave-dataset --version-id \
$LATEST
```

2. Vous devrez peut-être attendre plusieurs minutes jusqu'à ce que ce `DatasetContentState` soit le cas`SUCCEEDED`.

Étape 8 : Imprimer la sortie sur Octave

1. Utilisez le shell Octave pour imprimer la sortie du conteneur en exécutant la commande suivante.

```
bash> octave
octave> load output.mat
octave> disp(M)
-0.016393 -0.098061 0.380311 -0.564377 -1.318744
```

Visualisation des données AWS IoT Analytics

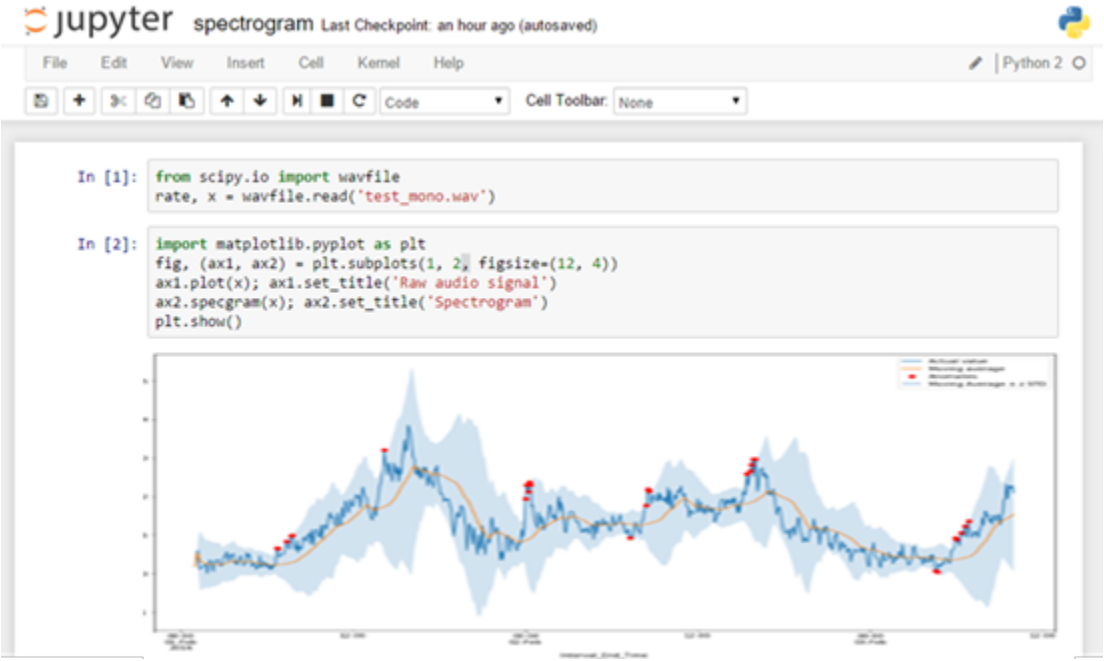
Pour visualiser vos AWS IoT Analytics données, vous pouvez utiliser la AWS IoT Analytics console ou Quick Suite.

Rubriques

- Visualisation des AWS IoT Analytics données avec la console
- Visualisation des AWS IoT Analytics données avec Quick Suite

Visualisation des AWS IoT Analytics données avec la console

AWS IoT Analytics peut intégrer la sortie HTML de votre jeu de données conteneur (qui se trouve dans le fichier`output.html`) sur la page de contenu du jeu de données conteneur de la AWS IoT Analytics console. Par exemple, si vous définissez un jeu de données conteneur qui exécute un bloc-notes Jupyter et que vous créez une visualisation dans votre bloc-notes Jupyter, votre jeu de données peut ressembler à ce qui suit.



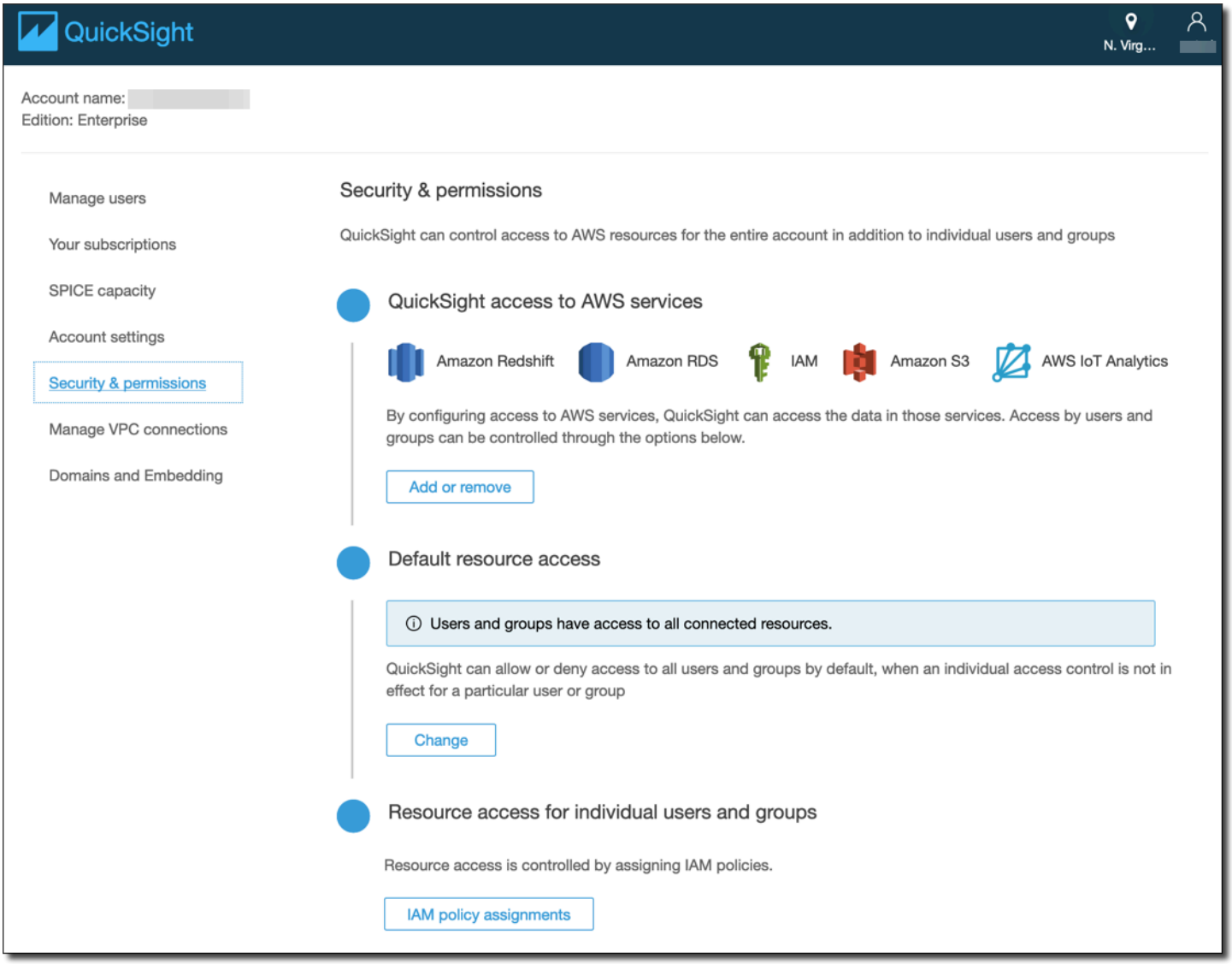
Ensuite, une fois le contenu du jeu de données conteneur créé, vous pouvez afficher cette visualisation sur la page de contenu de l'ensemble de données de la console.

Pour plus d'informations sur la création d'un jeu de données conteneur qui exécute un bloc-notes Jupyter, consultez Automatisation de votre flux de travail.

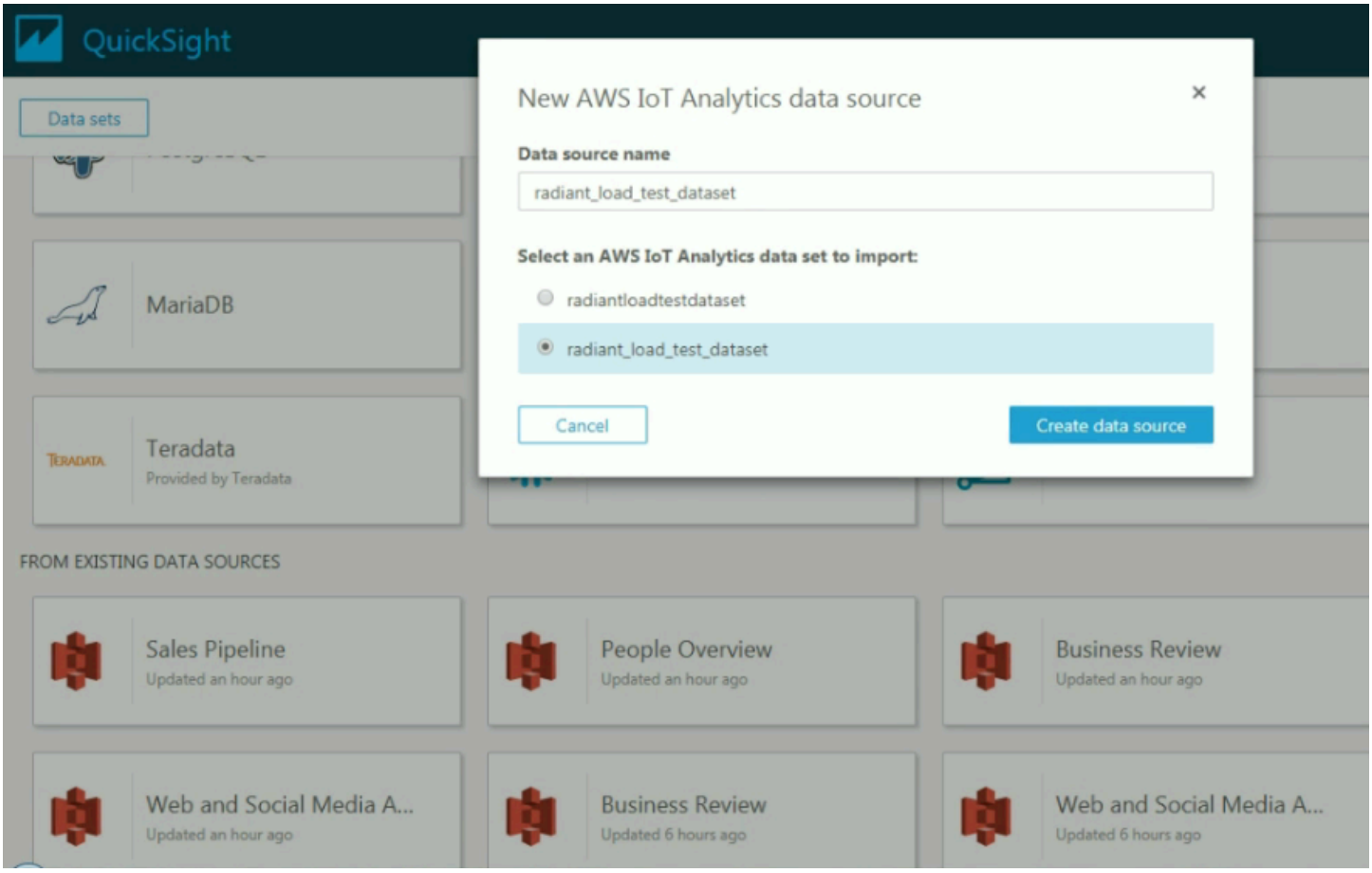
Visualisation des AWS IoT Analytics données avec Quick Suite

AWS IoT Analytics fournit une intégration directe avec Quick Suite. Quick Suite est un service d'analyse commerciale rapide que vous pouvez utiliser pour créer des visualisations, effectuer des analyses ad hoc et obtenir rapidement des informations commerciales à partir de vos données. Quick Suite permet aux entreprises de s'adapter à des centaines de milliers d'utilisateurs et fournit des performances réactives grâce à un puissant moteur en mémoire (SPICE). Vous pouvez sélectionner vos AWS IoT Analytics ensembles de données dans la console Quick Suite et commencer à créer des tableaux de bord et des visualisations. Quick Suite est disponible dans ces régions.

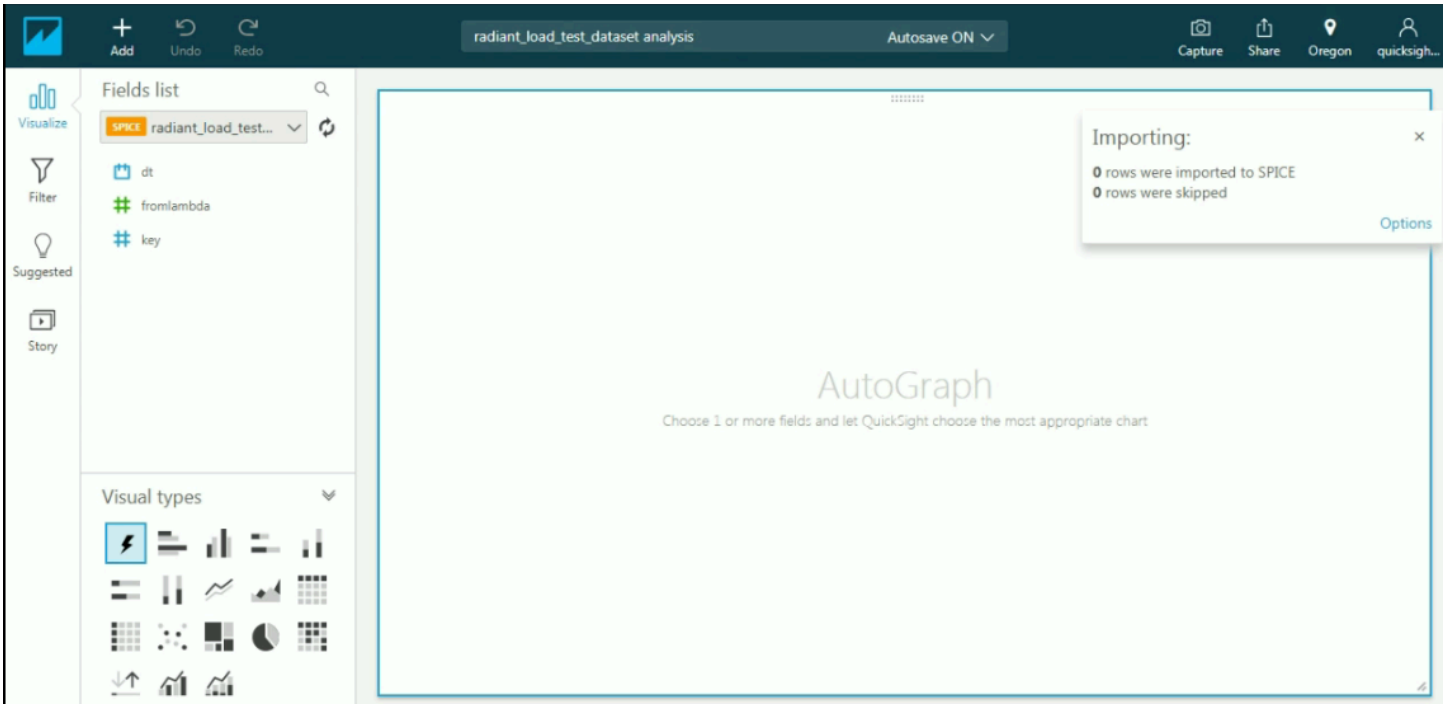
Pour commencer à utiliser vos visualisations Quick Suite, vous devez créer un compte Quick Suite. Assurez-vous d'autoriser Quick Suite à accéder à vos AWS IoT Analytics données lorsque vous configurez votre compte. Si vous avez déjà un compte, autorisez Quick Suite à accéder à vos AWS IoT Analytics données en choisissant Admin, Manage QuickSight, Security & permissions. Sous QuickSight Accès aux AWS services, choisissez Ajouter ou supprimer, puis cochez la case à côté AWS IoT Analyticset choisissez Mettre à jour.



Une fois votre compte configuré, sur la page de console d'administration de Quick Suite, sélectionnez Nouvelle analyse et Nouvel ensemble de données, puis choisissez AWS IoT Analytics comme source. Entrez un nom pour votre source de données, choisissez un jeu de données à importer, puis choisissez Créer une source de données.



Une fois votre source de données créée, vous pouvez créer des visualisations dans Quick Suite.



Pour plus d'informations sur les tableaux de bord et les ensembles de données de Quick Suite, consultez la documentation de Quick Suite.

Marquer vos ressources AWS IoT Analytics

Pour vous aider à gérer vos canaux, ensembles de données, magasins de données et pipelines, vous pouvez affecter vos propres métadonnées à chaque ressource sous la forme de balises. Ce chapitre décrit les balises et explique comment les créer.

Rubriques

- Principes de base des étiquettes
- Utilisation des balises avec des politiques IAM
- Restrictions liées aux étiquettes

Principes de base des étiquettes

Les balises vous permettent de classer vos AWS IoT Analytics ressources de différentes manières, par exemple par objectif, propriétaire ou environnement. C'est une fonction utile lorsque vous avez de nombreuses ressources de même type : elle vous permet d'identifier rapidement une ressource spécifique en fonction des balises que vous lui avez attribuées. Chaque balise est constituée d'une clé et d'une valeur facultative que vous définissez. Par exemple, vous pouvez définir un ensemble de balises pour vos canaux afin de suivre le type d'appareil responsable de la source de message de chaque canal. Nous vous recommandons de concevoir un ensemble de clés d'étiquette répondant à vos besoins pour chaque type de ressource. L'utilisation d'un ensemble de clés de balise cohérent facilite la gestion de vos ressources. Vous pouvez rechercher et filtrer les ressources en fonction des étiquettes que vous ajoutez.

Vous pouvez également utiliser les balises pour classer les coûts par catégorie et en effectuer le suivi. Lorsque vous appliquez des balises à des canaux, à des ensembles de données, à des banques de données ou à des pipelines, vous AWS générez un rapport de répartition des coûts sous forme de fichier CSV (valeurs séparées par des virgules) avec votre utilisation et vos coûts agrégés par vos balises. Vous pouvez appliquer des balises associées à des catégories métier (telles que les centres de coûts, les noms d'applications ou les propriétaires) pour organiser les coûts relatifs à divers services. Pour plus d'informations sur l'utilisation des balises pour la répartition des coûts, voir Utiliser les balises de répartition des coûts dans le Guide de AWS Billing l'utilisateur.

Pour faciliter l'utilisation, utilisez l'éditeur de balises de la AWS Billing and Cost Management console, qui fournit un moyen centralisé et unifié de créer et de gérer vos balises. Pour plus d'informations,

consultez la section Utilisation de l'éditeur de balises dans Getting started with the AWS Management Console.

Vous pouvez également travailler avec des balises à l'aide de l'API AWS CLI et de l' AWS IoT Analytics API. Vous pouvez associer des balises à des canaux, ensembles de données, magasins de données et pipelines lorsque vous les créez. Utilisez le champ Balises dans les commandes suivantes :

- CreateChannel
- CreateDataset
- CreateDatastore
- CreatePipeline

Vous pouvez ajouter, modifier ou supprimer des balises pour les ressources existantes qui prennent en charge le balisage. Utilisez les commandes suivantes :

- TagResource
- ListTagsForResource
- UntagResource

Vous pouvez modifier les clés et valeurs de balise, et vous pouvez retirer des balises d'une ressource à tout moment. Vous pouvez définir la valeur d'une balise sur une chaîne vide, mais vous ne pouvez pas définir la valeur d'une balise sur null. Si vous ajoutez une balise ayant la même clé qu'une balise existante sur cette ressource, la nouvelle valeur remplace l'ancienne valeur. Si vous supprimez une ressource, toutes les balises associées à celle-ci sont également supprimées.

Utilisation des balises avec des politiques IAM

Vous pouvez utiliser l'`Condition`élément (également appelé `Condition` bloc) avec le contexte de condition suivant keys/values dans une politique IAM pour contrôler l'accès des utilisateurs (autorisations) en fonction des balises d'une ressource :

- Permet d'`iotanalytics:ResourceTag/<tag-key>: <tag-value>`autoriser ou de refuser les actions des utilisateurs sur les ressources dotées de balises spécifiques.

- Utilisez `aws:RequestTag/<tag-key>: <tag-value>` pour exiger qu'une balise spécifique soit utilisée (ou ne soit pas utilisée) lorsque vous effectuez une demande d'API pour créer ou modifier une ressource qui autorise les balises.
- Utilisez `aws:TagKeys: [<tag-key>, ...]` pour exiger qu'un ensemble de clés de balise spécifique soit utilisé (ou ne soit pas utilisé) lorsque vous effectuez une demande d'API pour créer ou modifier une ressource qui autorise les balises.

> **Note**
>
> Le contexte de condition keys/values d'une politique IAM ne s'applique qu'aux AWS IoT Analytics actions pour lesquelles un identifiant pour une ressource susceptible d'être étiquetée est un paramètre obligatoire. Par exemple, l'utilisation de n'DescribeLoggingOptionsest pas allowed/denied basée sur le contexte des conditions keys/values car aucune ressource balisable (canal, ensemble de données, magasin de données ou pipeline) n'est référencée dans cette demande.

Pour de plus amples informations, veuillez consulter Contrôle de l'accès à l'aide d'identifications dans le Guide de l'utilisateur IAM. La section de référence des politiques JSON IAM de ce guide contient une syntaxe détaillée, des descriptions et des exemples des éléments, des variables et de la logique d'évaluation des politiques JSON dans IAM.

L'exemple de politique suivant applique des restrictions basées sur deux bases. Un utilisateur soumis à des restrictions en vertu de cette politique :

1. Impossible de donner à une ressource le tag « env=prod » (voir la ligne `"aws:RequestTag/env" : "prod"` dans l'exemple).
2. Impossible de modifier ou d'accéder à une ressource qui possède une balise existante « env=prod » (voir la ligne `"iotanalytics:ResourceTag/env" : "prod"` dans l'exemple).

JSON

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

```
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "aws:RequestTag/env" : "prod"
        }
      }
    },
    {
      "Effect" : "Deny",
      "Action" : "iotanalytics:*",
      "Resource" : "*",
      "Condition" : {
        "StringEquals" : {
          "iotanalytics:ResourceTag/env" : "prod"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iotanalytics:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Vous pouvez également spécifier plusieurs valeurs de balise pour une clé de balise donnée en les insérant dans une liste, comme dans l'exemple suivant.

```
"StringEquals" : {
  "iotanalytics:ResourceTag/env" : ["dev", "test"]
}
```

> **Note**
>
> Si vos allow/deny utilisateurs accèdent aux ressources par le biais de balises, il est important d'envisager de refuser explicitement aux utilisateurs la possibilité d'ajouter ces balises

ou de les supprimer des mêmes ressources. Sinon, il sera possible pour un utilisateur de contourner vos restrictions et d'obtenir l'accès à une ressource en modifiant ses balises.

Restrictions liées aux étiquettes

Les restrictions de base suivantes s'appliquent aux balises :

- Nombre maximal de balises par ressource : 50
- Longueur de clé maximale : 127 caractères Unicode en UTF-8
- Longueur de valeur maximale : 255 caractères Unicode en UTF-8
- Les clés et valeurs d'étiquette sont sensibles à la casse.
- N'utilisez pas le `aws: prefix` dans les noms ou les valeurs de vos balises, car il est réservé à un AWS usage réservé. Vous ne pouvez pas modifier ou supprimer des noms ou valeurs de balise ayant ce préfixe. Les balises portant ce préfixe ne sont pas prises en compte dans votre limite de balises par source.
- Si votre schéma de balisage est utilisé pour plusieurs services et ressources , n'oubliez pas que d'autres services peuvent avoir des restrictions concernant les caractères autorisés. Les caractères généralement autorisés sont les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : + - = . _ : / @.

Expressions SQL dans AWS IoT Analytics

Les ensembles de données sont générés à l'aide d'expressions SQL sur les données d'un magasin de données. AWS IoT Analytics utilise les mêmes requêtes, fonctions et opérateurs SQL qu'Amazon Athena.

AWS IoT Analytics prend en charge un sous-ensemble de la syntaxe SQL standard ANSI.

```
SELECT [ ALL | DISTINCT ] select_expression [, ...]
[ FROM from_item [, ...] ]
[[ INNER | OUTER ] LEFT | RIGHT | FULL | CROSS JOIN join_item [ ON join_condition ]]
[ WHERE condition ]
[ GROUP BY [ ALL | DISTINCT ] grouping_element [, ...] ]
[ HAVING condition ]
[ UNION [ ALL | DISTINCT ] union_query ]
[ ORDER BY expression [ ASC | DESC ] [ NULLS FIRST | NULLS LAST] [, ...] ]
[ LIMIT [ count | ALL ] ]
```

Pour obtenir une description des paramètres, consultez la section Paramètres dans la documentation Amazon Athena.

AWS IoT Analytics et Amazon Athena ne prend pas en charge les solutions suivantes :

- `WITH`clauses.
- Instructions `CREATE TABLE AS SELECT`
- Instructions `INSERT INTO`
- Des déclarations préparées que vous ne pouvez pas `EXECUTE` utiliser`USING`.
- `CREATE TABLE LIKE`
- `DESCRIBE INPUT` et `DESCRIBE OUTPUT`
- Instructions `EXPLAIN`
- Fonctions définies par l'utilisateur (UDFs ou UDAFs)
- Procédures stockées
- Connecteurs fédérés

Rubriques

- Fonctionnalité SQL prise en charge dans AWS IoT Analytics

- Résoudre les problèmes courants liés aux requêtes SQL dans AWS IoT Analytics

Fonctionnalité SQL prise en charge dans AWS IoT Analytics

Les ensembles de données sont générés à l'aide d'expressions SQL sur les données d'un magasin de données. Les requêtes que vous exécutez AWS IoT Analytics sont basées sur Presto 0.217.

Types de données pris en charge

AWS IoT Analytics et Amazon Athena prend en charge ces types de données.

- primitive_type
  - `TINYINT`
  - `SMALLINT`
  - `INT`
  - `BIGINT`
  - `BOOLEAN`
  - `DOUBLE`
  - `FLOAT`
  - `STRING`
  - `TIMESTAMP`
  - `DECIMAL(precision, scale)`
  - `DATE`
  - `CHAR`(données de caractères de longueur fixe avec une longueur spécifiée)
  - `VARCHAR`(données de caractères de longueur variable avec une longueur spécifiée)
- array_type
  - `ARRAY<data_type>`
- map_type
  - `MAP<primitive_type, data_type>`
- struct_type
  - `STRUCT<col_name:data_type[COMMENT col_comment][,...]>`

> **Note**
>
> AWS IoT Analytics et Amazon Athena ne prend pas en charge certains types de données.

Fonctions prises en charge

Amazon Athena et les fonctionnalités AWS IoT Analytics SQL sont basées sur Presto 0.217. Pour plus d'informations sur les fonctions, les opérateurs et les expressions associés, consultez la section Fonctions et opérateurs et les sections spécifiques suivantes de la documentation Presto.

- Opérateurs logiques
- Fonctions et opérateurs de comparaison
- Expressions conditionnelles
- Fonctions de conversion
- Fonctions et opérateurs mathématiques
- Fonctions bitwise
- Fonctions et opérateurs décimaux
- Fonctions et opérateurs de chaînes de caractères
- Fonctions binaires
- Fonctions et opérateurs de la date et de l'heure
- Fonctions d'expression régulière
- Fonctions et opérateurs JSON
- Fonctions URL
- Fonctions d'agrégation
- Fonctions de fenêtrage
- Fonctions de couleur
- Opérateurs et fonctions de tableau
- Fonctions et opérateurs de mappage
- Expressions et fonctions lambda
- Fonctions Teradata

> **Note**
>
> AWS IoT Analytics et Amazon Athena ne prennent pas en charge les fonctions définies par l'utilisateur (UDFs ou UDAFs) ni les procédures stockées.

Résoudre les problèmes courants liés aux requêtes SQL dans AWS IoT Analytics

Utilisez les informations suivantes pour résoudre les problèmes liés à vos requêtes SQL dans AWS IoT Analytics.

- Pour éviter un guillemet simple, faites-le précéder d'un autre guillemet simple. Ne confondez pas cela avec un guillemet double.

  Example exemple

  ```
  SELECT 'O''Reilly'
  ```

- Pour éviter les traits de soulignement, utilisez des backticks pour inclure les noms des colonnes du magasin de données commençant par un trait de soulignement.

  Example exemple

  ```
  SELECT `_myMessageAttribute` FROM myDataStore
  ```

- Pour éviter les noms contenant des chiffres, placez les noms des banques de données qui incluent des nombres entre guillemets doubles.

  Example exemple

  ```
  SELECT * FROM "myDataStore123"
  ```

- Pour éviter les mots clés réservés, placez les mots clés réservés entre guillemets. Pour plus d'informations, consultez la section Liste des mots clés réservés dans les instructions SQL SELECT.

Sécurité dans AWS IoT Analytics

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. L'efficacité de notre sécurité est régulièrement testée et vérifiée par des auditeurs tiers dans le cadre des programmes de conformitéAWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS IoT Analytics, consultez la section AWS Services concernés par programme de conformité.
- Sécurité dans le cloud : votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre organisation, et la législation et la réglementation applicables.

Cette documentation vous aidera à comprendre comment appliquer le modèle de responsabilité partagée lors de son utilisation AWS IoT Analytics. Les rubriques suivantes expliquent comment procéder à la configuration AWS IoT Analytics pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui peuvent vous aider à surveiller et à sécuriser vos AWS IoT Analytics ressources.

Gestion des identités et des accès AWS dans AWS IoT Analytics

Gestion des identités et des accès AWS (IAM) est un AWS service 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 Analytics les ressources. IAM est un AWS service que vous pouvez utiliser sans frais supplémentaires.

Public ciblé

La façon dont vous utilisez Gestion des identités et des accès AWS (IAM) varie en fonction de votre rôle :

- Utilisateur du service : demandez des autorisations à votre administrateur si vous ne pouvez pas accéder aux fonctionnalités (voir Résolution des problèmes AWS IoT Analytics d'identité et d'accès)
- Administrateur du service : déterminez l'accès des utilisateurs et soumettez les demandes d'autorisation (voir Comment AWS IoT Analytics fonctionne avec IAM)
- Administrateur IAM : rédigez des politiques pour gérer l'accès (voir AWS IoT Analytics exemples de politiques basées sur l'identité)

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en tant qu'identité fédérée à l'aide d'informations d'identification provenant d'une source d'identité telle que AWS IAM Identity Center (IAM Identity Center), d'une authentification unique ou d'informations d'identification. Google/Facebook Pour plus d'informations sur la connexion, consultez Connexion à votre Compte AWS dans le Guide de l'utilisateur Connexion à AWS .

Pour l'accès par programmation, AWS fournit un SDK et une CLI pour signer les demandes de manière cryptographique. Pour plus d'informations, consultez Signature AWS Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une seule identité de connexion appelée utilisateur Compte AWS root qui dispose d'un accès complet à toutes Services AWS les ressources. Il est vivement déconseillé d'utiliser l'utilisateur racine pour vos tâches quotidiennes. Pour les tâches qui requièrent des informations d'identification de l'utilisateur racine, consultez Tâches qui requièrent les informations d'identification de l'utilisateur racine dans le Guide de l'utilisateur IAM.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité qui dispose d'autorisations spécifiques pour une seule personne ou application. Nous vous recommandons d'utiliser ces informations d'identification temporaires au lieu des utilisateurs IAM avec des informations d'identification à long terme. Pour plus d'informations,

voir Exiger des utilisateurs humains qu'ils utilisent la fédération avec un fournisseur d'identité pour accéder à AWS l'aide d'informations d'identification temporaires dans le guide de l'utilisateur IAM.

Les groupes IAM spécifient une collection d'utilisateurs IAM et permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Pour plus d'informations, consultez la section Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité dotée d'autorisations spécifiques qui fournit des informations d'identification temporaires. Vous pouvez assumer un rôle en passant d'un rôle d'utilisateur à un rôle IAM (console) ou en appelant une opération d' AWS API AWS CLI ou d'API. Pour plus d'informations, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM sont utiles pour l'accès des utilisateurs fédérés, les autorisations temporaires des utilisateurs IAM, les accès entre comptes, les accès entre services et pour les applications exécutées sur Amazon. EC2 Pour plus d'informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique définit les autorisations lorsqu'elles sont associées à une identité ou à une ressource. AWS évalue ces politiques lorsqu'un directeur fait une demande. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations les documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

À l'aide de politiques, les administrateurs précisent qui a accès à quoi en définissant quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Un administrateur IAM crée des politiques IAM et les ajoute aux rôles, que les utilisateurs peuvent ensuite assumer. Les politiques IAM définissent les autorisations quelle que soit la méthode que vous utilisez pour exécuter l'opération.

Politiques basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous attachez à une identité (utilisateur, groupe ou rôle). Ces politiques contrôlent les actions que peuvent

exécuter ces identités, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être des politiques intégrées (intégrées directement dans une seule identité) ou des politiques gérées (politiques autonomes associées à plusieurs identités). Pour découvrir comment choisir entre des politiques gérées et en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Autres types de politique

AWS prend en charge des types de politiques supplémentaires qui peuvent définir les autorisations maximales accordées par les types de politiques les plus courants :

- Limites d'autorisations : une limite des autorisations définit le nombre maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM. Pour plus d'informations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCPs) — Spécifiez les autorisations maximales pour une organisation ou une unité organisationnelle dans AWS Organizations. Pour plus d'informations, veuillez consulter Politiques de contrôle de service du Guide de l'utilisateur AWS Organizations .
- Politiques de contrôle des ressources (RCPs) : définissez le maximum d'autorisations disponibles pour les ressources de vos comptes. Pour plus d'informations, voir Politiques de contrôle des ressources (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de session : politiques avancées que vous passez en tant que paramètre lorsque vous créez par programmation une session temporaire pour un rôle ou un utilisateur fédéré. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS IoT Analytics fonctionne avec IAM

Avant d'utiliser IAM pour gérer l'accès à AWS IoT Analytics, vous devez connaître les fonctionnalités IAM disponibles. AWS IoT Analytics Pour obtenir une vue d'ensemble de la façon dont AWS IoT

Analytics les autres AWS services fonctionnent avec IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur d'IAM.

Rubriques de cette page :

- AWS IoT Analytics politiques basées sur l'identité
- AWS IoT Analytics politiques basées sur les ressources
- Autorisation basée sur les AWS IoT Analytics tags
- AWS IoT Analytics Rôles IAM

AWS IoT Analytics politiques basées sur l'identité

Avec les politiques basées sur l'identité IAM, vous pouvez spécifier les actions et les ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. AWS IoT Analytics 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

L'élément `Action` d'une stratégie basée sur une identité IAM décrit les actions spécifiques qui seront autorisées ou refusées par la stratégie. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Les actions sont utilisées dans une politique pour accorder les autorisations nécessaires à l'exécution de l'opération associée.

Action de politique en AWS IoT Analytics utilisant le préfixe suivant avant l'action : `iotanalytics:` Par exemple, pour autoriser quelqu'un à créer un AWS IoT Analytics canal avec l'opération AWS IoT Analytics `CreateChannel` API, vous incluez l'`iotanalytics:BatchPuMessage`action dans sa politique. Les déclarations de politique doivent inclure un `NotAction` élément `Action` ou. AWS IoT Analytics 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": [
    "iotanalytics:action1",
    "iotanalytics: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": "iotanalytics:Describe*"
```

Pour consulter la liste des AWS IoT Analytics actions, reportez-vous à la section Actions définies par AWS IoT Analytics dans le guide de l'utilisateur IAM.

Ressources

L'élément `Resource` précise les objets auxquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Vous spécifiez une ressource à l'aide d'un ARN ou du caractère générique (*) pour indiquer que l'instruction s'applique à toutes les ressources.

La ressource du AWS IoT Analytics jeu de données possède l'ARN suivant.

```
arn:${Partition}:iotanalytics:${Region}:${Account}:dataset/${DatasetName}
```

Pour plus d'informations sur le format de ARNs, consultez Amazon Resource Names (ARNs) et espaces de noms AWS de services.

Par exemple, pour spécifier l'ensemble de données `Foobar` dans votre instruction, utilisez l'ARN suivant.

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/Foobar"
```

Pour spécifier toutes les instances qui appartiennent à un compte spécifique, utilisez le caractère générique (*).

```
"Resource": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/*"
```

Certaines AWS IoT Analytics 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": "*"
```

Certaines actions AWS IoT Analytics d'API impliquent plusieurs ressources. Par exemple, les `CreatePipeline` références sous forme de canal et de jeu de données, un utilisateur doit être

autorisé à utiliser le canal et le jeu de données. 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 Analytics ressources et leurs caractéristiques ARNs, reportez-vous à la section Ressources définies par AWS IoT Analytics 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 Analytics.

Clés de condition

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, comme égal ou inférieur, pour faire correspondre la condition de la stratégie 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 opération `OR` 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 l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur. Pour plus d'informations, consultez Éléments des politiques IAM : variables et balises dans le Guide de l'utilisateur IAM.

AWS IoT Analytics ne fournit aucune clé de condition spécifique au service, mais il prend en charge l'utilisation de certaines clés de condition globales. Pour voir toutes les clés de condition AWS globales, consultez la section clés contextuelles de condition AWS globales dans le guide de l'utilisateur IAM.

Exemples

Pour consulter des exemples de politiques AWS IoT Analytics basées sur l'identité, consultez. AWS IoT Analytics exemples de politiques basées sur l'identité

AWS IoT Analytics politiques basées sur les ressources

AWS IoT Analytics ne prend pas en charge les politiques basées sur les ressources. Pour consulter un exemple de page détaillée sur les politiques basées sur les ressources, consultez la section Utilisation des politiques basées sur les ressources AWS Lambda dans le Guide du développeur.AWS Lambda

Autorisation basée sur les AWS IoT Analytics tags

Vous pouvez associer des balises aux AWS IoT Analytics ressources ou transmettre des balises dans une demande à AWS IoT Analytics. Pour contrôler l'accès en fonction des balises, vous devez fournir les informations relatives aux balises dans l'élément de condition d'une politique à l'aide des clés de `aws:TagKeys` condition `iotanalytics:ResourceTag/{key-name}`, `aws:RequestTag/{key-name}` ou. Pour plus d'informations sur le balisage AWS IoT Analytics des ressources, consultez la section Marquage de vos AWS IoT Analytics ressources.

Pour consulter un exemple de politique basée sur l'identité visant à limiter l'accès à une ressource en fonction des balises de cette ressource, consultez la section Affichage des AWS IoT Analytics chaînes en fonction des balises.

AWS IoT Analytics Rôles IAM

Un rôle IAM est une entité au sein de votre Compte AWS qui dispose d'autorisations spécifiques.

Utilisation d'informations d'identification temporaires avec AWS IoT Analytics

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 AWS Security Token Service (AWS STS) des opérations d'API telles que AssumeRoleou GetFederationToken.

AWS IoT Analytics ne prend pas en charge l'utilisation d'informations d'identification temporaires.

Rôles liés à un service

Les rôles liés aux services permettent au AWS service d'accéder aux ressources d'autres services pour effectuer une action en votre nom. Les rôles liés à un service s'affichent dans votre compte IAM et sont la propriété du service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

AWS IoT Analytics ne prend pas en charge les rôles liés à un service.

Rôles du 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 s'affichent dans votre compte IAM et sont la propriété du 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 Analytics prend en charge les rôles de service.

Prévention du cas de figure de l'adjoint désorienté entre services

Le problème de député 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 responsables de service qui ont obtenu l'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. Cela limite les autorisations qui AWS IoT Analytics fournissent un autre service à la ressource. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de stratégie.

Le moyen le plus efficace de se protéger contre le problème de l'adjoint confus est d'utiliser la clé de `aws:SourceArn` contexte de condition globale avec le nom de ressource Amazon (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:iotanalytics::123456789012:*`.

Rubriques

- Prévention pour les compartiments Amazon S3

- Prévention avec Amazon CloudWatch Logs
- Prévention des adjoints confuse pour les AWS IoT Analytics ressources gérées par le client

Prévention pour les compartiments Amazon S3

Si vous utilisez un stockage Amazon S3 géré par le client pour votre stockage de AWS IoT Analytics données, le compartiment Amazon S3 qui stocke vos données peut être exposé à des problèmes secondaires confus.

Par exemple, Nikki Wolf utilise un compartiment Amazon S3 appartenant au client appelé*DOC-EXAMPLE-BUCKET*. Le bucket stocke les informations relatives à un magasin de AWS IoT Analytics données créé dans la région*us-east-1*. Elle définit une politique qui permet au directeur du AWS IoT Analytics service de poser des questions *DOC-EXAMPLE-BUCKET* en son nom. La collègue de Nikki, Li Juan, interroge *DOC-EXAMPLE-BUCKET* depuis son propre compte et crée un ensemble de données avec les résultats. En conséquence, le directeur du AWS IoT Analytics service a interrogé le bucket Amazon S3 de Nikki au nom de Li, même si celle-ci a exécuté la requête depuis son compte.

Pour éviter cela, Nikki peut spécifier la `aws:SourceAccount` condition ou la `aws:SourceArn` condition dans la politique pour*DOC-EXAMPLE-BUCKET*.

Spécifiez la **`aws:SourceAccount`** condition - L'exemple suivant de politique de compartiment indique que seules les AWS IoT Analytics ressources du compte de Nikki (*123456789012*) peuvent y accéder*DOC-EXAMPLE-BUCKET*.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
```

```
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
            ],
            "Resource": [
                "arn:aws:s3:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Spécifiez la **`aws:SourceArn`** condition - Nikki peut également utiliser la `aws:SourceArn` condition.

JSON

```
{
    "Version":"2012-10-17",
    "Id": "MyPolicyID",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": [
                "s3:GetBucketLocation",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListMultipartUploadParts",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:DeleteObject"
```

```
            ],
            "Resource": [
                "arn:aws:s3:*:*:amzn-s3-demo-bucket",
                "arn:aws:s3:*:*:amzn-s3-demo-bucket/*"
            ],
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": [
                        "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset",
                        "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastore"
                    ]
                }
            }
        }
    ]
}
```

Prévention avec Amazon CloudWatch Logs

Vous pouvez éviter le problème de confusion des adjoints lors de la surveillance avec Amazon CloudWatch Logs. La politique de ressources suivante montre comment éviter le problème de confusion des adjoints avec :

- La clé de contexte de la condition globale, `aws:SourceArn`
- Le `aws:SourceAccount` avec votre identifiant AWS de compte
- La ressource client associée à la `sts:AssumeRole` demande dans AWS IoT Analytics

*123456789012*Remplacez-le par votre identifiant de AWS compte et *us-east-1* par la région de votre AWS IoT Analytics compte dans l'exemple suivant.

JSON

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

```
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:*/*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Pour plus d'informations sur l'activation et la configuration d'Amazon CloudWatch Logs, consultezthe section called "Journalisation et surveillance".

Prévention des adjoints confuse pour les AWS IoT Analytics ressources gérées par le client

Si vous accordez AWS IoT Analytics l'autorisation d'effectuer des actions sur vos AWS IoT Analytics ressources, celles-ci risquent d'être exposées à des problèmes confus avec les adjoints. Pour éviter le problème des adjoints confus, vous pouvez limiter les autorisations accordées à l' AWS IoT Analytics aide des exemples de politiques de ressources suivants.

Rubriques

- Prévention pour les AWS IoT Analytics canaux et les magasins de données
- Prévention de la confusion entre les services en matière de règles de diffusion du contenu des ensembles de AWS IoT Analytics données

Prévention pour les AWS IoT Analytics canaux et les magasins de données

Vous utilisez les rôles IAM pour contrôler les AWS ressources auxquelles AWS IoT Analytics vous pouvez accéder en votre nom. Pour éviter d'exposer votre rôle à un problème d'adjoint confus, vous pouvez spécifier le AWS compte dans l'`aws:SourceAccount`élément et l'ARN de la AWS IoT

Analytics ressource dans l'`aws:SourceArn`élément de la politique de confiance que vous associez à un rôle.

Dans l'exemple suivant, remplacez-le *123456789012* par votre identifiant de AWS compte et *arn:aws:iotanalytics:aws-region:123456789012:channel/DOC-EXAMPLE-CHANNEL* par l'ARN d'un AWS IoT Analytics canal ou d'un magasin de données.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:channel/your-channel"
                }
            }
        }
    ]
}
```

Pour en savoir plus sur les options de stockage S3 gérées par le client pour les canaux et les magasins de données, consultez `CustomerManagedChannelS3Storage`et `CustomerManagedDatastoreS3Storage`dans la référence des AWS IoT Analytics API.

Prévention de la confusion entre les services en matière de règles de diffusion du contenu des ensembles de AWS IoT Analytics données

Le rôle IAM qui est censé AWS IoT Analytics fournir les résultats des requêtes d'ensemble de données à Amazon S3 ou qui AWS IoT Events peut être exposé à des problèmes d'adjoints confus. Pour éviter le problème des adjoints confus, spécifiez le AWS compte dans l'`aws:SourceAccount`élément et l'ARN de la AWS IoT Analytics ressource dans l'`aws:SourceArn`élément de la politique de confiance que vous associez à votre rôle.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExampleTrustPolicyDocument",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-dataset"
                }
            }
        }
    ]
}
```

Pour plus de détails sur la configuration des règles de diffusion du contenu des ensembles de données, consultez `contentDeliveryRules`la référence des AWS IoT Analytics API.

AWS IoT Analytics exemples de politiques basées sur l'identité

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou modifier les ressources AWS IoT Analytics . Ils ne peuvent pas non plus effectuer de tâches à l'aide de l' AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour savoir comment créer une politique basée sur l'identité IAM à l'aide de ces exemples de documents de stratégie JSON, voir Création de politiques dans l'onglet JSON du guide de l'utilisateur IAM

Rubriques de cette page :

- Bonnes pratiques en matière de stratégies
- Utilisation de la AWS IoT Analytics console
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Accès à une AWS IoT Analytics entrée
- Affichage des AWS IoT Analytics chaînes en fonction des tags

Bonnes pratiques en matière de stratégies

Les politiques basées sur l'identité sont très puissantes. Ils déterminent si quelqu'un peut créer, accéder ou supprimer AWS IoT Analytics 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 à utiliser les politiques AWS gérées - Pour commencer à les utiliser AWS IoT Analytics rapidement, utilisez des politiques AWS gérées pour donner à vos employés les autorisations dont ils ont besoin. Ces politiques sont déjà disponibles dans votre compte et sont maintenues et mises à jour par AWS. Pour plus d'informations, voir Commencer à utiliser les autorisations avec les politiques AWS gérées dans le Guide de l'utilisateur IAM.
- Accorder le moindre privilège : lorsque vous créez des politiques personnalisées, accordez uniquement les autorisations requises pour effectuer une tâche. Commencez avec un ensemble d'autorisations minimum et accordez-en d'autres si nécessaire. Cette méthode est plus sûre que de commencer avec des autorisations trop permissives et d'essayer de les restreindre plus tard. Pour

plus d'informations, consultez Accorder le moindre privilège possible dans le Guide de l'utilisateur IAM.

- Activez l'authentification multifactorielle pour les opérations sensibles : pour plus de sécurité, demandez aux utilisateurs d'utiliser l'authentification multifactorielle (MFA) pour accéder aux ressources sensibles ou aux opérations d'API. Pour plus d'informations, consultez Utilisation de l'authentification multifactorielle (MFA) dans AWS dans le Guide de l'utilisateur IAM.
- Utilisez des conditions de politique pour renforcer la sécurité - Dans la mesure du possible, définissez les conditions dans lesquelles vos politiques basées sur l'identité autorisent l'accès à une ressource. Par exemple, vous pouvez écrire une condition pour spécifier une plage d'adresses IP autorisées d'où doit provenir une demande. Vous pouvez également rédiger des conditions pour autoriser les demandes uniquement dans un intervalle de date ou d'heure spécifié, ou pour exiger l'utilisation du protocole SSL ou MFA. Pour de plus amples informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Utilisation de la AWS IoT Analytics console

Pour accéder à la AWS IoT Analytics console, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher les détails AWS IoT Analytics des ressources de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) dotées de cette politique.

Pour garantir que ces entités peuvent toujours utiliser la AWS IoT Analytics console, associez également la politique AWS gérée suivante aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

JSON

```
{
     "Version":"2012-10-17",
     "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                 "iotanalytics:BatchPutMessage",
                 "iotanalytics:CancelPipelineReprocessing",
                 "iotanalytics:CreateChannel",
                 "iotanalytics:CreateDataset",
```

```
                "iotanalytics:CreateDatasetContent",
                "iotanalytics:CreateDatastore",
                "iotanalytics:CreatePipeline",
                "iotanalytics:DeleteChannel",
                "iotanalytics:DeleteDataset",
                "iotanalytics:DeleteDatasetContent",
                "iotanalytics:DeleteDatastore",
                "iotanalytics:DeletePipeline",
                "iotanalytics:DescribeChannel",
                "iotanalytics:DescribeDataset",
                "iotanalytics:DescribeDatastore",
                "iotanalytics:DescribeLoggingOptions",
                "iotanalytics:DescribePipeline",
                "iotanalytics:GetDatasetContent",
                "iotanalytics:ListChannels",
                "iotanalytics:ListDatasetContents",
                "iotanalytics:ListDatasets",
                "iotanalytics:ListDatastores",
                "iotanalytics:ListPipelines",
                "iotanalytics:ListTagsForResource",
                "iotanalytics:PutLoggingOptions",
                "iotanalytics:RunPipelineActivity",
                "iotanalytics:SampleChannelData",
                "iotanalytics:StartPipelineReprocessing",
                "iotanalytics:TagResource",
                "iotanalytics:UntagResource",
                "iotanalytics:UpdateChannel",
                "iotanalytics:UpdateDataset",
                "iotanalytics:UpdateDatastore",
                "iotanalytics:UpdatePipeline"
            ],
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:channel/your-channel-name",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:dataset/your-datasetName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:datastore/your-datastoreName",
            "Resource": "arn:aws:iotanalytics:us-
east-1:123456789012:pipeline/your-pipelineName"
        }
    ]
 }
```

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous tentez d'effectuer.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une stratégie qui permet aux utilisateurs d'afficher les stratégies 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 .

JSON

```
{
    "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/username"
            ]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
```

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

Accès à une AWS IoT Analytics entrée

Dans cet exemple, vous souhaitez accorder à un utilisateur l' Compte AWS accès à l'une de vos AWS IoT Analytics chaînes,`exampleChannel`. Vous souhaitez également autoriser l'utilisateur à ajouter, mettre à jour et supprimer des chaînes.

La politique accorde les `iotanalytics:ListChannels, iotanalytics:DescribeChannel, iotanalytics:CreateChannel, iotanalytics:DeleteChannel, and iotanalytics:UpdateChannel` autorisations à l'utilisateur. Pour un exemple de procédure pas à pas pour le service Amazon S3 qui accorde des autorisations aux utilisateurs et les teste à l'aide de la console, consultez Un exemple de procédure pas à pas : utilisation de politiques utilisateur pour contrôler l'accès à votre compartiment.

JSON

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ListChannelsInConsole",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:ListChannels"
         ],
         "Resource":"arn:aws:iotanalytics:*:*:*"
      },
      {
         "Sid":"ViewSpecificChannelInfo",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:DescribeChannel"
         ],
         "Resource":"arn:aws:iotanalytics:*:*:exampleChannel"
      },
```

```
      {
         "Sid":"ManageChannels",
         "Effect":"Allow",
         "Action":[
            "iotanalytics:CreateChannel",
            "iotanalytics:DeleteChannel",
            "iotanalytics:DescribeChannel",
            "iotanalytics:ListChannels",
            "iotanalytics:UpdateChannel"
         ],
         "Resource":"arn:aws:iotanalytics:*:*:exampleChannel/*"
      }
   ]
}
```

Affichage des AWS IoT Analytics chaînes en fonction des tags

Vous pouvez utiliser des conditions dans votre politique basée sur l'identité pour contrôler l'accès aux AWS IoT Analytics ressources en fonction de balises. Cet exemple montre comment créer une stratégie qui permet d'afficher un élément `channel`. Toutefois, les autorisations ne sont accordées que si la `channel` balise `Owner` a la valeur du nom d'utilisateur de cet utilisateur. Cette stratégie accorde également les autorisations nécessaires pour réaliser cette action sur la console.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "ListChannelsInConsole",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "*"
        },
        {
            "Sid": "ViewChannelsIfOwner",
            "Effect": "Allow",
            "Action": "iotanalytics:ListChannels",
            "Resource": "arn:aws:iotanalytics:*:*:channel/*",
            "Condition": {
```

```
                "StringEquals": {"iotanalytics:ResourceTag/Owner":
 "${aws:username}"}
            }
        }
    ]
}
```

Vous pouvez attacher cette stratégie aux utilisateurs de votre compte. Si un utilisateur nommé `richard-roe` tente de consulter un AWS IoT Analytics `channel`, celui-ci `channel` doit être balisé`Owner=richard-roe or owner=richard-roe`. Dans le cas contraire, l'utilisateur se voit refuser l'accès. La clé de condition de balise `Owner` correspond à la fois à `Owner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.

Résolution des problèmes AWS IoT Analytics d'identité et d'accès

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pourriez rencontrer lors de votre utilisation AWS IoT Analytics.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS IoT Analytics
- Je ne suis pas autorisé à effectuer iam:PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS IoT Analytics ressources

Je ne suis pas autorisé à effectuer une action dans AWS IoT Analytics

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'`mateojacksonutilisateur` essaie d'utiliser la console pour afficher les détails d'un `channel` mais ne dispose pas des `iotanalytics:ListChannels` autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotanalytics:``ListChannels`` on resource: ``my-example-channel``
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la `my-example-channel` ressource à l'aide de l'`iotanalytics:ListChannel`action.

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 Analytics.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, vous devez disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS IoT Analytics. Toutefois, l'action nécessite que le service ait des autorisations accordées par une fonction de service. Mary n'est pas autorisée à transmettre le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dans ce cas, les politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes AWS IoT Analytics 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 (ACL), 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 Analytics en charge, consultez la section AWS IoT Analytics Fonctionnement 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 découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

Connexion et surveillance AWS IoT Analytics

AWS fournit des outils que vous pouvez utiliser pour surveiller AWS IoT Analytics. Vous pouvez configurer certains de ces outils afin qu'ils effectuent la surveillance à votre place. Une intervention manuelle est nécessaire pour certains outils. Nous vous recommandons d'automatiser le plus possible les tâches de supervision.

Outils de surveillance automatique

Vous pouvez utiliser les outils de surveillance automatique suivants pour surveiller AWS IoT et signaler tout problème :

- Amazon CloudWatch Logs : surveillez, stockez et accédez à vos fichiers journaux depuis AWS CloudTrail ou d'autres sources. Pour plus d'informations, consultez la section Qu'est-ce que les fichiers journaux de AWS CloudTrail surveillance dans le guide de CloudWatch l'utilisateur Amazon.
- AWS CloudTrail surveillance des journaux - Partagez les fichiers journaux entre les comptes, surveillez les fichiers CloudTrail journaux en temps réel en les envoyant à CloudWatch Logs, écrivez des applications de traitement des journaux en Java et vérifiez que vos fichiers journaux n'ont pas changé après leur livraison par. CloudTrail Pour plus d'informations, consultez la section Utilisation des fichiers CloudTrail journaux dans le Guide de AWS CloudTrail l'utilisateur.

Outils de surveillance manuelle

Une autre partie importante de la surveillance AWS IoT consiste à surveiller manuellement les éléments non couverts par les CloudWatch alarmes. Le tableau de bord AWS IoT CloudWatch, et les autres tableaux de bord de la console de AWS service, fournissent une at-a-glance vue d'ensemble de l'état de votre AWS environnement. Nous vous recommandons de vérifier également les fichiers journaux AWS IoT Analytics.

- La AWS IoT Analytics console affiche :
  - Canaux
  - Pipelines
  - Magasins de données
  - Ensembles de données
  - Blocs-notes
  - Settings
  - Apprendre
- La page d' CloudWatch accueil indique :
  - Alarmes et statuts en cours
  - Graphiques des alarmes et des ressources
  - Statut d'intégrité du service

  En outre, vous pouvez utiliser CloudWatch pour effectuer les opérations suivantes :
  - Créer des tableaux de bord personnalisés pour surveiller les services de votre choix
  - Représenter graphiquement les données de métriques pour résoudre les problèmes et découvrir les tendances
  - Recherchez et parcourez tous les indicateurs de vos AWS ressources
  - Créer et modifier des alarmes pour être informé des problèmes

Surveillance avec Amazon CloudWatch Logs

AWS IoT Analytics prend en charge la journalisation avec Amazon CloudWatch. Vous pouvez activer et configurer la CloudWatch journalisation Amazon pour AWS IoT Analytics en utilisant l'opération `PutLoggingOptions` d'API. Cette section décrit comment vous pouvez utiliser

`PutLoggingOptions` with Gestion des identités et des accès AWS (IAM) pour configurer et activer la CloudWatch journalisation Amazon pour AWS IoT Analytics.

Pour plus d'informations sur CloudWatch les journaux, consultez le guide de l'utilisateur Amazon CloudWatch Logs. Pour plus d'informations sur AWS IAM, consultez le guide de l'Gestion des identités et des accès AWS utilisateur.

> **Note**
>
> Avant d'activer la AWS IoT Analytics journalisation, assurez-vous de bien comprendre les autorisations d'accès aux CloudWatch journaux. Les utilisateurs ayant accès aux CloudWatch journaux peuvent voir vos informations de débogage. Pour plus d'informations, consultez Authentification et contrôle d'accès pour Amazon CloudWatch Logs.

Créez un rôle IAM pour activer la journalisation

Pour créer un rôle IAM afin d'activer la journalisation pour Amazon CloudWatch

1. Utilisez la console AWS IAM ou la commande AWS IAM CLI suivante pour créer un nouveau rôle IAM avec une politique de relation de confiance (politique de confiance). CreateRole La politique de confiance accorde à une entité, telle qu'Amazon CloudWatch, l'autorisation d'assumer ce rôle.

   ```
   aws iam create-role --role-name exampleRoleName --assume-role-policy-document
         exampleTrustPolicy.json
   ```

   Le fichier `exampleTrustPolicy.json` contient le contenu suivant.

   > **Note**
   >
   > Cet exemple inclut une clé de contexte de condition globale pour se protéger contre le problème confus de sécurité adjoint. *123456789012*Remplacez-le par votre numéro de AWS compte et *aws-region* par la AWS région de vos AWS ressources. Pour de plus amples informations, veuillez consulter the section called “Prévention du cas de figure de l’adjoint désorienté entre services”.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                },
                "ArnLike": {
                "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:*"
                }
            }
        }
    ]
}
```

Vous utiliserez l'ARN de ce rôle ultérieurement lorsque vous appellerez la AWS IoT Analytics `PutLoggingOptions` commande.

2. Utilisez AWS IAM PutRolePolicypour associer une politique d'autorisation (`arole policy`) au rôle que vous avez créé à l'étape 1.

   ```
   aws iam put-role-policy --role-name exampleRoleName --policy-name
   examplePolicyName --policy-document exampleRolePolicy.json
   ```

   Le fichier exampleRolePolicy .json contient le contenu suivant.

   JSON

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

```
    "Statement": [
     {
      "Effect": "Allow",
      "Action": [
       "logs:CreateLogGroup",
       "logs:CreateLogStream"
      ],
      "Resource": [
       "arn:aws:logs:*:*:*"
      ]
     }
    ]
}
```

3. Pour AWS IoT Analytics autoriser la mise en ligne des événements de journalisation sur Amazon CloudWatch, utilisez la CloudWatch commande Amazon PutResourcePolicy.

> **Note**
>
> Pour éviter le problème confus lié à la sécurité des adjoints, nous vous recommandons de le spécifier `aws:SourceArn` dans votre politique de ressources. Cela restreint l'accès pour autoriser uniquement les demandes provenant d'un compte spécifique. Pour de plus amples informations sur le problème de l'adjoint confus, veuillez consulter the section called "Prévention du cas de figure de l'adjoint désorienté entre services".

```
aws logs put-resource-policy --policy-in-json
exampleResourcePolicy.json
```

Le `exampleResourcePolicy.json` fichier contient la politique de ressources suivante.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iotanalytics.amazonaws.com"
            },
```

```
            "Action": "logs:PutLogEvents",
            "Resource": "*",
            "Condition": {
                "ArnLike": {
                "aws:SourceArn": "arn:aws:iotanalytics:us-
east-1:123456789012:*/*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}
```

Configurer et activer la journalisation

Utilisez la `PutLoggingOptions` commande pour configurer et activer la CloudWatch journalisation Amazon pour AWS IoT Analytics. La valeur `roleArn` dans le champ `loggingOptions` doit correspondre à l'ARN du rôle que vous avez créé dans la section précédente. Vous pouvez également utiliser la commande `DecribeLoggingOptions` pour vérifier les paramètres de vos options de journalisation.

PutLoggingOptions

Définit ou met à jour les options de AWS IoT Analytics journalisation. Si vous mettez à jour la valeur d'un `loggingOptions` champ, il faut jusqu'à une minute pour que la modification prenne effet. En outre, si vous modifiez la politique associée au rôle que vous avez spécifié dans le `roleArn` champ (par exemple, pour corriger une politique non valide), la prise en compte de cette modification peut prendre jusqu'à cinq minutes. Pour de plus amples informations, veuillez consulter `PutLoggingOptions`.

DescribeLoggingOptions

Récupère les paramètres actuels des options de AWS IoT Analytics journalisation. Pour de plus amples informations, consultez `DescribeLoggingOptions`.

Espace de noms, métriques et dimensions

AWS IoT Analytics place les métriques suivantes dans le CloudWatch référentiel Amazon :

Namespace
AWS/E/S TAnalytics

Métrique	Description
ActionExecution	Le nombre d'actions exécutées.
ActionExecutionThrottled	Le nombre d'actions limitées.
ActivityExecutionError	Nombre d'erreurs générées pendant l'exécution de l'activité de pipeline.
IncomingMessages	Nombre de messages qui arrivent dans le canal.
PipelineConcurrentExecutionCount	Le nombre d'activités du pipeline exécutées simultanément.

Dimension	Description
ActionType	Type d'action qui est surveillée.
ChannelName	Nom du canal surveillé.
DatasetName	Le nom de l'ensemble de données qui est surveillé.
DatastoreName	Nom du magasin de données surveillé.
PipelineActivityName	Nom de l'activité de pipeline surveillée.
PipelineActivityType	Type de l'activité de pipeline surveillée.
PipelineName	Nom du pipeline surveillé.

Surveillez avec Amazon CloudWatch Events

AWS IoT Analytics publie automatiquement un événement sur Amazon CloudWatch Events lorsqu'une erreur d'exécution se produit au cours d'une AWS Lambda activité. Cet événement contient un message d'erreur détaillé et les clés des objets Amazon Simple Storage Service (Amazon S3) qui stockent les messages de canal non traités. Vous pouvez utiliser les clés Amazon S3 pour retraiter les messages de canal non traités. Pour plus d'informationsRetraitement des messages du canal, consultez l'StartPipelineReprocessingAPI dans la référence des AWS IoT Analytics API et What Is Amazon CloudWatch Events dans le guide de l'utilisateur Amazon CloudWatch Events.

Vous pouvez également configurer des cibles qui permettent à Amazon CloudWatch Events d'envoyer des notifications ou de prendre d'autres mesures. Par exemple, vous pouvez envoyer la notification à une file d'attente Amazon Simple Queue Service (Amazon SQS), puis appeler `StartReprocessingMessage` l'API pour traiter les messages de canal enregistrés dans les objets Amazon S3. Amazon CloudWatch Events prend en charge de nombreux types de cibles, tels que les suivants :

- Amazon Kinesis Streams
- AWS Lambda fonctions
- Rubriques Amazon Simple Notification Service (Amazon SNS)
- Files d'attente Amazon Simple Queue Service (Amazon SQS)

Pour obtenir la liste des cibles prises en charge, consultez Amazon EventBridge Targets dans le guide de EventBridge l'utilisateur Amazon.

Vos ressources d' CloudWatch événements et les cibles associées doivent se trouver dans la AWS région où vous avez créé vos AWS IoT Analytics ressources. Pour plus d'informations, consultez la section Points de terminaison et quotas du service dans le Références générales AWS.

La notification envoyée à Amazon CloudWatch Events pour les erreurs d'exécution liées à l' AWS Lambda activité utilise le format suivant.

```
{
    "version": "version-id",
    "id": "event-id",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "aws-account",
```

```
    "time": "timestamp",
    "region": "aws-region",
    "resources": [
        "pipeline-arn"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "pipeline-name",
        "error-code": "LAMBDA_FAILURE",
        "message": "error-message",
        "channel-messages": {
            "s3paths": [
                "s3-keys"
            ]
        },
        "activity-name": "lambda-activity-name",
        "lambda-function-arn": "lambda-function-arn"
    }
}
```

Exemple de notification :

```
{
    "version": "0",
    "id": "204e672e-ef12-09af-4cfd-de3b53673ec6",
    "detail-type": "IoT Analytics Pipeline Failure Notification",
    "source": "aws.iotanalytics",
    "account": "123456789012",
    "time": "2020-10-15T23:47:02Z",
    "region": "ap-southeast-2",
    "resources": [
        "arn:aws:iotanalytics:ap-southeast-2:123456789012:pipeline/
test_pipeline_failure"
    ],
    "detail": {
        "event-detail-version": "1.0",
        "pipeline-name": "test_pipeline_failure",
        "error-code": "LAMBDA_FAILURE",
        "message": "Temp unavaliable",
        "channel-messages": {
        "s3paths": [
            "test_pipeline_failure/channel/cmr_channel/__dt=2020-10-15
 00:00:00/1602805530000_1602805560000_123456789012_cmr_channel_0_257.0.json.gz"
```

```
        ]
    },
    "activity-name": "LambdaActivity_33",
    "lambda-function-arn": "arn:aws:lambda:ap-
southeast-2:123456789012:function:lambda_activity"
    }
}
```

Recevoir des notifications de données en retard via Amazon CloudWatch Events

Lorsque vous créez le contenu d'un ensemble de données à partir de données issues d'une période spécifiée, certaines données peuvent ne pas arriver à temps pour être traitées. Pour permettre un délai, vous pouvez spécifier un `deltaTime` décalage `QueryFilter` lors de la création d'un ensemble de données en appliquant une `queryAction` (une requête SQL). AWS IoT Analytics traite toujours les données qui arrivent dans le temps delta, et le contenu de votre ensemble de données est décalé dans le temps. La fonctionnalité de notification tardive des données permet AWS IoT Analytics d'envoyer des notifications via Amazon CloudWatch Events lorsque les données arrivent après l'heure delta.

Vous pouvez utiliser la AWS IoT Analytics console, l'API AWS Command Line Interface (AWS CLI) ou le AWS SDK pour définir les règles relatives aux données tardives pour un ensemble de données.

Dans l' AWS IoT Analytics API, l'`LateDataRuleConfiguration`objet représente les derniers paramètres des règles de données d'un ensemble de données. Cet objet fait partie de l'`Dataset`objet associé aux opérations `CreateDataset` et à `UpdateDataset` l'API.

Paramètres

Lorsque vous créez une règle de données en retard pour un ensemble de données avec AWS IoT Analytics, vous devez spécifier les informations suivantes :

`ruleConfiguration` (**`LateDataRuleConfiguration`**)

Structure contenant les informations de configuration d'une règle de données tardive.

`deltaTimeSessionWindowConfiguration`

Structure qui contient les informations de configuration d'une fenêtre de session de temps delta.

DeltaTime spécifie un intervalle de temps. Vous pouvez utiliser `DeltaTime` pour créer un contenu d'ensemble de données avec des données arrivées dans le magasin de données

depuis la dernière exécution. Pour un exemple`DeltaTime`, voir Création d'un jeu de données SQL avec une fenêtre delta (CLI).

`timeoutInMinutes`

Intervalle de temps. Vous pouvez l'utiliser `timeoutInMinutes` pour regrouper les notifications de données en retard générées depuis la dernière exécution. AWS IoT Analytics AWS IoT Analytics envoie un lot de notifications à CloudWatch Events à la fois.

Type : entier

Plage valide : 1 à 60

`ruleName`

Nom de la règle de données tardives.

Type : String

> **Important**
>
> Pour le spécifier`lateDataRules`, l'ensemble de données doit utiliser un `DeltaTime` filtre.

Configuration des règles relatives aux données tardives (console)

La procédure suivante explique comment configurer la règle des données tardives d'un ensemble de données dans la AWS IoT Analytics console.

Pour configurer les règles relatives aux données tardives

1. Connectez-vous à la console AWS IoT Analytics.
2. Dans le volet de navigation, sélectionnez Ensembles de données.
3. Sous Ensembles de données, choisissez le jeu de données cible.
4. Dans le volet de navigation, sélectionnez Détails.
5. Dans la section de la fenêtre Delta, choisissez Modifier.
6. Sous Configurer le filtre de sélection des données, procédez comme suit :
   a. Pour la fenêtre de sélection des données, choisissez Delta time.
   b. Pour Décalage, entrez une période, puis choisissez une unité.

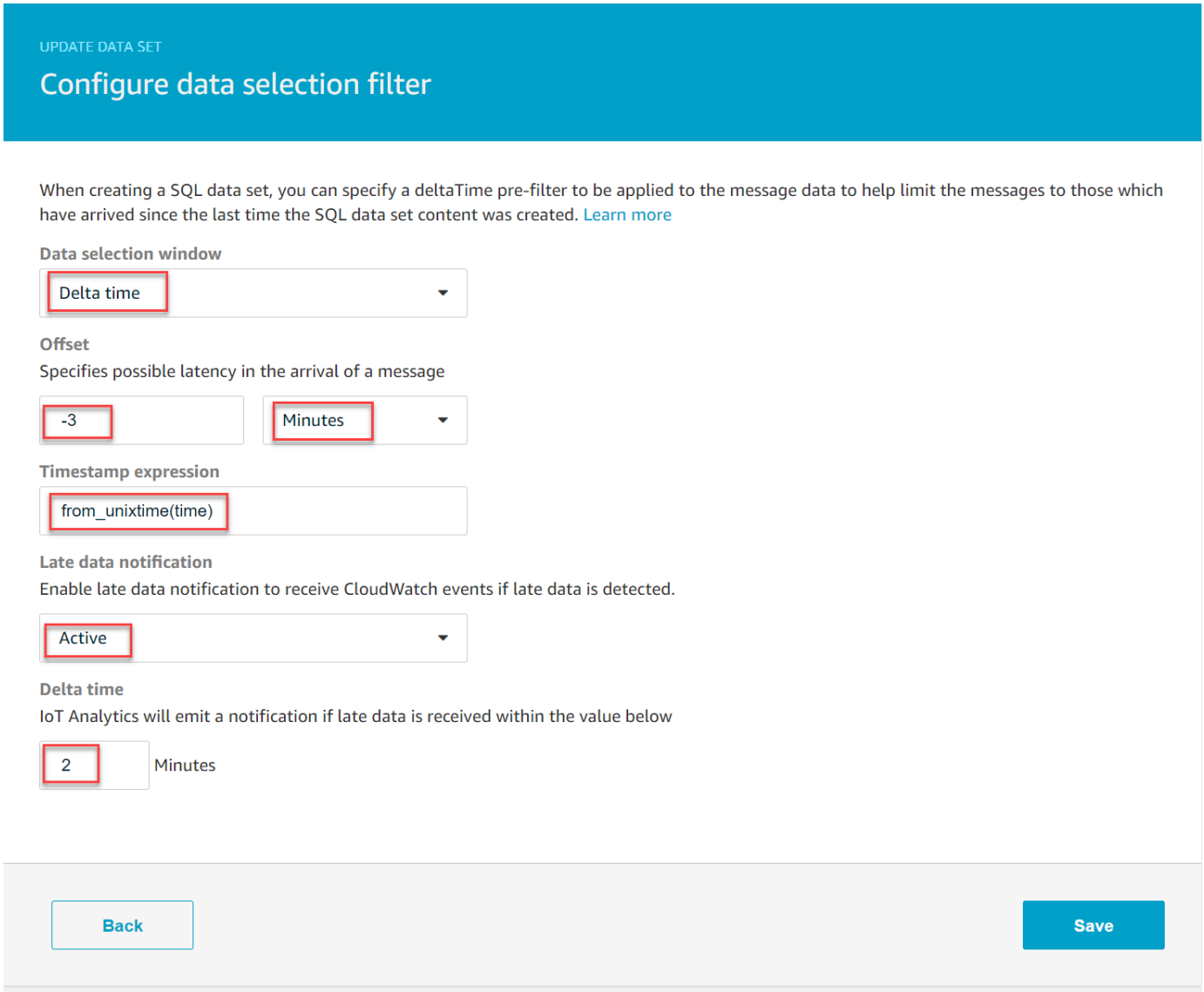
c. Pour Expression d'horodatage, entrez une expression. Il peut s'agir du nom d'un champ d'horodatage ou d'une expression SQL permettant de déduire l'heure, par exemple. `from_unixtime(time)`

   Pour plus d'informations sur la façon d'écrire une expression d'horodatage, voir Fonctions et opérateurs de date et d'heure dans la documentation de Presto 0.172.

d. Pour la notification tardive des données, sélectionnez Actif.

e. Pour l'heure Delta, entrez un entier. La plage valide est comprise entre 1 et 60.

f. Choisissez Enregistrer.



Configuration des règles relatives aux données tardives (CLI)

Dans l' AWS IoT Analytics API, l'`LateDataRuleConfiguration`objet représente les derniers paramètres des règles de données d'un ensemble de données. Cet objet fait partie de l'`Dataset`objet associé à `CreateDataset` et`UpdateDataset`. Vous pouvez utiliser l'API ou le AWS SDK pour définir les règles relatives aux données tardives pour un ensemble de données. AWS CLI L'exemple suivant repose sur AWS CLI.

Pour créer votre ensemble de données avec des règles de données tardives spécifiées, exécutez la commande suivante. La commande suppose que le `dataset.json` fichier se trouve dans le répertoire en cours.

> **Note**
>
> Vous pouvez utiliser l'UpdateDatasetAPI pour mettre à jour un ensemble de données existant.

```
aws iotanalytics create-dataset --cli-input-json file://dataset.json
```

Le `dataset.json` fichier doit contenir les éléments suivants :

- Remplacez *`demo_dataset`* par le nom du jeu de données cible.
- Remplacez *`demo_datastore`* par le nom du magasin de données cible.
- *`from_unixtime(time)`*Remplacez-le par le nom d'un champ d'horodatage ou d'une expression SQL permettant de dériver l'heure.

  Pour plus d'informations sur la façon d'écrire une expression d'horodatage, voir Fonctions et opérateurs de date et d'heure dans la documentation de Presto 0.172.
- Remplacez *`timeout`* par un entier compris entre 1 et 60.
- Remplacez *`demo_rule`* par n'importe quel nom.

```
{
    "datasetName": "demo_dataset",
    "actions": [
        {
            "actionName": "myDatasetAction",
```

```
            "queryAction": {
                "filters": [
                    {
                        "deltaTime": {
                            "offsetSeconds": -180,
                            "timeExpression": "from_unixtime(time)"
                        }
                    }
                ],
                "sqlQuery": "SELECT * FROM demo_datastore"
            }
        }
    ],
    "retentionPeriod": {
        "unlimited": false,
        "numberOfDays": 90
    },
    "lateDataRules": [
        {
            "ruleConfiguration": {
                "deltaTimeSessionWindowConfiguration": {
                    "timeoutInMinutes": timeout
                }
            },
            "ruleName": "demo_rule"
        }
    ]
}
```

S'abonner pour recevoir des notifications de données tardives

Vous pouvez créer des règles dans CloudWatch Events qui définissent comment traiter les notifications de données en retard envoyées depuis AWS IoT Analytics. Lorsque CloudWatch Events reçoit les notifications, il invoque les actions cibles spécifiées dans vos règles.

Conditions préalables à la création de règles relatives CloudWatch aux événements

Avant de créer une règle d' CloudWatch événements pour AWS IoT Analytics, vous devez effectuer les opérations suivantes :

- Familiarisez-vous avec les événements, les règles et les cibles dans CloudWatch Événements.
- Créez et configurez les cibles invoquées par vos règles d' CloudWatch événements. Les règles peuvent invoquer de nombreux types de cibles, tels que les suivants :

- Amazon Kinesis Streams
- AWS Lambda fonctions
- Rubriques Amazon Simple Notification Service (Amazon SNS)
- Files d'attente Amazon Simple Queue Service (Amazon SQS)

Votre règle relative aux CloudWatch événements et les cibles associées doivent se trouver dans la AWS région où vous avez créé vos AWS IoT Analytics ressources. Pour plus d'informations, consultez la section Points de terminaison et quotas du service dans le Références générales AWS.

Pour plus d'informations, voir Qu'est-ce que CloudWatch les événements ? et Getting started with Amazon CloudWatch Events dans le guide de l'utilisateur Amazon CloudWatch Events.

Événement de notification de données tardif

L'événement relatif aux notifications de données tardives utilise le format suivant.

```
{
 "version": "0",
 "id": "7f51dfa7-ffef-97a5-c625-abddbac5eadd",
 "detail-type": "IoT Analytics Dataset Lifecycle Notification",
 "source": "aws.iotanalytics",
 "account": "123456789012",
 "time": "2020-05-14T02:38:46Z",
 "region": "us-east-2",
 "resources": ["arn:aws:iotanalytics:us-east-2:123456789012:dataset/demo_dataset"],
 "detail": {
  "event-detail-version": "1.0",
  "dataset-name": "demo_dataset",
  "late-data-rule-name": "demo_rule",
  "version-ids": ["78244852-8737-4650-aa4d-3071a01338fa"],
  "message": null
 }
}
```

Créez une règle d' CloudWatch événements pour recevoir des notifications de données en retard

La procédure suivante explique comment créer une règle qui envoie des notifications de données AWS IoT Analytics en retard à une file d'attente Amazon SQS.

Pour créer une règle d' CloudWatch événements

1. Connectez-vous à la CloudWatchconsole Amazon.
2. Sous Events (Événements) dans le panneau de navigation, choisissez Rules (Règles).
3. Sur la page Règles, choisissez Créer une règle.
4. Sous Source d'événement, choisissez Event Pattern.
5. Dans la section Créer un modèle d'événement correspondant aux événements par service, procédez comme suit :
   a. Pour le nom du service, choisissez IoT Analytics
   b. Pour Type d'événement, choisissez IoT Analytics Dataset Lifecycle Notification.
   c. Choisissez Nom (s) de jeu de données spécifique, puis entrez le nom du jeu de données cible.
6. Sous Cibles, choisissez Ajouter une cible*.
7. Choisissez la file d'attente SQS, puis procédez comme suit :
   - Pour Queue*, choisissez la file d'attente cible.
8. Choisissez Configure details (Configurer les détails).
9. Sur la page Étape 2 : Configuration des détails des règles, entrez un nom et une description.
10. Choisissez Créer une règle.

Journalisation des appels d' AWS IoT Analytics API avec AWS CloudTrail

AWS IoT Analytics 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 Analytics. CloudTrail capture un sous-ensemble d'appels d'API sous AWS IoT Analytics forme d'événements, y compris les appels provenant de la AWS IoT Analytics console et les appels de code adressés au AWS IoT Analytics APIs. 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 Analytics. 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 Analytics, 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 en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS IoT Analytics informations dans AWS CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans AWS IoT Analytics, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS 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 Analytics, 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 . 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 en savoir plus, consultez :

- Présentation de la création d'un journal d'activité
- 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

AWS IoT Analytics prend en charge l'enregistrement des actions suivantes sous forme d'événements dans les fichiers CloudTrail journaux :

- CancelPipelineReprocessing
- CreateChannel
- CreateDataset
- CreateDatasetContent
- CreateDatastore
- CreatePipeline
- DeleteChannel
- DeleteDataset
- DeleteDatasetContent

- DeleteDatastore
- DeletePipeline
- DescribeChannel
- DescribeDataset
- DescribeDatastore
- DescribeLoggingOptions
- DescribePipeline
- GetDatasetContent
- ListChannels
- ListDatasets
- ListDatastores
- ListPipelines
- PutLoggingOptions
- RunPipelineActivity
- SampleChannelData
- StartPipelineReprocessing
- UpdateChannel
- UpdateDataset
- UpdateDatastore
- UpdatePipeline

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 Gestion des identités et des accès AWS d'identification root ou utilisateur.
- 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 .

Comprendre les entrées du fichier AWS IoT Analytics journal

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment 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'action demandée, la date et l'heure de l'action, 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'`CreateChannel`action.

```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsChannelTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsChannelTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:43:12Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:55:14Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateChannel",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"channelName": "channel_channeltest"
},
```

```
"responseElements": {
"retentionPeriod": {
"unlimited": true
},
"channelName": "channel_channeltest",
"channelArn": "arn:aws:iotanalytics:us-east-1:123456789012:channel/channel_channeltest"
},
"requestID": "7f871429-11e2-11e8-9eee-0781b5c0ac59",
"eventID": "17885899-6977-41be-a6a0-74bb95a78294",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateDataset`action.

```
{
"eventVersion": "1.05",
"userIdentity": {
"type": "AssumedRole",
"principalId": "ABCDE12345FGHIJ67890B:AnalyticsDatasetTestFunction",
"arn": "arn:aws:sts::123456789012:assumed-role/AnalyticsRole/
AnalyticsDatasetTestFunction",
"accountId": "123456789012",
"accessKeyId": "ABCDE12345FGHIJ67890B",
"sessionContext": {
"attributes": {
 "mfaAuthenticated": "false",
 "creationDate": "2018-02-14T23:41:36Z"
},
"sessionIssuer": {
 "type": "Role",
 "principalId": "ABCDE12345FGHIJ67890B",
 "arn": "arn:aws:iam::123456789012:role/AnalyticsRole",
 "accountId": "123456789012",
 "userName": "AnalyticsRole"
}
}
},
"eventTime": "2018-02-14T23:53:39Z",
"eventSource": "iotanalytics.amazonaws.com",
"eventName": "CreateDataset",
"awsRegion": "us-east-1",
"sourceIPAddress": "198.162.1.0",
```

```
"userAgent": "aws-internal/3 exec-env/AWS_Lambda_java8",
"requestParameters": {
"datasetName": "dataset_datasettest"
},
"responseElements": {
"datasetArn": "arn:aws:iotanalytics:us-east-1:123456789012:dataset/
dataset_datasettest",
"datasetName": "dataset_datasettest"
},
"requestID": "46ee8dd9-11e2-11e8-979a-6198b668c3f0",
"eventID": "5abe21f6-ee1a-48ef-afc5-c77211235303",
"eventType": "AwsApiCall",
"recipientAccountId": "123456789012"
}
```


Validation de conformité pour AWS IoT Analytics

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. Pour plus d'informations sur votre responsabilité en matière de conformité lors de l'utilisation Services AWS, consultez AWS la documentation de sécurité.

Résilience dans AWS IoT Analytics

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. AWS Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, 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 entre les zones de disponibilité sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS globale.

Sécurité de l'infrastructure dans AWS IoT Analytics

En tant que service géré, AWS IoT Analytics 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 Analytics 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.

AWS IoT Analytics quotas

Le Références générales AWS Guide fournit les quotas par défaut AWS IoT Analytics pour un AWS compte. Sauf indication contraire, chaque quota est établi par AWS région. Pour plus d'informations, consultez les sections AWS IoT Analytics Points de terminaison, quotas et quotas AWS de service dans le Références générales AWS Guide.

Pour demander une augmentation du quota de service, soumettez un dossier d'assistance dans la console du centre de support. Pour de plus amples informations, veuillez consulter Demande d'augmentation de quota dans le Guide de l'utilisateur Service Quotas.

AWS IoT Analytics commandes

Lisez cette rubrique pour en savoir plus sur les opérations d'API pour les protocoles de services Web pris en charge AWS IoT Analytics, notamment les exemples de demandes, de réponses et d'erreurs.

AWS IoT Analytics actions

Vous pouvez utiliser les commandes d' AWS IoT Analytics API pour collecter, traiter, stocker et analyser vos données IoT. Pour plus d'informations, consultez les actions prises en charge AWS IoT Analytics dans la référence AWS IoT Analytics d'API.

Les AWS IoT Analytics sections de la référence des AWS CLI commandes incluent les AWS CLI commandes que vous pouvez utiliser pour administrer et manipuler AWS IoT Analytics.

AWS IoT Analytics données

Vous pouvez utiliser les commandes de l'API de AWS IoT Analytics données pour effectuer des activités avancées avec AWS IoT Analytics `channelpipeline`,`datastore`, et`dataset`. Pour plus d'informations, consultez les types de données pris en charge par AWS IoT Analytics Data dans la référence AWS IoT Analytics d'API.

Résolution des problèmes AWS IoT Analytics

Consultez la section suivante pour résoudre les erreurs et trouver des solutions possibles pour résoudre les problèmes liés à AWS IoT Analytics.

Rubriques

- Comment savoir si mes messages arrivent AWS IoT Analytics ?
- Pourquoi mon pipeline perd-il des messages ? Comment puis-je résoudre ce problème ?
- Pourquoi n'y a-t-il aucune donnée dans mon magasin de données ?
- Pourquoi mon jeu de données s'affiche-t-il simplement __dt ?
- Comment coder un événement provoqué par la complétion de l'ensemble de données ?
- Comment configurer correctement mon instance de bloc-notes à utiliser AWS IoT Analytics ?
- Pourquoi ne puis-je pas créer de blocs-notes dans une instance ?
- Pourquoi est-ce que je ne vois pas mes ensembles de données dans Quick Suite ?
- Pourquoi le bouton de conteneurisation ne s'affiche-t-il pas sur mon bloc-notes Jupyter existant ?
- Pourquoi l'installation de mon plugin de conteneurisation échoue-t-elle ?
- Pourquoi mon plugin de conteneurisation génère-t-il une erreur ?
- Pourquoi est-ce que je ne vois pas mes variables pendant la conteneurisation ?
- Quelles variables puis-je ajouter à mon conteneur en tant qu'entrée ?
- Comment définir la sortie de mon conteneur comme entrée pour une analyse ultérieure ?
- Pourquoi mon jeu de données de conteneurs échoue-t-il ?

Comment savoir si mes messages arrivent AWS IoT Analytics ?

Vérifiez si la règle d'injection de données dans le canal via le moteur de règles est correctement configurée.

```
aws iot get-topic-rule --rule-name your-rule-name
```

La réponse doit ressembler à ce qui suit.

```
{
```

```
    "ruleArn": "arn:aws:iot:us-west-2:your-account-id:rule/your-rule-name",
    "rule": {
        "awsIotSqlVersion": "2016-03-23",
        "sql": "SELECT * FROM 'iot/your-rule-name'",
        "ruleDisabled": false,
        "actions": [
            {
                "iotAnalytics": {
                    "channelArn":
 "arn:aws:iotanalytics:region:your_account_id:channel/your-channel-name"
                }
            }
        ],
        "ruleName": "your-rule-name"
    }
}
```

Assurez-vous que la région et le nom du canal utilisés dans la règle sont corrects. Pour vous assurer que vos données atteignent le moteur de règles et que celles-ci sont exécutées correctement, vous pouvez ajouter une nouvelle cible pour stocker temporairement les messages entrants dans le compartiment Amazon S3.

Pourquoi mon pipeline perd-il des messages ? Comment puis-je résoudre ce problème ?

- Une activité a reçu une entrée JSON non valide :

  Toutes les activités, à l'exception des activités Lambda, nécessitent spécialement une chaîne JSON valide en entrée. Si le fichier JSON reçu par une activité n'est pas valide, le message est supprimé et n'atteint pas le magasin de données. Vérifiez que vous ingérez des messages JSON valides dans le service. Dans le cas d'une entrée binaire, assurez-vous que la première activité de votre pipeline est une activité Lambda qui convertit les données binaires en un code JSON valide avant de les transmettre à l'activité suivante ou de les stocker dans le magasin de données. Pour plus d'informations, consultez l'exemple 2 de la fonction Lambda.

- Une fonction Lambda appelée par une activité Lambda dispose d'autorisations insuffisantes :

  Assurez-vous que chaque fonction Lambda d'une activité Lambda est autorisée à être invoquée depuis le service. AWS IoT Analytics Vous pouvez utiliser la AWS CLI commande suivante pour accorder une autorisation.

```
aws lambda add-permission --function-name <name> --region <region> --statement-id
 <id> --principal iotanalytics.amazonaws.com --action lambda:InvokeFunction
```

- Un filtre ou l'activité removeAttribute est mal défini :

  Assurez-vous que les définitions `filter` ou les `removeAttribute` activités sont correctes, le cas échéant. Si vous filtrez un message ou que vous supprimez tous les attributs d'un message, ce message n'est pas ajouté au magasin de données.

Pourquoi n'y a-t-il aucune donnée dans mon magasin de données ?

- Il existe un décalage entre l'ingestion des données et leur disponibilité :

  Cela peut prendre plusieurs minutes après que des données ont été ingérées dans un canal avant que ces données soient disponibles dans le magasin de données. Ce délai varie en fonction du nombre d'activités de pipeline et de la définition des activités lambda personnalisées de votre pipeline.
- Des messages sont filtrés dans votre pipeline :

  Assurez-vous que vous ne supprimez pas des messages dans le pipeline. (Consultez la question et la réponse précédentes.)
- Votre requête de jeu de données est incorrecte :

  Assurez-vous que la requête qui génère l'ensemble de données à partir du magasin de données est correcte. Supprimez les filtres inutiles de la requête pour vous assurer que les données atteignent votre magasin de données.

Pourquoi mon jeu de données s'affiche-t-il simplement `__dt` ?

- Cette colonne est ajoutée automatiquement par le service et contient l'heure approximative d'ingestion des données. Elle peut être utilisée pour optimiser vos requêtes. Si votre jeu de données ne contient rien d'autre que cela, reportez-vous à la question et à la réponse précédentes.

Comment coder un événement provoqué par la complétion de l'ensemble de données ?

- Vous devez configurer le sondage en fonction de la **`describe-dataset`** commande pour vérifier si le statut de l'ensemble de données avec un horodatage particulier est RÉUSSI.

Comment configurer correctement mon instance de bloc-notes à utiliser AWS IoT Analytics ?

Suivez ces étapes pour vous assurer que le rôle IAM que vous utilisez pour créer l'instance de bloc-notes dispose des autorisations requises :

1. Accédez à la console SageMaker AI et créez une instance de bloc-notes.
2. Renseignez les détails et choisissez Create a new role (Créer un rôle). Notez l'ARN du rôle.
3. Créez l'instance de blocs-notes. Cela crée également un rôle que l' SageMaker IA peut utiliser.
4. Accédez à la console IAM et modifiez le rôle SageMaker AI nouvellement créé. Lorsque vous ouvrez ce rôle, il doit comporter une stratégie gérée.
5. Cliquez sur Ajouter une politique intégrée, choisissez Io TAnalytics comme service et, sous autorisation de lecture, sélectionnez GetDatasetContent.
6. Examinez la stratégie, ajoutez un nom de stratégie, puis créez-la. Le rôle nouvellement créé dispose désormais d'une autorisation politique lui permettant de lire un ensemble de données AWS IoT Analytics.
7. Accédez à la AWS IoT Analytics console et créez des blocs-notes dans l'instance de bloc-notes.
8. Attendez que l'instance de bloc-notes soit à l'état « In Service » (En service).
9. Choisissez create notebooks (créer des blocs-notes) et sélectionnez l'instance que vous avez créée. Cela crée un bloc-notes Jupyter avec le modèle sélectionné qui peut accéder à vos ensembles de données.

Pourquoi ne puis-je pas créer de blocs-notes dans une instance ?

- Veillez à créer une instance de bloc-notes avec la stratégie IAM correcte. (Suivez les étapes de la question précédente.)

- Assurez-vous que l'instance de bloc-notes est à l'état « In Service » (En service). Lorsque vous créez une instance, elle démarre dans un état « En attente ». En général, il faut environ cinq minutes pour qu'elle passe à l'état « In Service ». Si l'instance du bloc-notes passe à l'état « Échec » au bout de cinq minutes environ, vérifiez à nouveau les autorisations.

Pourquoi est-ce que je ne vois pas mes ensembles de données dans Quick Suite ?

Quick Suite peut avoir besoin d'une autorisation pour lire le contenu AWS IoT Analytics de votre ensemble de données. Pour donner votre autorisation, procédez comme suit.

1. Choisissez le nom de votre compte dans le coin supérieur droit de Quick Suite, puis sélectionnez Gérer. QuickSight
2. Dans le volet de navigation de gauche, sélectionnez Security & permissions (Sécurité et autorisations). Sous QuickSight Accès aux AWS services, vérifiez que l'accès est accordé à AWS IoT Analytics.
   a. S'il AWS IoT Analytics n'y a pas accès, choisissez Ajouter ou supprimer.
   b. Cochez la case à côté, AWS IoT Analyticspuis sélectionnez Mettre à jour. Cela donne à Quick Suite l'autorisation de lire le contenu de votre ensemble de données.
3. Réessayez pour visualiser vos données.

Assurez-vous de choisir la même AWS région pour Quick Suite AWS IoT Analytics et pour les deux. Dans le cas contraire, vous pourriez rencontrer des difficultés pour accéder aux AWS ressources. Pour la liste des régions prises en charge, consultez les sections AWS IoT Analytics Points de terminaison et quotas et Points de terminaison et quotas Quick Suite dans le. Référence générale d'Amazon Web Services

Pourquoi le bouton de conteneurisation ne s'affiche-t-il pas sur mon bloc-notes Jupyter existant ?

- Cela est dû à l'absence d'un plugin de AWS IoT Analytics conteneurisation. Si vous avez créé votre instance de SageMaker bloc-notes avant le 23 août 2018, vous devez installer le plug-in manuellement en suivant les instructions de la section Conteneurisation d'un bloc-notes.

- Si le bouton de conteneurisation ne s'affiche pas après avoir créé l'instance de SageMaker bloc-notes à partir de la AWS IoT Analytics console ou après l'avoir installée manuellement, contactez le support AWS IoT Analytics technique.

Pourquoi l'installation de mon plugin de conteneurisation échoue-t-elle ?

- En général, l'installation du plugin échoue en raison d'autorisations manquantes dans l'instance du SageMaker bloc-notes. Pour vérifier les autorisations nécessaires pour l'instance de bloc-notes, consultez Permissions et ajoutez les autorisations requises au rôle de l'instance de bloc-notes. Si le problème persiste, créez une nouvelle instance de bloc-notes à partir de la AWS IoT Analytics console.
- Vous pouvez ignorer en toute sécurité le message suivant dans le journal s'il apparaît lors de l'installation du plugin : « Pour initialiser cette extension dans le navigateur chaque fois que le bloc-notes (ou une autre application) se charge. »

Pourquoi mon plugin de conteneurisation génère-t-il une erreur ?

- La conteneurisation peut échouer et générer des erreurs pour plusieurs raisons. Assurez-vous que vous utilisez le kernel correct avant de conteneuriser le bloc-notes. Les kernels conteneurisés commencent par le préfixe « Containerized ».
- Étant donné que le plug-in crée et enregistre une image Docker dans un référentiel ECR, assurez-vous que le rôle de votre instance de bloc-notes dispose d'autorisations suffisantes pour lire, répertorier et créer des référentiels ECR. Pour vérifier les autorisations nécessaires pour l'instance de bloc-notes, consultez Permissions et ajoutez les autorisations requises au rôle de l'instance de bloc-notes.
- Assurez-vous également que le nom du référentiel est conforme aux exigences ECR. Les nom de référentiel ECR doivent commencer par une lettre et peuvent uniquement contenir des lettres minuscules, des chiffres, des tirets, des traits de soulignement et des barres obliques.
- Si le processus de conteneurisation échoue avec l'erreur suivante : « Cette instance ne dispose pas d'espace libre suffisant pour exécuter la conteneurisation », essayez d'utiliser une instance plus grande pour résoudre le problème.
- Si vous observez des erreurs de connexion ou de création d'image, veuillez réessayer. Si le problème persiste, redémarrez l'instance et installez la dernière version du plug-in.

Pourquoi est-ce que je ne vois pas mes variables pendant la conteneurisation ?

- Le plugin de AWS IoT Analytics conteneurisation reconnaît automatiquement toutes les variables de votre bloc-notes après avoir exécuté le bloc-notes avec le noyau « conteneurisé ». Utilisez l'un des kernels conteneurisées pour exécuter le bloc-notes, puis effectuez la conteneurisation.

Quelles variables puis-je ajouter à mon conteneur en tant qu'entrée ?

- Vous pouvez ajouter n'importe quelle variable dont vous voulez modifier la valeur lors de l'exécution en tant qu'entrée dans votre conteneur. Cela vous permet d'exécuter le même conteneur avec différents paramètres qui doivent être fournis au moment de la création du jeu de données. Le plugin Jupyter de AWS IoT Analytics conteneurisation simplifie ce processus en reconnaissant automatiquement les variables du bloc-notes et en les rendant disponibles dans le cadre du processus de conteneurisation.

Comment définir la sortie de mon conteneur comme entrée pour une analyse ultérieure ?

- Un emplacement S3 spécifique où les artefacts exécutés peuvent être stockés est créé pour chaque exécution de l'ensemble de données de conteneur. Pour accéder à l'emplacement de cette sortie, créez une variable avec le type `outputFileUriValue` dans l'ensemble de données du conteneur. La valeur de cette variable doit être un chemin S3 qui est utilisé pour stocker vos fichiers de sortie supplémentaires. Pour accéder à ces artefacts enregistrés lors des exécutions suivantes, vous pouvez utiliser l'`getDatasetContent`API et sélectionner le fichier de sortie approprié requis pour l'exécution suivante.

Pourquoi mon jeu de données de conteneurs échoue-t-il ?

- Assurez-vous de transmettre le bon code au jeu `executionRole` de données du conteneur. La politique de confiance du `executionRole` doit inclure à la fois `iotanalytics.amazonaws.com` et`sagemaker.amazonaws.com`.

- Si vous considérez `AlgorithmError` que c'est la raison de l'échec, essayez de déboguer le code de votre conteneur manuellement. Cela se produit si le code du conteneur contient un bug ou si le rôle d'exécution ne dispose pas de l'autorisation d'exécuter le conteneur. Si vous avez conteneurisé à l'aide du plug-in AWS IoT Analytics Jupyter, créez une nouvelle instance de SageMaker bloc-notes ayant le même rôle que le rôle ExecutionRole du ContainerDataset et essayez d'exécuter le bloc-notes manuellement. Si le conteneur a été créé en dehors du plug-in Jupyter, essayez d'exécuter manuellement le code et de limiter l'autorisation au rôle d'exécution.

Historique du document

Le tableau suivant décrit les modifications importantes apportées au guide de l'AWS IoT Analytics utilisateur après le 3 novembre 2020. Pour plus d'informations sur les mises à jour de cette documentation, vous pouvez vous abonner à un flux RSS.

Modification	Description	Date
Avis de fin de support	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.	20 mai 2025
AWS IoT Analytics n'est plus disponible pour les nouveaux clients	AWS IoT Analytics n'est plus disponible pour les nouveaux clients. Les clients existants de AWS IoT Analytics peuvent continuer à utiliser le service normalement. En savoir plus	8 août 2024
Lancement de la région	AWS IoT Analytics est désormais disponible dans la région Asie-Pacifique (Mumbai).	18 août 2021
Requête avec `JOIN`	Cette mise à jour vous permet de l'utiliser `JOIN` pour interroger un AWS IoT Analytics ensemble de données.	27 Juillet 2021

Intégration avec AWS IoT SiteWise	Vous pouvez désormais l'utiliser AWS IoT Analytics pour interroger AWS IoT SiteWise des données.	27 Juillet 2021
Cloisons personnalisées	AWS IoT Analytics prend désormais généralement en charge le partitionnement de vos données en fonction des attributs des messages ou des attributs ajoutés par le biais des activités du pipeline.	14 juin 2021
Retraitement des messages du canal	Cette mise à jour vous permet de retraiter les données de canal dans les objets Amazon S3 spécifiés.	15 décembre 2020
Schéma du parquet	AWS IoT Analytics les magasins de données prennent désormais en charge le format de fichier Parquet.	15 décembre 2020
Surveillance à l'aide d' CloudWatch événements	AWS IoT Analytics publie automatiquement un événement sur Amazon CloudWatch Events lorsqu'un e erreur d'exécution se produit au cours d'une AWS Lambda activité.	15 décembre 2020
Notification tardive des données	Vous pouvez utiliser cette fonctionnalité pour recevoir des notifications via Amazon CloudWatch Events lorsque des données arrivent en retard.	9 novembre 2020

Lancement de la région	Lancé AWS IoT Analytics en Chine (Pékin).	4 novembre 2020

Mises à jour antérieures

Le tableau suivant décrit les modifications importantes apportées au guide de AWS IoT Analytics l'utilisateur avant le 4 novembre 2020.

Modification	Description	Date
Lancement de la région	Lancé AWS IoT Analytics dans la région Asie-Pacifique (Sydney).	16 juillet 2020
Mettre à jour	Réorganisation de la documentation.	07 mai 2020

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.