AWS IoT Core

Manuel du développeur

AWS IoT Core : Manuel du développeur

Table of Contents

Présentation d'AWS IoT

AWS IoTfournit les services cloud qui connectent vos appareils IoT à d'autres appareils etAWSservices cloud.AWS IoTfournit un logiciel d'appareil qui peut vous aider à intégrer vos appareils IoT dansAWS IoTsolutions basées sur des solutions. Si vos appareils peuvent se connecter àAWS IoT,AWS IoTpeut les connecter aux services cloudAWSfournit.

AWS IoTvous permet de sélectionner les technologies les plus appropriées et les plus à jour pour votre solution. Pour vous aider à gérer et à prendre en charge vos appareils IoT sur le terrain, AWS IoT Core prend en charge ces protocoles :

- MQTT (Message Queuing and Telemetry Transport) (p. 82)
- MQTT sur WSS (Websockets Secure) (p. 82)
- HTTPS (protocole de transfert hypertexte - sécurisé) (p. 86)
- LoRaWan (réseau étendu à longue portée) (p. 1062)

La . AWS IoT Core Message Broker prend en charge les appareils et les clients qui utilisent MQTT et MQTT sur les protocoles WSS pour publier et s'abonner aux messages. Il prend également en charge les appareils et les clients qui utilisent le protocole HTTPS pour publier des messages.

AWS IoT Core pour LoRaWan vous aide à connecter et à gérer des appareils sans fil LoRaWan (réseau étendu à longue portée de faible consommation). AWS IoT Core pour LoRaWan remplace la nécessité pour vous de développer et d'exploiter un serveur réseau LoRaWan (LNS).

Si vous n'avez pas besoinAWS IoTfonctionnalités telles que les communications sur les appareils,Règles (p. 394), oujobs (p. 595), voir,AWSMessageriepour plus d'informations sur d'autresAWS IoTqui pourraient mieux répondre à vos besoins.

Que vous soyez nouveau dans l'IoT ou que vous ayez des années d'expérience, assurez-vous de consulterFonctionnement de AWS IoT (p. 4). Cette rubrique vous aide à comprendreAWS IoTLes concepts et les termes pour vous aider à démarrer avecAWS IoTplus efficacement.

Comment démarrer avec AWS IoT

- RegarderInsideAWS IoTet ses composants dansFonctionnement de AWS IoT (p. 4).
- ProcédureEn savoir plus surAWS IoT (p. 12)de notre collection de matériel de formation et de vidéos. Cette rubrique inclut également une liste de services quiAWS IoTpeut se connecter à, des liens de médias sociaux et des liens vers des spécifications de protocole de communication.
- Connexionvotre premier appareil àAWS IoTinMise en route avec AWS IoT Core (p. 18).
- Développementvos solutions IoT enConnexion à AWS IoT Core (p. 67)et d'explorer leDidacticiels AWS IoT (p. 113).
- Tester et validervos appareils IoT pour une communication sécurisée et fiable à l'aide duDevice Advisor (p. 998).
- Gérervotre solution en utilisant AWS IoT Core Les services de gestion tels queService d'indexation de flotte (p. 758),Jobs (p. 595), etAWS IoT Device Defender (p. 790).
- Analyserles données de vos appareils à l'aide de l'outilServices de données AWS IoT (p. 9).

Comment vos appareils et applications accèdentAWS IoT

AWS IoTfournit les interfaces suivantes pourDidacticiels AWS IoT (p. 113) :

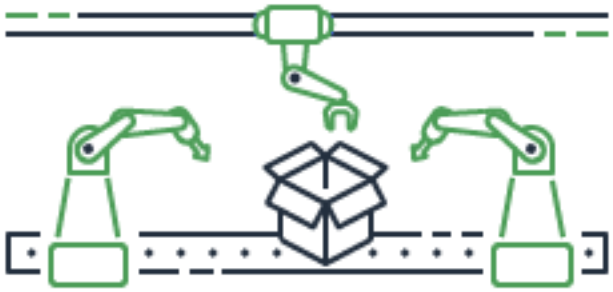
- AWS IoTKits SDK pour les appareils : créez sur vos appareils des applications qui envoient et reçoivent des messages à destination et en provenance d'AWS IoT. Pour plus d'informations, consultez AWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil (p. 1140).
- AWS IoT Core pour LoRaWAN : connectez et gérez vos périphériques et passerelles longue portée WAN (LoRaWan) à l'aide de AWS IoT Core pour LoRaWAN (p. 1062).
- AWS Command Line Interface (AWS CLI) – Exécute des commandes pour AWS IoT sur Windows, macOS et Linux. Ces commandes vous permettent de créer et de gérer des objets objets, des certificats, des règles, des tâches et des stratégies. Consultez le pour démarrer le fichierAWS Command Line InterfaceGuide de l'utilisateur. Pour plus d'informations sur les commandes pourAWS IoT, voir,iotdans leAWS CLIRéférence des commandes.
- API AWS IoT – Crée vos applications IoT à l'aide de requêtes HTTP ou HTTPS. Ces actions d'API vous permettent par programmation de créer et de gérer des objets objets, des certificats, des règles et des stratégies. Pour plus d'informations sur les actions d'API pour AWS IoT, consultez Actions dans la AWS IoTRéférence d'API.
- AWSKits SDK : créez vos applications IoT à l'aide d'API propres au langage. Ces kits SDK intègrent les API HTTP/HTTPS et vous permettent de programmer dans n'importe quelle langue prise en charge. Pour plus d'informations, consultez Kits SDK et outils AWS.

Vous pouvez également accéder àAWS IoTvia le fichierAWS IoTconsole, qui fournit une interface utilisateur graphique (GUI) via laquelle vous pouvez configurer et gérer les objets objets, certificats, règles, tâches, stratégies et autres éléments de vos solutions IoT.

Présentation d'AWS IoTpeut faire

Cette rubrique décrit certaines des solutions dont vous pourriez avoir besoinAWS IoTprend en charge.

IoT dans l'industrie



Voici quelques exemples d'AWS IoTSolutions pourCas d'utilisation industriellequi appliquent les technologies IoT pour améliorer les performances et la productivité des processus industriels.

Solutions pour les cas d'utilisation industrielle

- UtiliserAWS IoTconstruire des modèles de qualité prédictifs dans les opérations industrielles

  Voir commentAWS IoTpeut collecter et analyser des données provenant d'opérations industrielles pour construire des modèles de qualité prédictifs. En savoir plus
- UtiliserAWS IoTpour soutenir la maintenance prédictive dans les opérations industrielles

  Voir commentAWS IoTpeut aider à planifier la maintenance préventive afin de réduire les temps d'arrêt imprévus. En savoir plus

IoT dans la domotique

Voici quelques exemples d'AWS IoTSolutions pourCas d'utilisation domotiquequi appliquent les technologies IoT pour créer des applications IoT évolutives qui automatisent les activités domestiques à l'aide d'appareils domestiques connectés.

Solutions pour la domotique

- UtiliserAWS IoTdans votre maison connectée

  Voir commentAWS IoTpeut fournir des solutions de domotique intégrées.
- UtiliserAWS IoTassurer la sécurité et la surveillance du domicile

Voir commentAWS IoTpeut appliquer l'apprentissage automatique et l'informatique de périphérie à votre solution domotique.

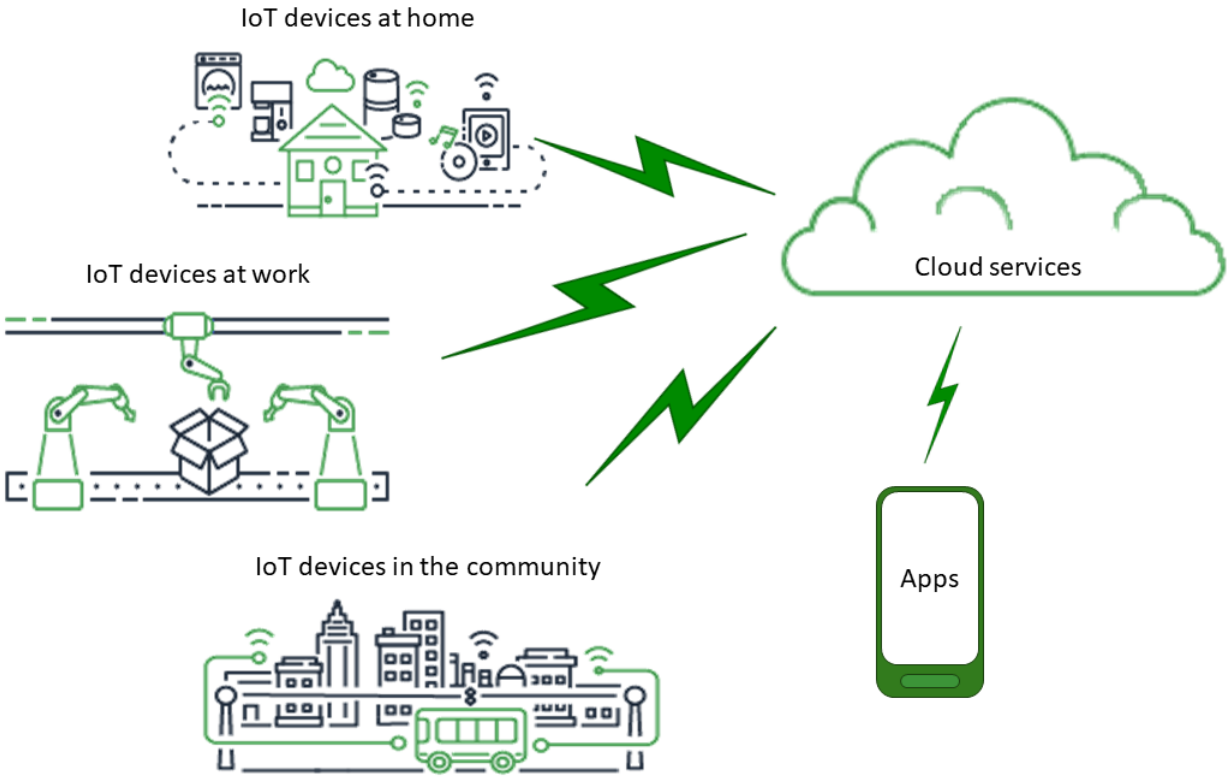
Pour obtenir la liste des solutions pour les cas d'utilisation industrielle, grand public et commerciale, consultez laAWS IoTDépôt de solutions.

Fonctionnement de AWS IoT

AWS IoTfournit des services cloud et une prise en charge des appareils que vous pouvez utiliser pour implémenter des solutions IoT.AWSfournit de nombreux services cloud pour prendre en charge les applications basées sur l'IOT. Donc, pour vous aider à comprendre par où commencer, cette section fournit un diagramme et une définition de concepts essentiels pour vous initier à l'univers de l'IoT.

L'univers IoT

En général, l'Internet des objets (IoT) se compose des composants clés présentés dans ce diagramme.



Apps

Les applications permettent aux utilisateurs finaux d'accéder aux appareils IoT et aux fonctionnalités fournies par les services cloud auxquels ces appareils sont connectés.

Services cloud

Les services cloud sont des services de stockage et de traitement de données distribués à grande échelle qui sont connectés à Internet. En voici quelques exemples :

- Services de connexion et de gestion IoT.

  AWS IoTest un exemple de service de connexion et de gestion IoT.
- Services de calcul, tels qu'Amazon Elastic Compute Cloud etAWS Lambda.
- Services de base de données, tels qu'Amazon DynamoDB

Communications

Les appareils communiquent avec les services cloud à l'aide de diverses technologies et protocoles. En voici quelques exemples :

- WiFi/Internet haut débit
- Données cellulaires haut débit
- Données cellulaires à bande étroite
- Réseau étendu à longue portée (LoRaWan)
- Communications RF propriétaires

Devices

Un périphérique est un type de matériel qui gère les interfaces et les communications. Les appareils sont généralement situés à proximité des interfaces réelles qu'ils surveillent et contrôlent. Les périphériques peuvent inclure des ressources informatiques et de stockage, telles que des microcontrôleurs, des CPU, de la mémoire. En voici quelques exemples :

- Raspberry Pi
- Arduino
- Assistants d'interface vocale
- LoRaWAN et appareils
- Appareils Amazon Strottoir
- Périphériques IoT personnalisés

Interfaces

Une interface est un composant qui connecte un appareil au monde physique.

- Interfaces utilisateur

  Les composants qui permettent aux appareils et aux utilisateurs de communiquer entre eux.
  - Interfaces d'entrée

    Permettre à un utilisateur de communiquer avec un appareil

    Exemples : clavier, bouton
  - Interfaces de sortie

    Activer un périphérique pour communiquer avec un utilisateur

    Exemples : Affichage alphanumérique, affichage graphique, voyant lumineux, sonnette d'alarme
- Détecteurs

  Insérez des composants qui mesurent ou sentent quelque chose dans le monde extérieur d'une manière qu'un appareil comprend. En voici quelques exemples :

- Capteur de température (convertit la température en tension analogique)
- Capteur d'humidité (convertit l'humidité relative en tension analogique)
- Convertisseur analogique-numérique (convertit une tension analogique en une valeur numérique)
- Unité de mesure de distance par ultrasons (convertit une distance en valeur numérique)
- Capteur optique (convertit un niveau de lumière en valeur numérique)
- Appareil photo (convertit les données d'image en données numériques)

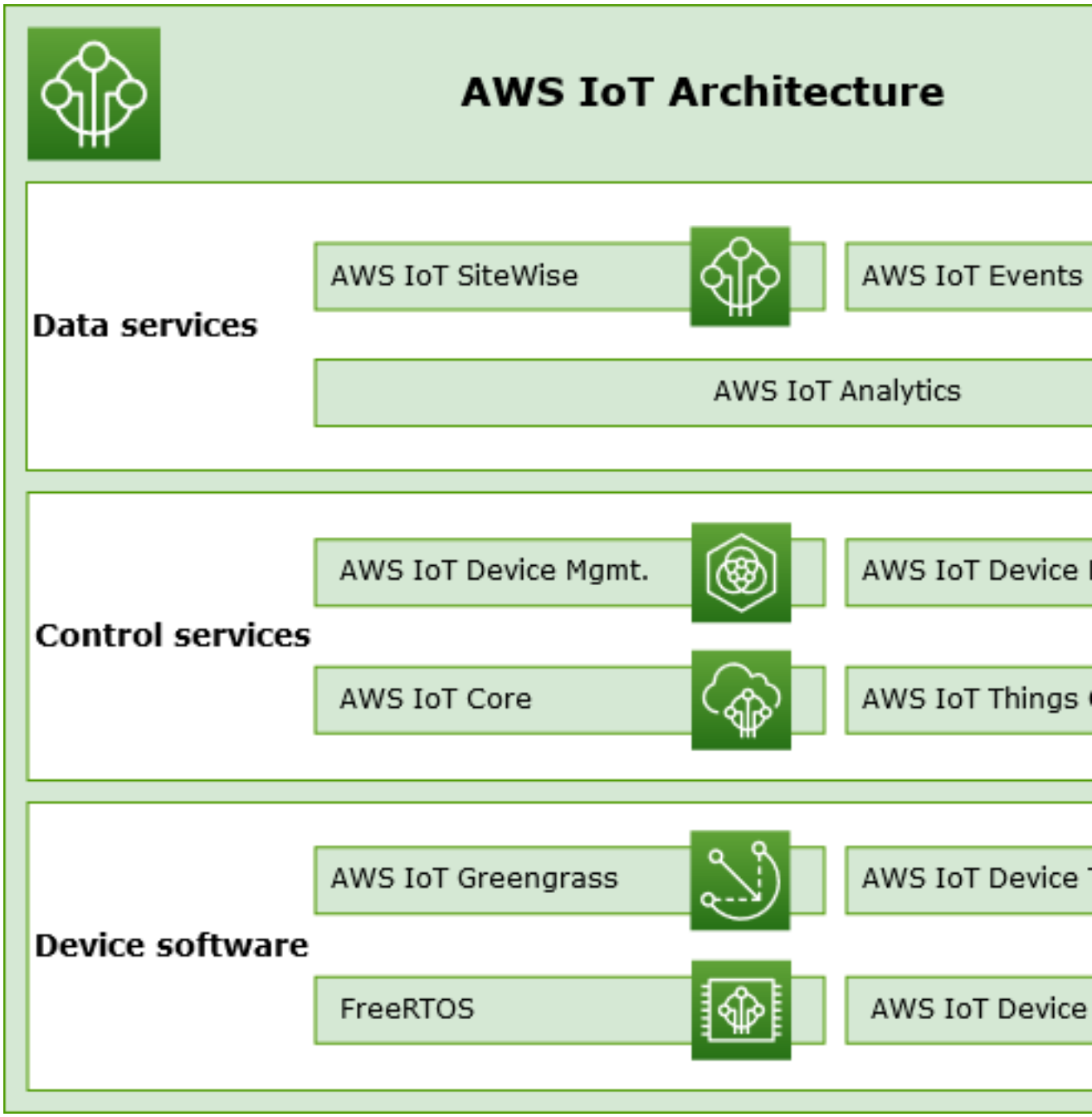
- Actionneurs

  Composants de sortie que l'appareil peut utiliser pour contrôler quelque chose dans le monde extérieur. En voici quelques exemples :

  - Moteurs pas à pas (convertir les signaux électriques en mouvement)
  - Relais (contrôle des tensions et des courants électriques élevés)

AWS IoTPrésentation des services

Dans l'univers de l'IoT,AWS IoTfournit les services qui prennent en charge les appareils qui interagissent avec le monde et les données qui passent entre eux etAWS IoT.AWS IoTest composé des services présentés dans cette illustration pour prendre en charge votre solution IoT.



AWS IoTlogiciel de périphérique

AWS IoTfournit ce logiciel pour prendre en charge vos appareils IoT.

AWS IoTGreengrass

> AWS IoTGreengrasss'étendAWS IoTpour une exploitation en local des données qu'ils génèrent et utiliser le cloud pour la gestion, l'analyse et le stockage de longue durée. avecAWS IoTGreengrass, les appareils connectés peuvent fonctionnerAWS Lambda, les conteneurs Docker, ou les deux, exécutent des prédictions basées sur les modèles de Machine Learning, assurent la synchronisation

des données des appareils et communiquent en toute sécurité avec d'autres appareils, même en l'absence de connexion Internet.

AWS IoT Device Tester

AWS IoT Device Tester pour FreeRTOS etAWS IoT Greengrassest un outil d'automatisation de test pour les microcontrôleurs. AWS IoT Device Tester , testez votre appareil pour déterminer s'il exécutera FreeRTOS ouAWS IoT Greengrasset interagir avecAWS IoTServices .

Kits SDK pour les appareils AWS IoT

La .AWS IoTKit de développement logiciel (SDK) des appareils et mobiles (p. 1140)vous aider à connecter efficacement vos appareils àAWS IoT. Les kits SDK pour les appareils et mobiles AWS IoT incluent des bibliothèques open source, des manuels pour développeurs avec des exemples, ou encore des manuels de transfert afin de vous permettre de créer des produits et des solutions IoT innovantes sur les plateformes matérielles de votre choix.

FreeRTOS

FreeRTOSest un système d'exploitation open source en temps réel pour les microcontrôleurs qui vous permet d'inclure de petits périphériques de périphérie de faible consommation dans votre solution IoT. FreeRTOS inclut un noyau et un ensemble croissant de bibliothèques logicielles qui prennent en charge de nombreuses applications. Les systèmes FreeRTOS peuvent connecter en toute sécurité vos petits appareils à faible consommation àAWS IoTet prennent en charge les périphériques périphériques plus puissants exécutantAWS IoT Greengrass.

AWS IoTservices de contrôle

Connect à l'adresse suivanteAWS IoTpour gérer les appareils de votre solution IoT.

AWS IoT Core

AWS IoT Core est un service basé sur le cloud géré qui permet aux appareils connectés d'interagir en toute sécurité avec les applications cloud et d'autres appareils. AWS IoT Core peut prendre en charge de nombreux périphériques et messages, et il peut traiter et acheminer ces messages versAWS IoTpoints de terminaison et autres périphériques. avec AWS IoT Core , vos applications peuvent interagir avec tous vos appareils même lorsqu'ils ne sont pas connectés.

AWS IoTGestion des appareils

AWS IoTGestion des appareilsvous permettent de suivre, de surveiller et de gérer la pléthore d'appareils connectés qui composent vos flottes d'appareils.AWS IoT Les services de gestion des appareils vous aident à vous assurer que vos appareils IoT fonctionnent correctement et en toute sécurité après leur déploiement. Ils fournissent également des tunnels sécurisés pour accéder à vos appareils, surveiller leur état de santé, détecter et résoudre à distance les problèmes, ainsi que des services de gestion des mises à jour logicielles et micrologiciels des appareils.

Device Defender AWS IoT

AWS IoTDevice Defendervous aide à sécuriser votre flotte d'appareils IoT.AWS IoT Device Defender vérifie en permanence vos configurations IoT pour s'assurer qu'elles ne s'écartent pas des meilleures pratiques en matière de sécurité.AWS IoT Device Defender envoie une alerte lorsqu'il détecte des lacunes dans votre configuration IoT susceptibles de créer un risque de sécurité, telles que des certificats d'identité partagés sur plusieurs appareils ou un appareil doté d'un certificat d'identité révoqué essayant de se connecter à AWS IoT Core .

AWS IoTGraphique d'objets

AWS IoTGraphique d'objetsest un service qui vous permet de connecter visuellement différents appareils et services Web pour créer des applications IoT.AWS IoT Things Graph fournit une interface visuelle glisser-déposer pour connecter et coordonner les interactions entre les appareils et les services Web, afin que vous puissiez créer efficacement des applications IoT.

Services de données AWS IoT

Analysez les données des appareils de votre solution IoT et prenez les mesures appropriées à l'aide des outils suivantsAWS IoTServices .

Analyse AWS IoT

AWS IoTAnalyseVous permet d'exécuter et d'opérationnaliser efficacement des analyses avancées sur de gros volumes de données IoT non structurées.AWS IoT Analytics automatise chaque étape difficile requise pour l'analyse de données provenant d'appareils IoT.AWS IoT Analytics filtre, transforme et enrichit les données IoT avant de les stocker dans un magasin de données en séries chronologiques à des fins d'analyse. Vous pouvez analyser vos données en effectuant des requêtes ponctuelles ou planifiées à l'aide du moteur de requête SQL intégré ou du Machine Learning.

AWS IoTSiteWise

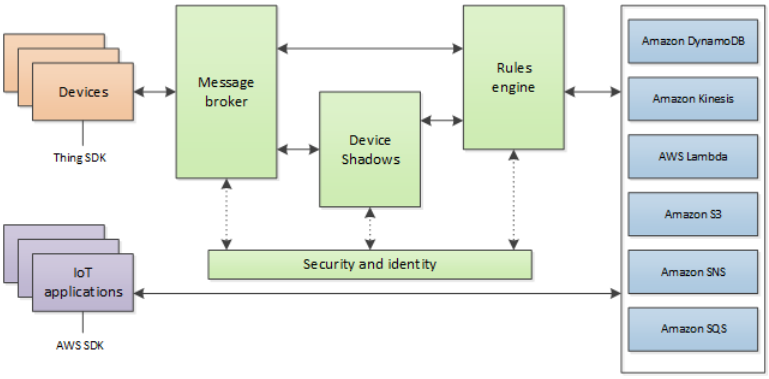
AWS IoTSiteWisecollecte, stocke, organise et surveille les données transmises à partir d'équipements industriels par des messages MQTT ou des API à grande échelle en fournissant des logiciels qui s'exécutent sur une passerelle dans vos installations. La passerelle se connecte en toute sécurité à vos serveurs de données locaux et automatise le processus de collecte et d'organisation des données et de leur envoi auAWSCloud.

Événements AWS IoT

AWS IoTEvents (Événements)détecte les événements provenant des capteurs et des applications IoT et y répond. Les événements sont des modèles de données qui identifient des circonstances plus compliquées que prévu, comme les détecteurs de mouvement utilisant des signaux de mouvement pour activer les lumières et les caméras de sécurité.AWS IoT Les événements surveillent en permanence les données provenant de plusieurs capteurs et applications IoT, et s'intègrent à d'autres services, tels que AWS IoT Core , IoT SiteWise, DynamoDB et autres pour permettre une détection précoce et des informations uniques.

Services AWS IoT Core

AWS IoT Core fournit les services qui connectent vos appareils IoT auAWSCloud afin que d'autres services et applications cloud puissent interagir avec vos appareils connectés à Internet.



La section suivante décrit chacun des AWS IoT Core affichés dans l'illustration.

AWS IoT Core services de messagerie

La . AWS IoT Core fournissent une communication sécurisée avec les appareils IoT et gèrent les messages qui passent entre eux etAWS IoT.

Passerelle pour les appareils

Permet aux appareils de communiquer de manière efficace et en toute sécurité avec AWS IoT. La communication de l'appareil est sécurisée par des protocoles sécurisés utilisant les certificats X.509.

Agent de messages

Offre un mécanisme sécurisé pour que les appareils et les applications AWS IoT publient et reçoivent des messages les uns des autres. Vous pouvez utiliser le protocole MQTT directement ou MQTT sur WebSocket pour effectuer des publications et vous abonner. Pour plus d'informations sur les protocoles queAWS IoTprend en charge, voirthe section called “Protocoles de communication de périphérique” (p. 79). Les appareils et les clients peuvent également utiliser l'interface HTTP REST pour publier des données vers le courtier de messages.

Le courtier de messages distribue les données de périphérique aux périphériques qui y sont abonnés et à d'autres AWS IoT Core , tels que le service Device Shadow et le moteur de règles.

AWS IoT Core pour LoRaWAN

AWS IoT Core pour LoRaWan permet de configurer un réseau LoRaWan privé en connectant vos appareils et passerelles LoRaWan àAWSsans qu'il soit nécessaire de développer et d'exploiter un serveur réseau LoRaWan (LNS). Les messages reçus des appareils LoRaWan sont envoyés au moteur de règles où ils peuvent être formatés et envoyés à d'autresAWS IoTServices .

Moteur de règles

Le moteur de règles connecte les données du courtier de messages à d'autresAWS IoTservices de stockage et de traitement supplémentaire. Vous pouvez, par exemple, insérer, mettre à jour ou interroger une table DynamoDB ou appeler une fonction Lambda en fonction d'une expression que vous avez définie dans le moteur de règles. Vous pouvez utiliser un langage SQL pour sélectionner des données à partir des charges utiles de messages, puis traiter et envoyer les données à d'autres services, comme Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB etAWS Lambda. Vous pouvez également créer des règles qui republient des messages sur le courtier de messages et sur d'autres abonnés. Pour plus d'informations, consultez Règles pour AWS IoT (p. 394).

AWS IoT Core services de contrôle

La . AWS IoT Core fournissent des fonctionnalités de sécurité, de gestion et d'enregistrement des appareils.

Service d'authentification personnalisé

Vous pouvez définir des mécanismes d'autorisation personnalisée qui vous permettent de gérer votre propre stratégie d'authentification et d'autorisation à l'aide d'un service d'authentification personnalisé et d'une fonction Lambda. Les mécanismes d'autorisation personnalisée permettent à AWS IoT d'authentifier vos appareils et d'autoriser des opérations à l'aide de stratégies d'autorisation et d'authentification par jeton de porteur.

Les autorisations personnalisées peuvent implémenter diverses stratégies d'authentification, par exemple, la vérification du jeton Web JSON ou la légende du fournisseur OAuth. Ils doivent renvoyer les documents de stratégie utilisés par la passerelle pour les appareils de façon à autoriser les opérations MQTT. Pour plus d'informations, consultez Authentification personnalisée (p. 256).

Service de mise en service d'appareils

Permet de mettre en service des appareils à l'aide d'un modèle qui décrit les ressources requises par chaque appareil : unobjet d'un objet, un certificat et une ou plusieurs stratégies. Un objet objet objet objet est une entrée du registre qui contient des attributs décrivant un appareil. Les appareils utilisent des certificats pour l'authentification auprès d'AWS IoT. Les stratégies déterminent les opérations qu'un appareil peut effectuer dans AWS IoT.

Les modèles contiennent des variables remplacées par des valeurs dans un dictionnaire (map). Vous pouvez utiliser le même modèle pour mettre en service plusieurs appareils en transmettant différentes valeurs pour les variables du modèle dans le dictionnaire. Pour plus d'informations, consultez Mise en service des appareils (p. 728).

Registre de groupe

Les groupes vous permettent de gérer plusieurs appareils simultanément en les classant dans des groupes. Les groupes peuvent eux-mêmes contenir d'autres groupes ; vous pouvez ainsi créer une hiérarchie de groupes. Toute action réalisée sur un groupe parent s'appliquera à ses groupes enfants. La même action s'applique également à tous les périphériques du groupe parent et à tous les périphériques des groupes enfants. Les autorisations octroyées à un groupe s'appliquent à tous les appareils du groupe et de ses groupes enfants. Pour plus d'informations, consultez Gestion des appareils avec AWS IoT (p. 203).

Service Jobs

Permet de définir un ensemble d'opérations distantes qui sont envoyées vers un ou plusieurs appareils connectés à un ou plusieurs appareils connectés àAWS IoT. Par exemple, vous pouvez définir une tâche qui ordonne à un ensemble d'appareils de télécharger et d'installer les mises à jour d'une application ou d'un microprogramme, de redémarrer, de procéder à une rotation des certificats ou d'exécuter des opérations de dépannage à distance.

Pour créer une tâche, vous devez préciser une description des opérations à distance à effectuer et une liste des cibles qui doivent les effectuer. Les cibles peuvent être des appareils individuels, des groupes ou les deux. Pour plus d'informations, consultez Jobs (p. 595).

Registre

Organise les ressources associées à chaque appareil dans leAWSCloud. Vous enregistrez vos appareils et associez jusqu'à trois attributs personnalisés à chacun d'entre eux. Vous pouvez également associer des certificats et des ID de client MQTT à chaque appareil pour améliorer votre capacité à gérer et dépanner les appareils. Pour plus d'informations, consultez Gestion des appareils avec AWS IoT (p. 203).

Service de sécurité et d'identité

Partage la responsabilité de la sécurité dans leAWSCloud. Vos appareils doivent protéger leurs informations d'identification pour envoyer des données en toute sécurité au courtier de messages. Le courtier de messages et le moteur de règles utilisentAWSFonctions de sécurité pour envoyer des données en toute sécurité à des appareils ou autres appareilsAWSServices . Pour plus d'informations, consultez Authentication (p. 232).

Services de données AWS IoT Core

La . AWS IoT Core aident vos solutions IoT à offrir une expérience applicative fiable, même avec des appareils qui ne sont pas toujours connectés.

Shadow d'appareil

Document JSON utilisé pour stocker et récupérer des informations d'état actualisées concernant un appareil.

Service Device Shadow

Le service Device Shadow conserve l'état d'un périphérique afin que les applications puissent communiquer avec un périphérique, que celui-ci soit en ligne ou non. Lorsqu'un périphérique est hors connexion, le service Device Shadow gère ses données pour les applications connectées. Lorsque le périphérique se reconnecte, il synchronise son état avec celui de son ombre dans le service Device Shadow. Vos appareils peuvent également publier leur état actuel dans un shadow utilisé par des applications ou d'autres appareils qui peuvent ne pas être connectés tout le temps. Pour plus d'informations, consultez Service AWS IoT Device Shadow (p. 547).

AWS IoT Core Service d'assistance

Intégration d'Alexa Voice Service (AVS) pourAWS IoT

Apporte Alexa Voice à n'importe quel appareil connecté. AVS pourAWS IoTréduit le coût et la complexité de l'intégration d'Alexa. Cette fonction utiliseAWS IoTPour décharger les tâches audio intensives de calcul et de mémoire de l'appareil vers le cloud. Étant donnée la réduction du coût de la nomenclature des matériaux d'ingénierie (EBOM) qui en résulte, les fabricants d'appareils peuvent amener Alexa à des appareils IoT dont les ressources sont limitées, et permettre aux consommateurs de parler directement à Alexa à certains endroits de leur maison, de leur bureau ou de leur chambre d'hôtel. Ils profitent ainsi d'une expérience ambiante.

AVS pourAWS IoTactive la fonctionnalité intégrée Alexa sur les microcontrôleurs, tels que les processeurs ARM Cortex de classe M avec moins de 1 Mo de RAM intégrée. Pour ce faire, AVS décharge la mémoire et les tâches de calcul vers un appareil virtuel avec Alexa intégré dans le cloud. Cela réduit le coût EBOM jusqu'à 50 %. Pour plus d'informations, consultez Intégration Alexa Voice Service (AVS) pourAWS IoT (p. 1132).

Intégration d'Amazon Sidewalk pour AWS IoT Core

Amazon Strowalkest un réseau partagé qui améliore les options de connectivité pour aider les appareils à mieux travailler ensemble. Amazon Sidewalk prend en charge un large éventail d'appareils clients, tels que ceux qui localisent des animaux domestiques ou des objets de valeur, ceux qui assurent la sécurité de la maison intelligente et le contrôle de l'éclairage, et ceux qui fournissent des diagnostics à distance pour les appareils et les outils. Intégration d'Amazon Sidewalk pour AWS IoT Core permet aux fabricants d'appareils d'ajouter leur parc d'appareils Sidewalk à laAWS IoTCloud.

Pour de plus amples informations, veuillez consulter Intégration d'Amazon Sidewalk pour AWS IoT Core (p. 1125)

En savoir plus sur AWS IoT

Cette rubrique vous aide à vous familiariser avec le monde deAWS IoT. Vous pouvez obtenir des informations générales sur la façon dont les solutions IoT sont appliquées dans divers cas d'utilisation, des ressources de formation, des liens vers les médias sociaux pourAWS IoTet tous les autresAWS, ainsi qu'une liste de services et de protocoles de communication quiAWS IoTUtilisations.

Ressources de formation pourAWS IoT

Nous offrons ces formations pour vous aider à en savoir plus surAWS IoTet la façon de les appliquer à la conception de votre solution.

- Introduction à AWS IoT

  Présentation vidéo deAWS IoTet ses services de base.
- Plongez profond dansAWS IoTAuthentification et autorisation

  Un cours avancé qui explore les concepts deAWS IoTAuthentification et autorisation. Vous apprendrez à authentifier et à autoriser les clients à accéder auAWS IoTLes API du plan de contrôle et du plan de données.
- Série Internet des objets (IoT)

  Un parcours d'apprentissage des modules eLearning IoT sur différentes technologies et fonctionnalités IoT.

AWS IoTressources et guides

Il s'agit de ressources techniques approfondies sur des aspects spécifiques deAWS IoT.

- Lens IoT —AWS IoTCadre Well-Architected

  Document décrivant les meilleures pratiques pour l'architecture de vos applications IoT surAWS.
- Conception de rubriques MQTT pour AWS IoT Core

  Document PDF décrivant les meilleures pratiques pour la conception des rubriques MQTT dans AWS IoT Core et la mise à profit AWS IoT Core avec MQTT.
- Fabrication et provisionnement de périphériques avec les certificats X.509 dans AWS IoT Core

  Document PDF qui décrit les différentes façons dontAWS IoTprévoit de fournir de grandes flottes d'appareils.
- AWS IoT Ressources

  Des ressources spécifiques à l'OIT, telles que des guides techniques, des architectures de référence, des livres électroniques et des articles de blog organisés, sont présentées dans un index consultable.
- Atlas IoT

  Vue d'ensemble sur la façon de résoudre les problèmes courants de conception de l'IoT. La .Atlas IoTfournit une analyse approfondie des défis de conception que vous êtes susceptible de rencontrer lors du développement de votre solution IoT.
- AWS Livres blancs et guides

  Notre collection actuelle de livres blancs et de guides surAWS IoTet autresAWSTechnologies.

AWS IoTdans les médias sociaux

Ces réseaux sociaux fournissent des informations surAWS IoTandAWSsujets connexes.

- Internet des objets (IoT) surAWS IoT— Blog officiel
- AWS IoT Vidéos sur la chaîne Amazon Web Services sur YouTube

Ces comptes de médias sociaux couvrent tous lesAWSservices, y comprisAWS IoT

- La chaîne Amazon Web Services sur YouTube
- Amazon Web Services
- Amazon Web Services
- Amazon Web Services
- Amazon Web Services

AWSles services utilisés par le AWS IoT Core Moteur de règles

La . AWS IoT Core peut se connecter à cesAWSServices .

- Amazon DynamoDB

Amazon DynamoDB est un service de base de données NoSQL évolutif offrant des performances de base de données rapides et prévisibles.

- Amazon Kinesis

  Amazon Kinesis facilite la collecte, le traitement et l'analyse des données en continu en temps réel afin que vous puissiez obtenir des informations opportunes et réagir rapidement aux nouvelles informations. Amazon Kinesis peut ingérer des données en temps réel telles que la vidéo, l'audio, les journaux d'applications, les flux de clics de sites Web et les données de télémétrie IoT pour l'apprentissage automatique, l'analyse et d'autres applications.

- AWS Lambda

  AWS Lambda vous permet d'exécuter du code sans avoir à allouer ou gérer des serveurs. Vous pouvez configurer votre code pour qu'il se déclenche automatiquement à partir deAWS IoTdonnées et événements ou appelez-les directement à partir d'une application Web ou mobile.

- Amazon Simple Storage Service

  Amazon Simple Storage Service (Amazon S3) peut stocker et récupérer n'importe quel volume de données à tout instant et depuis n'importe quel emplacement sur le Web.AWS IoTpeuvent envoyer des données à Amazon S3 pour le stockage.

- Amazon Simple Notification Service

  Amazon Simple Notification Service (Amazon SNS) est un service web qui permet à des applications, des utilisateurs finaux et des périphériques d'envoyer et recevoir des notifications depuis le cloud.

- Amazon Simple Queue Service

  Amazon Simple Queue Service (Amazon SQS) est un service de file d'attente de messagerie qui découplage et redimensionner les microservices, les systèmes distribués et les applications sans serveur.

- Amazon Elasticsearch Service

  Amazon Elasticsearch Service (Amazon ES) est un service géré qui facilite le déploiement, l'utilisation et le dimensionnement d'Elasticsearch, un moteur d'analyse et de recherche couramment utilisé et à code source libre.

- Amazon Machine Learning

  Amazon Machine Learning peut créer des modèles d'apprentissage automatique (ML) en recherchant des modèles dans vos données IoT. Le service utilise ces modèles pour traiter de nouvelles données et générer des prévisions pour votre application.

- Amazon CloudWatch

  Amazon CloudWatch fournit une solution de surveillance fiable, évolutive et flexible qui aide à régler, gérer et redimensionner vos propres systèmes et infrastructures de surveillance.

Protocoles de communication supportés par AWS IoT Core

Ces rubriques fournissent plus d'informations sur les protocoles de communication utilisés parAWS IoT. Pour plus d'informations sur les protocoles utilisés parAWS IoTet la connexion d'appareils et de services àAWS IoT, voir,Connexion à AWS IoT Core (p. 67).

- MQTT (Transport de télémétrie Message Queuing)

La page d'accueil du site MQT.org où vous pouvez trouver les spécifications du protocole MQTT. Pour plus d'informations sur la façon dontAWS IoTprend en charge MQTT, voirMQTT (p. 82).

- HTTPS (protocole de transfert hypertexte - sécurisé)

  Les appareils et les applications peuvent accéder àAWS IoTen utilisant HTTPS.
- LoRaWan (réseau étendu à longue portée)

  Les appareils et passerelles LoRaWan peuvent se connecter à AWS IoT Core en utilisant AWS IoT Core pour LoRaWAN.
- TLS (Transport Layer Security) v1.2

  La spécification du TLS v1.2 (RFC 5246).AWS IoTutilise TLS v1.2 pour établir des connexions sécurisées entre les appareils etAWS IoT.

Nouveautés en matière deAWS IoTconsole

Nous sommes sur le point de mettre à jour l'interface utilisateur duAWS IoTà une nouvelle expérience. Nous mettons à jour l'interface utilisateur par étapes, de sorte que certaines pages de la console auront une nouvelle expérience, d'autres peuvent avoir à la fois l'expérience originale et la nouvelle, et d'autres peuvent n'avoir que l'expérience originale.

Ce tableau s'affiche.AWS IoTInterface utilisateur de la console.

AWS IoTÉtat de l'interface utilisateur de la

Page de console	Expérience originale	Nouvelle expérience	Commentaires
Surveiller	Disponible	Disponible	
Activité	Non disponible	Disponible	
Intégration- Mise en route	Disponible	Pas encore disponible	
IntégrationModèles de mise en service d'une flotte	Disponible	Pas encore disponible	
Gérer- Des objets	Disponible	Disponible	
Gérer- Types	Disponible	Disponible	
GérerGroupes d'objets	Disponible	Disponible	
Gérer- Groupes de facturation	Disponible	Disponible	
Gérer- Tâches	Disponible	Disponible	
GérerModèles de Job	Disponible	Disponible	Les fonctionnalités Enregistrer un travail en tant que modèle de tâche et créer un travail avec un modèle de tâche ne sont pas disponibles dans l'expérience d'origine.

Page de console	Expérience originale	Nouvelle expérience	Commentaires
Gérer- Tunnels	Non disponible	Disponible	
Hub de flotte- Mise en route	Non disponible	Disponible	Pas disponible dans tous Région AWS s
Hub de flotte- Applications	Non disponible	Disponible	Pas disponible dans tous Région AWS s
Greengrass- Mise en route	Non disponible	Disponible	
Greengrass- Appareils de base	Non disponible	Disponible	
Greengrass- Composants	Non disponible	Disponible	
GreengrassDéploiements	Non disponible	Disponible	
Greengrass- Classique (V1)	Disponible	Non disponible	
La connectivité sans fil- Introduction	Non disponible	Disponible	Non disponible dans toutes les Région AWS s
La connectivité sans fil- passerelles	Non disponible	Disponible	Non disponible dans toutes les Région AWS s
La connectivité sans filAppareils -	Non disponible	Disponible	Non disponible dans toutes les Région AWS s
La connectivité sans fil- Profils	Non disponible	Disponible	Non disponible dans toutes les Région AWS s
La connectivité sans filDestinations Destinations	Non disponible	Disponible	Non disponible dans toutes les Région AWS s
Secure- Certificats	Disponible	Pas encore disponible	
Secure- Stratégies	Disponible	Pas encore disponible	
Secure- Autorité de certification	Disponible	Pas encore disponible	
Secure- Alias de rôle	Disponible	Pas encore disponible	
Secure- Autorisateurs d'autorisation	Disponible	Pas encore disponible	
Défendre- Introduction	Disponible	Pas encore disponible	
Défendre- Audit	Disponible	Pas encore disponible	
Défendre- Détect	Disponible	Pas encore disponible	

Page de console	Expérience originale	Nouvelle expérience	Commentaires
DéfendreActions d'atténuation	Disponible	Pas encore disponible	
Défendre- Paramètres	Disponible	Pas encore disponible	Non disponible dans toutes les Région AWS s
Acte- Règles	Disponible	Pas encore disponible	
ActeDestinations Destinations	Disponible	Pas encore disponible	
Test- Device Advisor	Disponible	Pas encore disponible	Non disponible dans toutes les Région AWS s
Test- Client de test MQTT	Disponible	Disponible	
Logiciels	Disponible	Pas encore disponible	
Paramètres	Disponible	Disponible	
Apprendre	Disponible	Pas encore disponible	

Legend

Valeurs d'état

- Available

  Cette expérience d'interface utilisateur peut être utilisée.
- Non disponible

  Cette expérience d'interface utilisateur ne peut pas être utilisée.
- Pas encore disponible

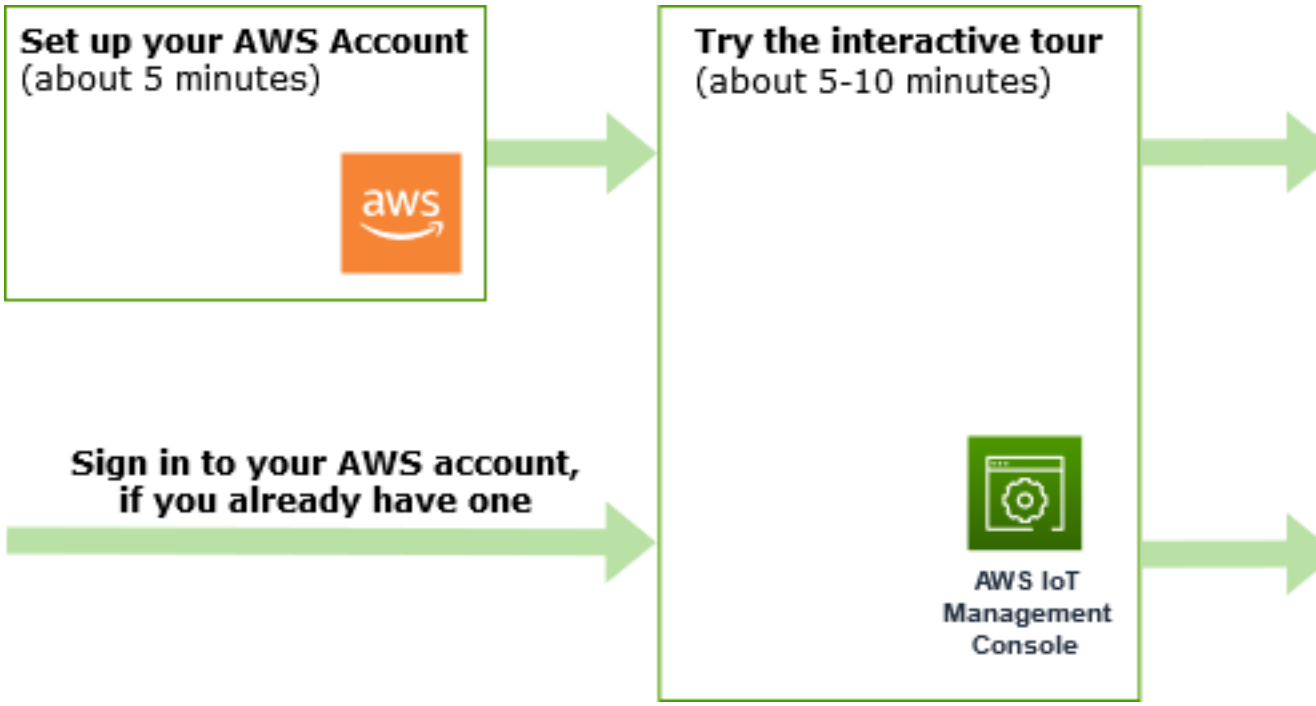
  La nouvelle interface utilisateur est en cours de travail, mais elle n'est pas encore prête.
- En cours

  La nouvelle interface utilisateur est sur le point d'être mise à jour. Cependant, certaines pages peuvent toujours avoir l'expérience utilisateur d'origine.

Mise en route avec AWS IoT Core

AWS IoT Core connectent des appareils IoT àAWS IoTServices et autresAWSServices . AWS IoT Core inclut la passerelle de périphériques et le courtier de messages, qui connectent et traitent les messages entre vos appareils IoT et le cloud.

Voici comment vous pouvez commencer avec AWS IoT Core andAWS IoT.



Cette section présente la AWS IoT Core Pour présenter ses services clés et fournit plusieurs exemples de la façon de connecter un appareil à AWS IoT Core et passer des messages entre eux. La transmission de messages entre les appareils et le cloud est fondamentale pour chaque solution IoT et c'est la façon dont vos appareils peuvent interagir avec d'autresAWSServices .

- Configurer votre Compte AWS (p. 19)

  Avant de pouvoir utiliserAWS IoTServices, vous devez configurer un Compte AWS . Si vous avez déjà un Compte AWS et un utilisateur IAM pour vous-même, vous pouvez les utiliser et ignorer cette étape.
- Essayez la démo interactive (p. 21)

  Cette démo est le mieux si vous voulez voir ce qu'est une baseAWS IoTpeut se passer de connecter un appareil ou de télécharger un logiciel. La démo interactive présente une solution simulée basée sur AWS IoT Core qui illustre la façon dont ils interagissent.
- Essayez le tutoriel de connexion rapide (p. 25)

  Ce tutoriel est le meilleur si vous voulez commencer à utiliser rapidementAWS IoTet voir comment cela fonctionne dans un scénario limité. Dans ce didacticiel, vous avez besoin d'un appareil et vous allez installer quelquesAWS IoTlogiciel dessus. Si vous ne disposez d'aucun appareil IoT, vous pouvez utiliser votre ordinateur personnel Windows, Linux ou macOS comme appareil pour ce didacticiel. Si vous voulez essayerAWS IoTSi vous n'avez pas d'appareil, essayez l'option suivante.
- Exploration d' AWS IoT Core services avec un tutoriel pratique (p. 34)

Ce tutoriel est le meilleur pour les développeurs qui veulent commencer avecAWS IoTafin qu'ils puissent continuer à explorer d'autres AWS IoT Core telles que le moteur de règles et les ombres. Ce didacticiel suit un processus similaire au didacticiel de connexion rapide, mais fournit plus de détails sur chaque étape pour permettre une transition plus fluide vers les didacticiels plus avancés.

Note

Si vous voulez essayer plusieurs de ces didacticiels de démarrage ou répéter le même didacticiel, vous devez supprimer l'objet chose que vous avez créé à partir d'un didacticiel antérieur avant d'en démarrer un autre. Si vous ne supprimez pas l'objet chose d'un didacticiel précédent, vous devrez utiliser un nom différent pour les didacticiels suivants. C'est parce que le nom de l'objet doit être unique dans votre compte et Région AWS .

Pour de plus amples informations sur AWS IoT Core , veuillez consulter Qu'est-ce qu' AWS IoT Core (p. 1) ?

Rubriques

Configurer votre Compte AWS

Avant d'utiliser AWS IoT Core pour la première fois, exécutez les tâches suivantes :

Si vous avez déjà un Compte AWS et un utilisateur IAM pour vous-même, vous pouvez les utiliser et passer àthe section called “Ouverture d'AWS IoTconsole” (p. 21).

Inscription à Compte AWS

Lorsque vous vous inscrivez àAWSVotre compte est automatiquement inscrit à tous les services dansAWS. Si vous disposez d'un Compte AWS déjà, ignorez cette procédure. Si vous n'avez pas de Compte AWS Utilisez la procédure suivante pour en créer un.

Vous pouvez vous attendre à passer environ 5 minutes à configurer votre Compte AWS .

Si vous n'avez pas de compte AWS, complétez les étapes suivantes pour en créer un.

Pour créer un compte AWS

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

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

Note

Enregistrez vos dépenses Compte AWS Si nécessaire, vous en aurez besoin lors de la prochaine tâche.

Créez un utilisateur et accordez-lui des autorisations

Cette procédure décrit la façon de créer un utilisateur IAM pour vous-même et d'ajouter cet utilisateur à un groupe disposant des autorisations d'administrateur provenant d'une stratégie gérée attachée. IAM estAWSqui gère les utilisateurs et l'accès àAWSAWS. Vous devez le faire afin que vous puissiez créer leAWS IoTdans votre compte et accordez-leur l'autorisation de faire ce qu'ils doivent faire.

Pour créer un administrateur pour vous-même et ajouter l'utilisateur à un groupe d'administrateurs (console)

1. Connectez-vous à la console IAM en tant que propriétaire du compte en choisissant Utilisateur racine et en saisissant l'adresse e-mail de votre compte AWS. Sur la page suivante, saisissez votre mot de passe.

   Note

   Nous vous recommandons vivement de respecter la bonne pratique qui consiste à avoir recours à l'utilisateur IAM **`Administrator`** suivant et protéger les informations d'identification de l'utilisateur racine. Connectez-vous en tant qu'utilisateur racine pour effectuer certaines tâches de gestion des comptes et des services.
2. Dans le panneau de navigation, choisissez Utilisateurs, puis Add user (Ajouter un utilisateur).
3. Dans User name (Nom d'utilisateur), entrez **`Administrator`**.
4. Cochez la case à côté de AWS Management Console access (accès à la console). Puis, sélectionnez Custom password (Mot de passe personnalisé, et entrez votre nouveau mot de passe dans la zone de texte.
5. (Facultatif) Par défaut, AWS oblige le nouvel utilisateur à créer un nouveau mot de passe lors de sa première connexion. Décochez la case en regard de User must create a new password at next sign-in (L'utilisateur doit créer un nouveau mot de passe à sa prochaine connexion) pour autoriser le nouvel utilisateur à réinitialiser son mot de passe une fois qu'il s'est connecté.
6. Choisissez Next (Suivant) Permissions (Autorisations).
7. Sous Set permissions (Accorder des autorisations), choisissez Add user to group (Ajouter un utilisateur au groupe).
8. Choisissez Create group.
9. Dans la boîte de dialogue Create group (Créer un groupe), pour Group name (Nom du groupe), tapez **`Administrators`**.
10. Choisissez Filter policies (Filtrer les stratégies), puis sélectionnez AWS managed - job function (Fonction professionnelle gérée par AWS) pour filtrer le contenu de la table.
11. Dans la liste des stratégies, cochez la case AdministratorAccess. Choisissez ensuite Create group.

    Note

    Vous devez activer l'accès des rôles et utilisateurs IAM à la facturation avant de pouvoir utiliser les autorisations `AdministratorAccess` pour accéder à la console AWS Billing and Cost Management. Pour ce faire, suivez les instructions de l'étape 1 du didacticiel portant sur comment déléguer l'accès à la console de facturation.
12. De retour dans la liste des groupes, activez la case à cocher du nouveau groupe. Choisissez Refresh si nécessaire pour afficher le groupe dans la liste.
13. Choisissez Next (Suivant) Tags (Balises).

14. (Facultatif) Ajoutez des métadonnées à l'utilisateur en associant les balises sous forme de paires clé-valeur. Pour de plus amples informations sur l'utilisation des balises dans IAM, veuillez consulter Balisage des utilisateurs et des rôles IAM dans le Guide de l'utilisateur IAM.
15. Choisissez Next (Suivant) VérificationPour afficher la liste des membres du groupe à ajouter au nouvel utilisateur. Une fois que vous êtes prêt à continuer, choisissez Create user.

Vous pouvez utiliser ce même processus pour créer d'autres groupes et utilisateurs et pour accorder à vos utilisateurs l'accès aux ressources de votre compte AWS. Pour en savoir plus sur l'utilisation des stratégies afin de limiter les autorisations d'accès des utilisateurs à certaines ressources AWS, consultez Gestion des accès et Exemples de stratégies.

Ouverture d'AWS IoTconsole

La plupart des sujets axés sur la console de cette section commencent à partir duAWS IoTconsole. Si vous n'êtes pas déjà connecté à votre Compte AWS , connectez-vous, puis ouvrez l'ongletAWS IoTconsoleet passez à la section suivante pour commencer à utiliserAWS IoT.

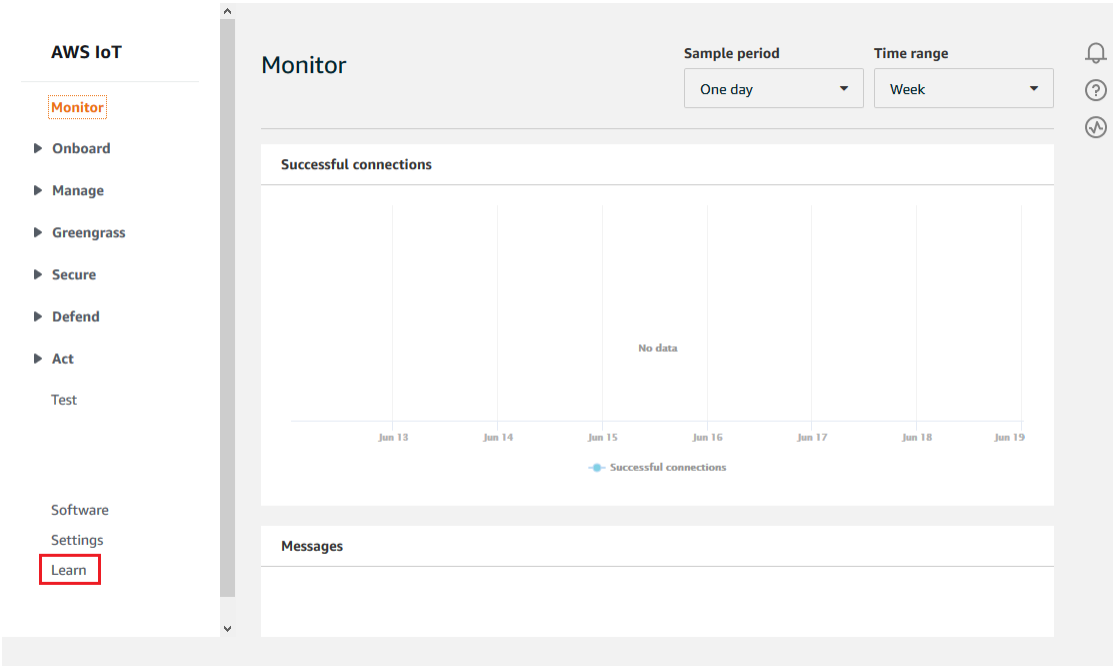
À vous d'essayer AWS IoT Core Démonstration interactive

La démo interactive montre les composants d'une solution IoT simple construite surAWS IoTet comment ils interagissent avec AWS IoT Core Services . Il ne crée pas deAWS IoTet il ne nécessite pas non plus que vous téléchargez un logiciel ou que vous faites un codage.

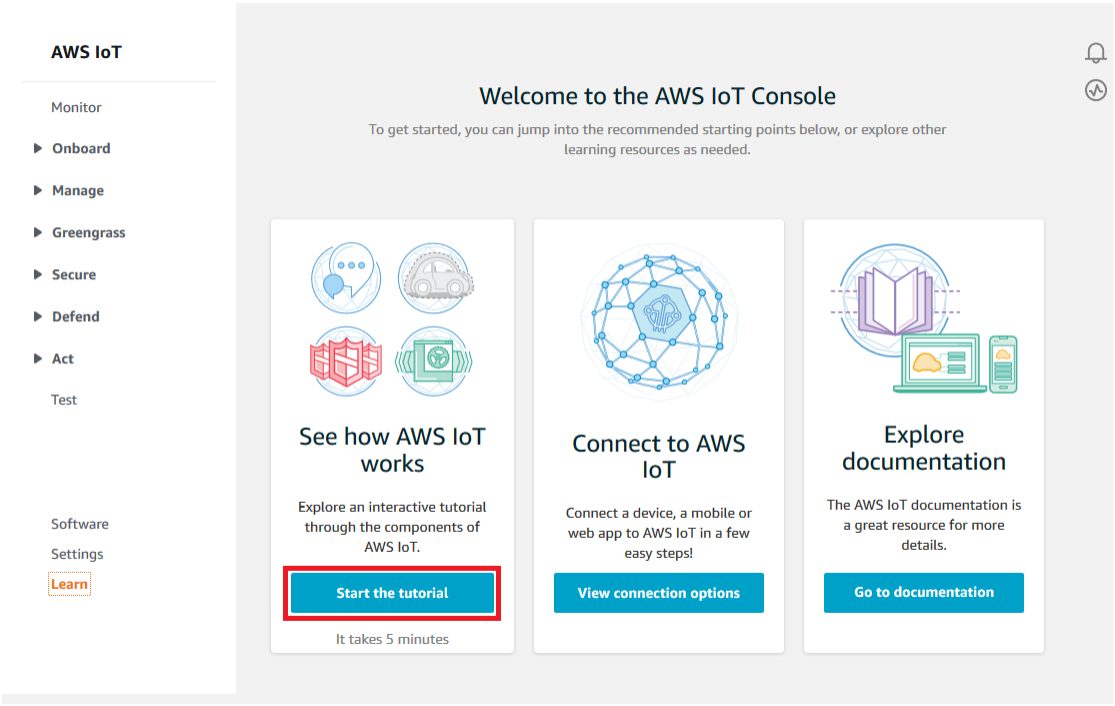
Vous pouvez vous attendre à passer environ 10 minutes avec cette démo. La tuile de la console dit que cela prend 5 minutes, et vous pouvez probablement le faire en 5 minutes, mais 10 minutes vous donnera le temps d'explorer chacune des différentes étapes plus à fond.

Pour exécuter le AWS IoT Core Démonstration interactive

1. Ouvrez vos dépensesAWS IoTconsoleDans le menu de gauche, choisissez.Procédure.



2. Dans la pageVoir commentAWS IoTŒuvresVignette, choisissezLancement du didacticiel.



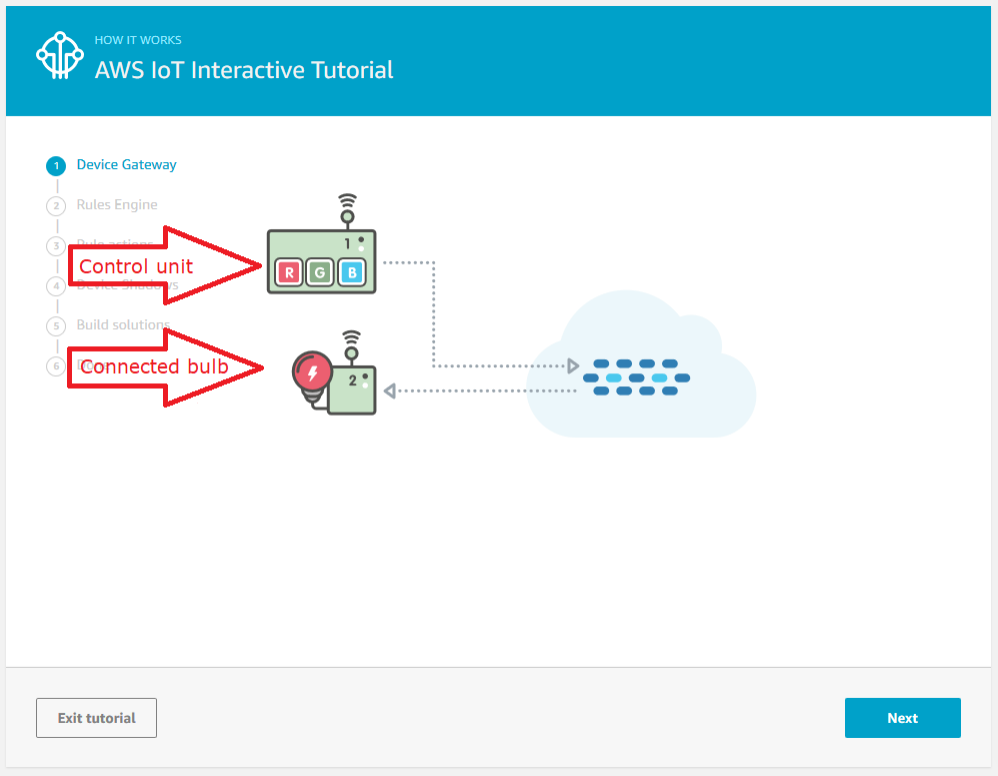
3. Dans la pageAWS IoTDidacticiel interactif, consultez la vue d'ensemble du didacticiel pour avoir une idée de ce que vous apprendrez.

   ChoisissezDidacticiel de démarrageLorsque vous êtes prêt à continuer.

4. Démo interactive : étape 1

   Lisez les instructions dans le panneau de droite, qui décrivent les composants du système IoT.

   Choisissez les différents boutons de l'unité de commande et surveillez l'effet sur l'ampoule connectée.



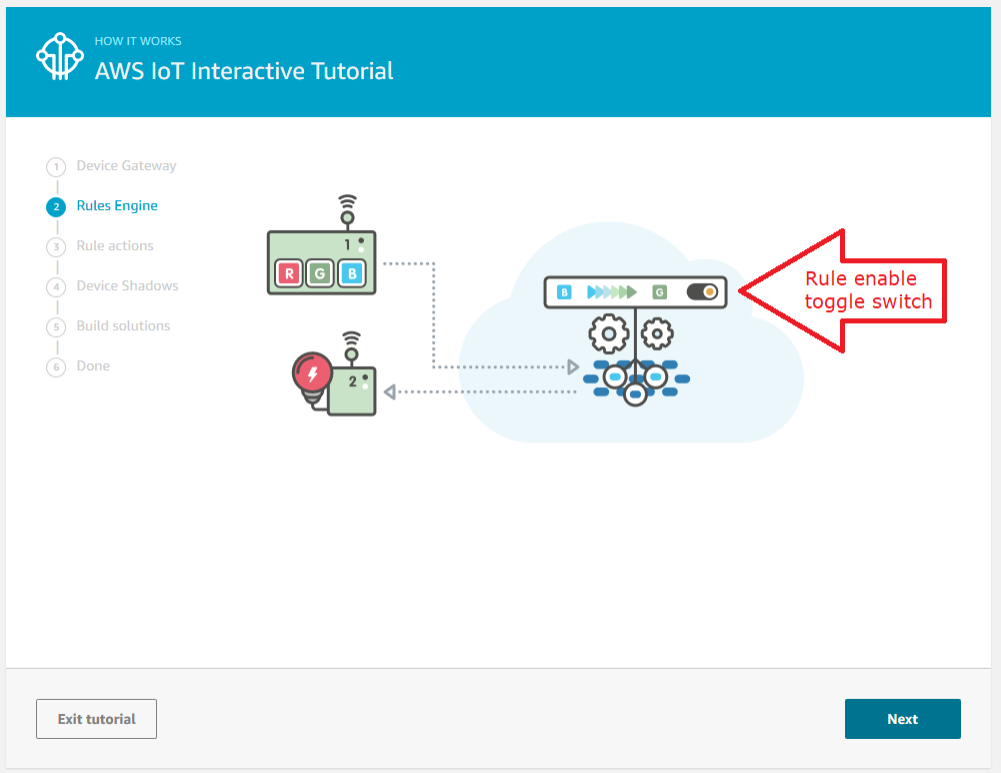
Choisissez Next pour continuer.

5. Démo interactive : étape 2

   Lisez les instructions dans le panneau de droite, qui décrivent les nouveaux composants.

   Choisissez les différents boutons de l'unité de commande et surveillez l'effet sur l'ampoule connectée.

   DéplacerActiver la règle commutateur à basculepour désactiver la règle et choisir différents boutons sur l'unité de commande pour voir comment l'ampoule se comporte différemment.



Choisissez Next pour continuer.

6. Démo interactive : étape 3

   Cette étape ajoute une autre règle. Lisez les instructions dans le panneau de droite pour comprendre la nouvelle règle, puis essayez les interactions décrites dans le panneau de droite.

   Choisissez Next pour continuer.

7. Démo interactive : étape 4

   Cette étape ajoute une ombre de périphérique àAWS IoTet un basculement de puissance à l'ampoule connectée.

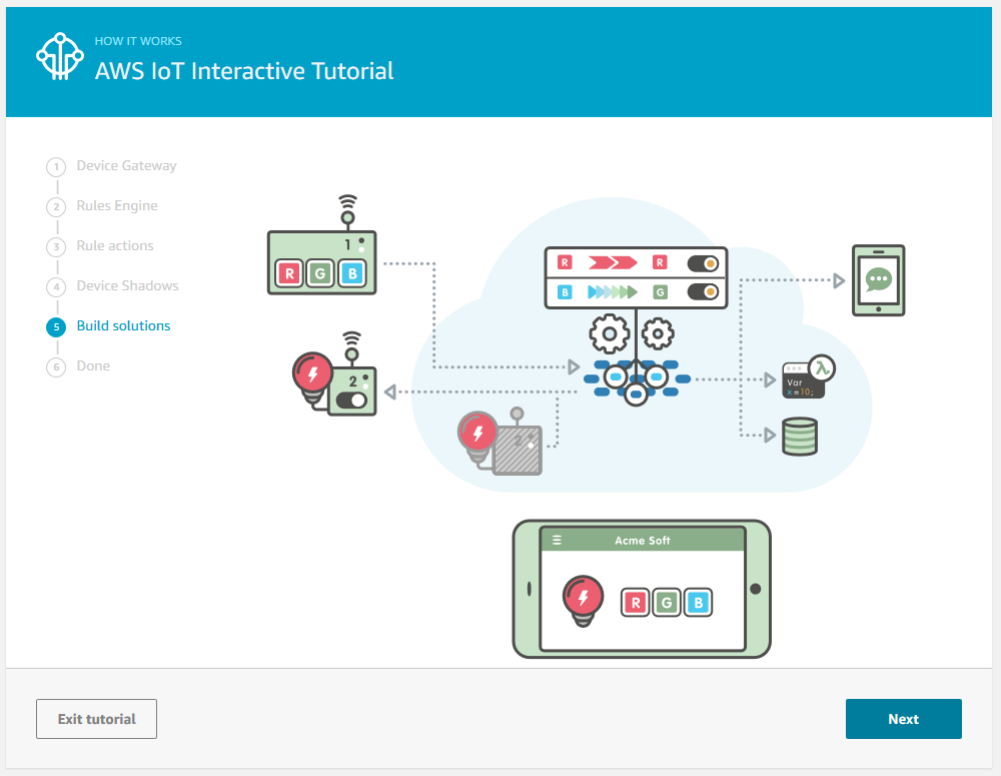
   Lisez les instructions dans le panneau de droite pour comprendre comment les nouveaux composants affecteront le fonctionnement du système, puis essayez les interactions décrites dans le panneau de droite.

   Choisissez Next pour continuer.

8. Démo interactive étape 5

   Cette étape ajoute un appareil mobile avec une application en bas de l'écran, qui interagit avec la solution IoT.

   Lisez les instructions dans le panneau de droite pour comprendre le fonctionnement des nouveaux composants, puis essayez les interactions décrites dans le panneau de droite.



Choisissez Next pour continuer.

9. Démo interactive étape 6

   C'est la dernière étape de la démonstration où vous pouvez examiner et explorer le fonctionnement de cette solution IoT.

   ChoisissezMise en routePour continuer.

10. Dans la pageEnregistrer un objetPage, choisissezAnnulerPour quitter la démonstration.

À vous d'essayerAWS IoTConnexion rapide

Dans ce didacticiel, vous allez créer votre premier objet, y connecter un appareil et le regarder envoyer des messages MQTT.

Vous pouvez vous attendre à passer 15-20 minutes sur ce tutoriel.

Ce tutoriel est le meilleur pour les personnes qui veulent commencer à utiliser rapidementAWS IoTpour voir comment cela fonctionne dans un scénario limité. Si vous cherchez un exemple qui vous aidera à démarrer afin de pouvoir explorer plus de fonctionnalités et de services, essayezExploration d' AWS IoT Core services dans le tutoriel pratique (p. 34).

Dans ce didacticiel, vous allez télécharger et exécuter un logiciel sur un périphérique qui se connecte à unobjet d'un objetin AWS IoT Core dans le cadre d'une solution IoT très petite. Le périphérique peut être un périphérique IoT, tel qu'un Raspberry Pi, ou il peut également s'agir d'un ordinateur fonctionnant sous Linux, OS et OSX, ou Windows. Si vous cherchez à connecter un périphérique Long Range WAN (LoRaWan) àAWS IoT, reportez-vous au tutorielConnexion de périphériques et de passerelles à AWS IoT Core pour LoRaWAN (p. 1062).

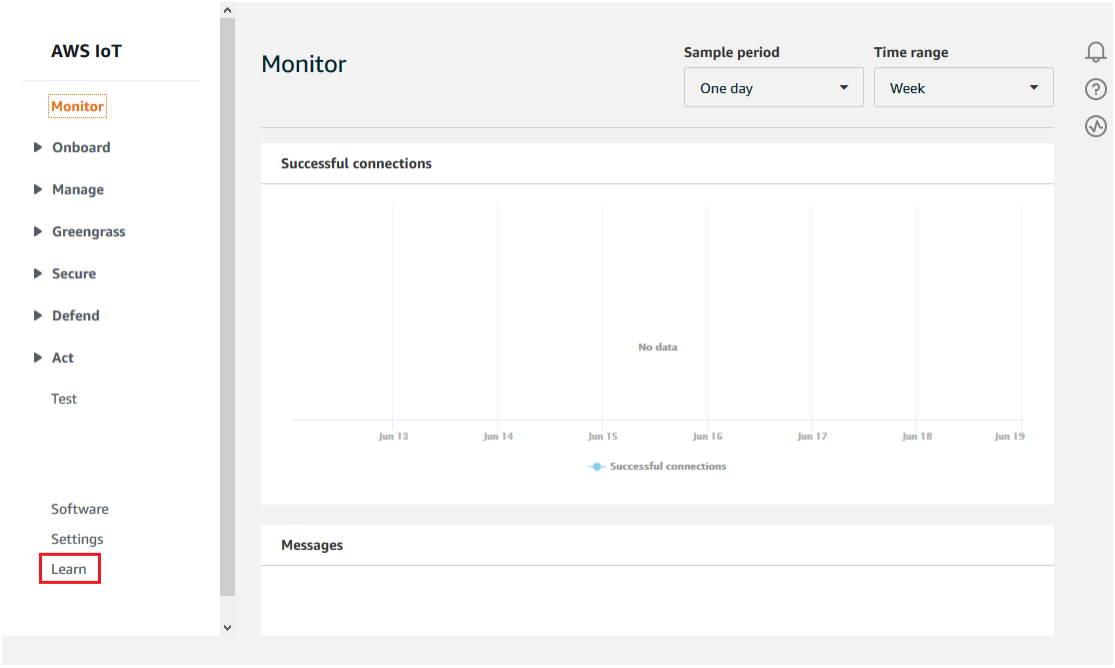
Si votre appareil prend en charge un navigateur capable d'exécuter leAWS IoTconsole, nous vous recommandons de compléter ce didacticiel sur cet appareil.

> Note
>
> Si votre appareil ne dispose pas d'un navigateur compatible, suivez ce didacticiel sur un ordinateur. Lorsque la procédure vous demande de télécharger le fichier, téléchargez-le sur votre ordinateur, puis transférez le fichier téléchargé sur votre appareil à l'aide de Secure Copy (SCP) ou d'un processus similaire.

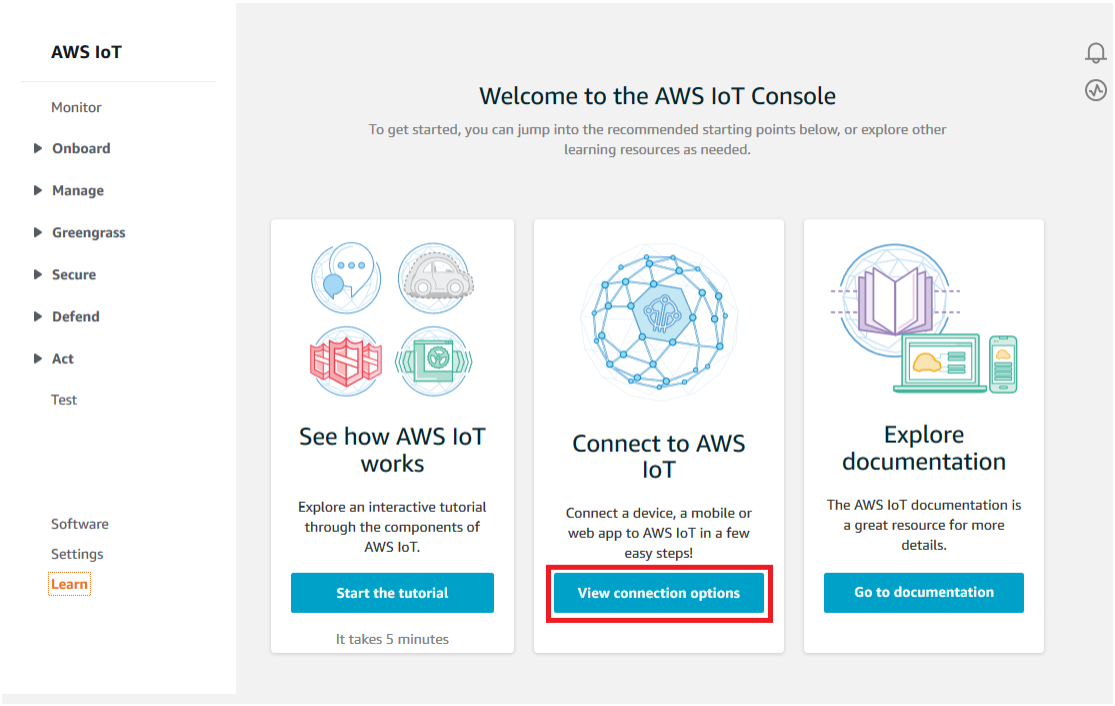
Étape 1. Lancement du didacticiel

Si possible, effectuez cette procédure sur votre appareil ; sinon, soyez prêt à transférer un fichier vers votre appareil plus tard dans cette procédure.

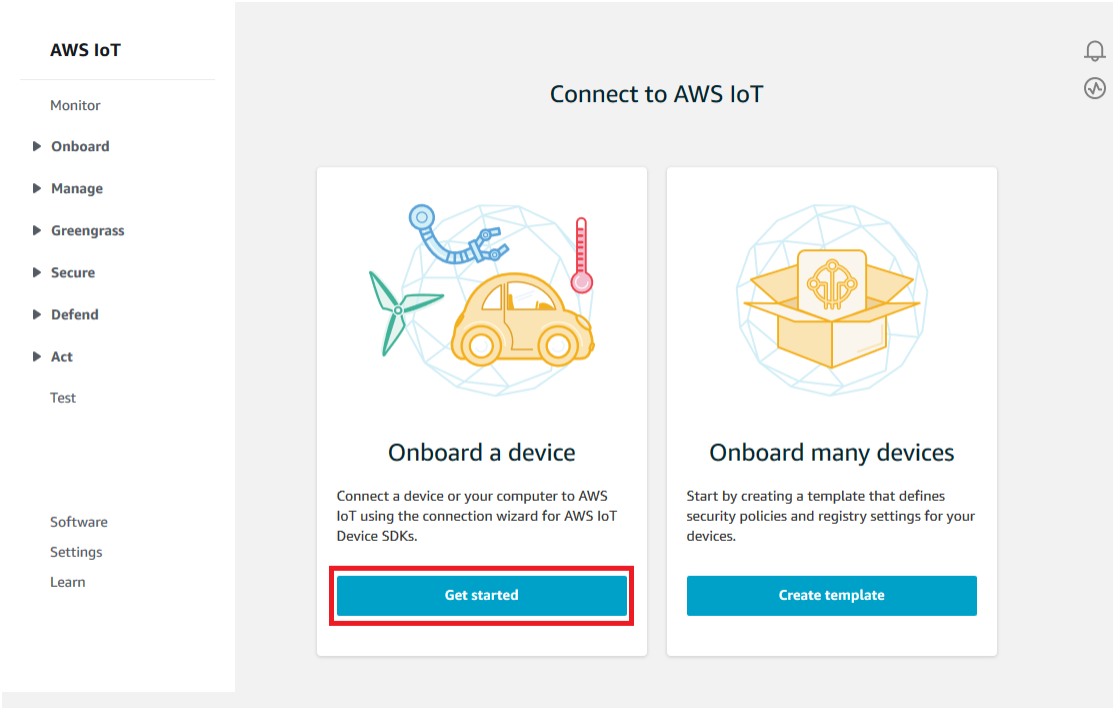
1. Ouvrez vos dépensesAWS IoTconsoleDans le menu de gauche, choisissez.Procédure.



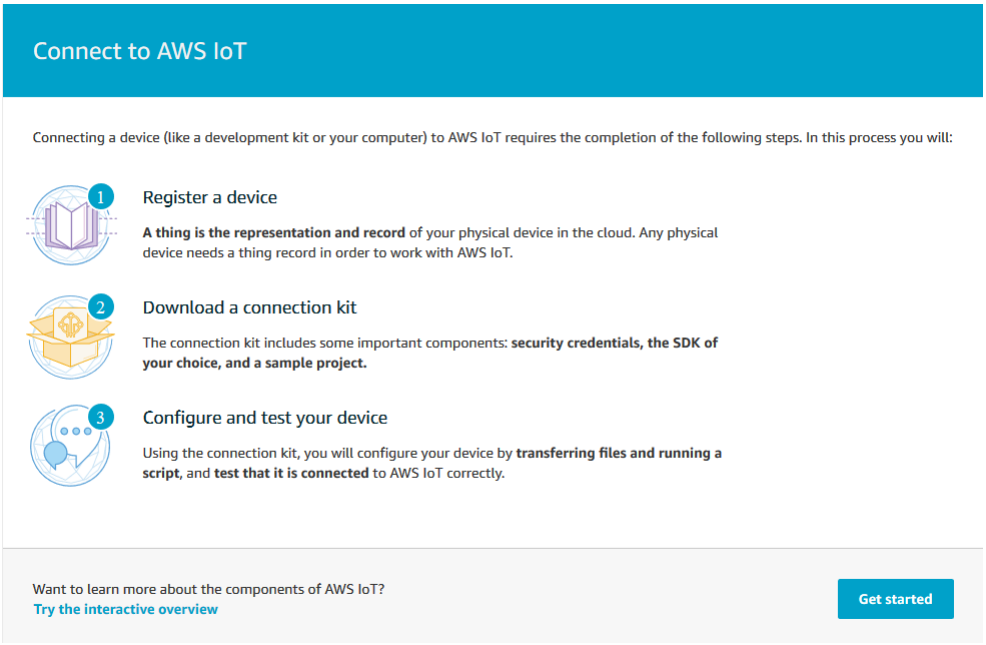
2. Dans la pageConnexion àAWS IoTVignette, choisissezAfficher les options de connexion.



3. DansIntégrer à un appareilVignette, choisissezMise en route.



4. Passez en revue les étapes qui décrivent ce que vous allez faire dans ce didacticiel. Lorsque vous êtes prêt à poursuivre, choisissez l'optionMise en route.



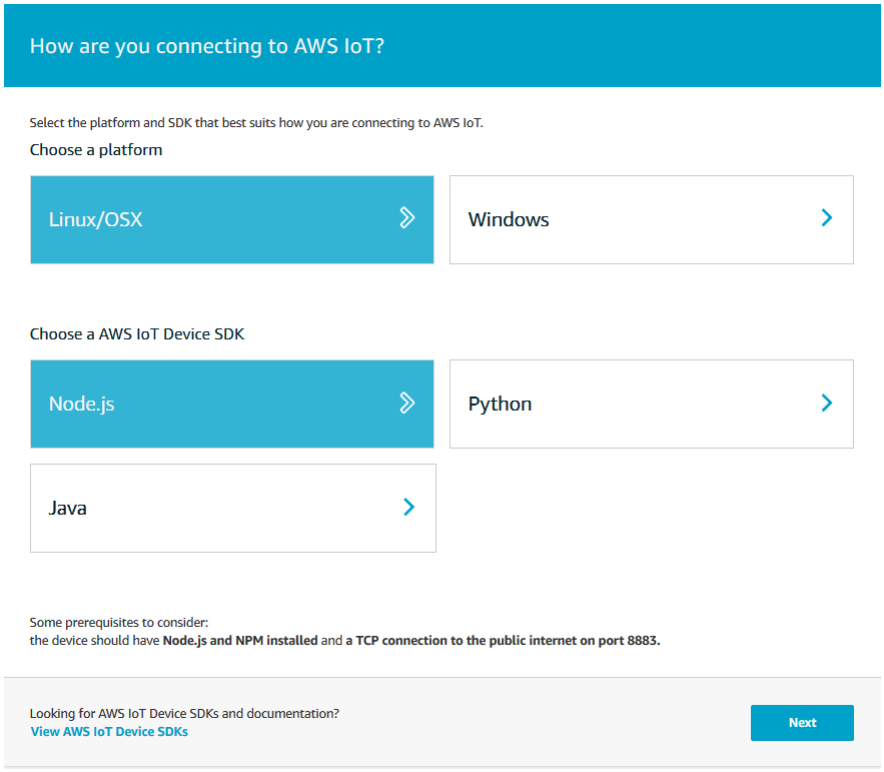
Étape 2. Créer un objet d'objet

1. Dans la pageComment vous connectez-vous àAWSIoT ?, choisissez la plateforme et la langue duAWS IoTKit SDK des appareils que vous souhaitez utiliser. Cet exemple utilise la plate-forme Linux/OSX et le SDK Node.js.

   Note

   Assurez-vous de vérifier la liste des logiciels requis par le SDK choisi au bas de la page de la console.
   Vous devez avoir installé le logiciel requis sur votre ordinateur cible avant de passer à l'étape suivante.

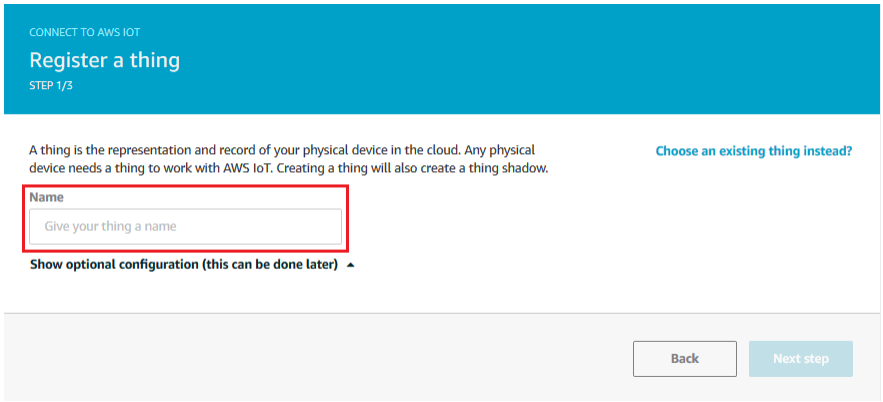
   Après avoir choisi le langage SDK de la plate-forme et de l'appareil, choisissezSuivant.



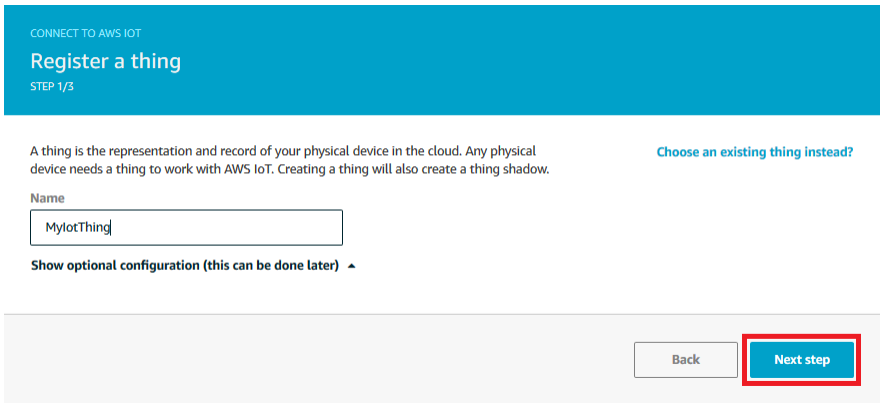
2. DansNomEntrez le nom de votre objet objet objet. Le nom de chose utilisé dans cet exemple est**`MyIotThing`**.

   Important

   Vérifiez le nom de votre truc avant de continuer.
   Un nom de chose ne peut pas être modifié après la création de l'objet chose. Si vous souhaitez changer le nom d'un objet objet objet, vous devez créer un objet objet objet objet objet objet avec le nom de objet correct, puis supprimer celui portant le nom incorrect.



3. Après avoir donné un nom à votre objet truc, choisissezÉtape suivante.

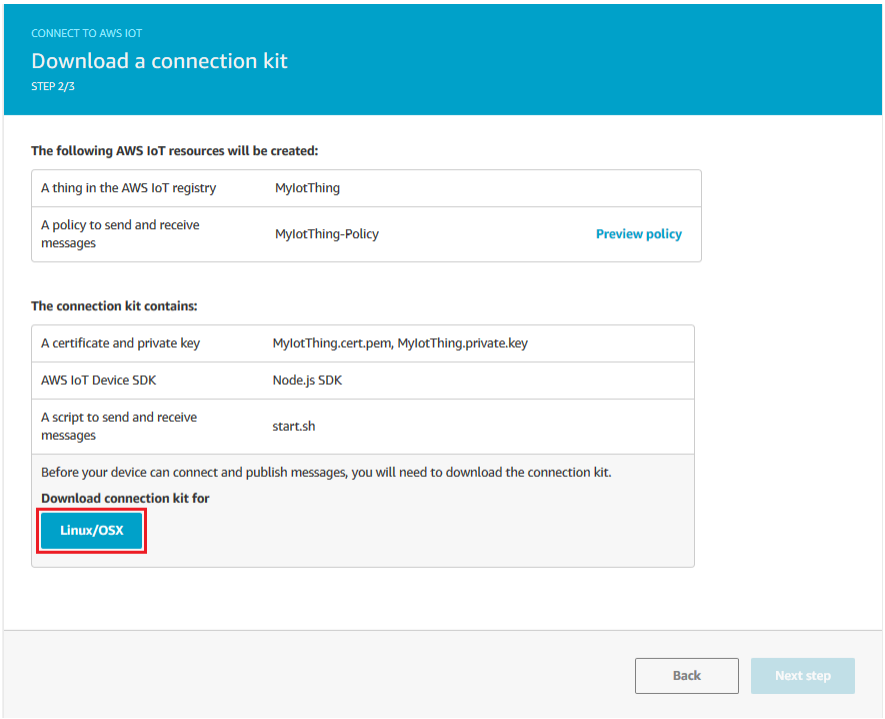


Étape 3. Télécharger des fichiers sur votre appareil

Cette page s'affiche aprèsAWS IoTa créé le kit de connexion, qui comprend les fichiers et ressources suivants dont votre appareil a besoin :

- Les fichiers de certificat de l'objet utilisés pour authentifier l'appareil
- Une ressource de stratégie permettant d'autoriser votre objet de chose à interagir avecAWS IoT
- Le script pour télécharger le fichierAWSKit SDK des appareils et exécutez l'exemple de programme sur votre appareil

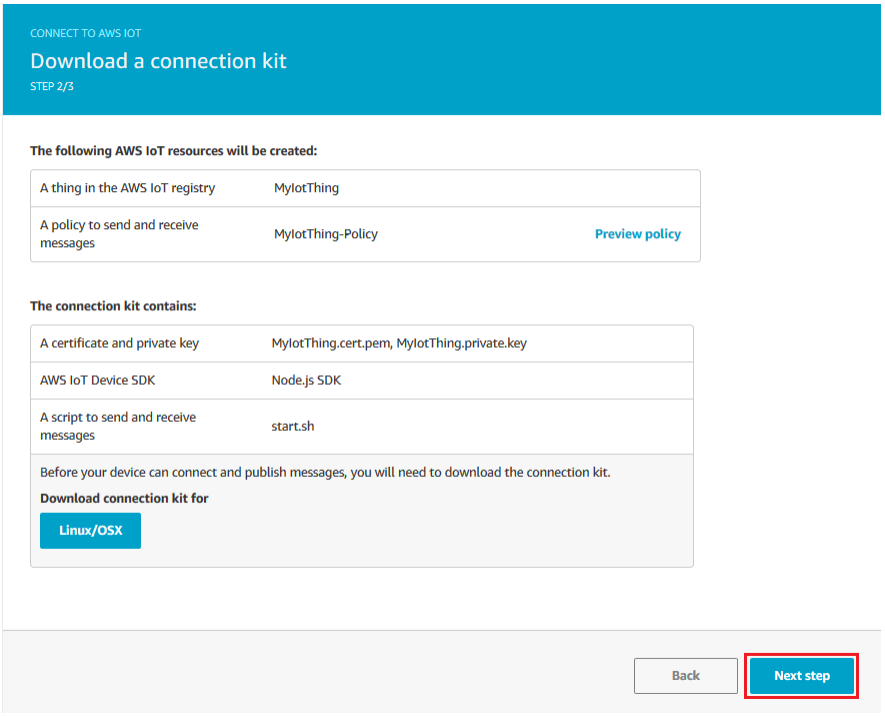
1. Lorsque vous êtes prêt à poursuivre, choisissez le boutonTélécharger le kit de connexion pourpour télécharger le kit de connexion de la plate-forme que vous avez choisie précédemment.



2. Si vous exécutez cette procédure sur votre appareil, enregistrez le fichier du kit de connexion dans un répertoire à partir duquel vous pouvez exécuter des commandes de ligne de commande.

Si vous n'exécutez pas cette procédure sur votre appareil, enregistrez le fichier du kit de connexion dans un répertoire local, puis transférez le fichier sur votre appareil.

3. Après avoir installé le fichier du kit de connexion sur l'appareil, poursuivez le didacticiel en choisissantÉtape suivante.

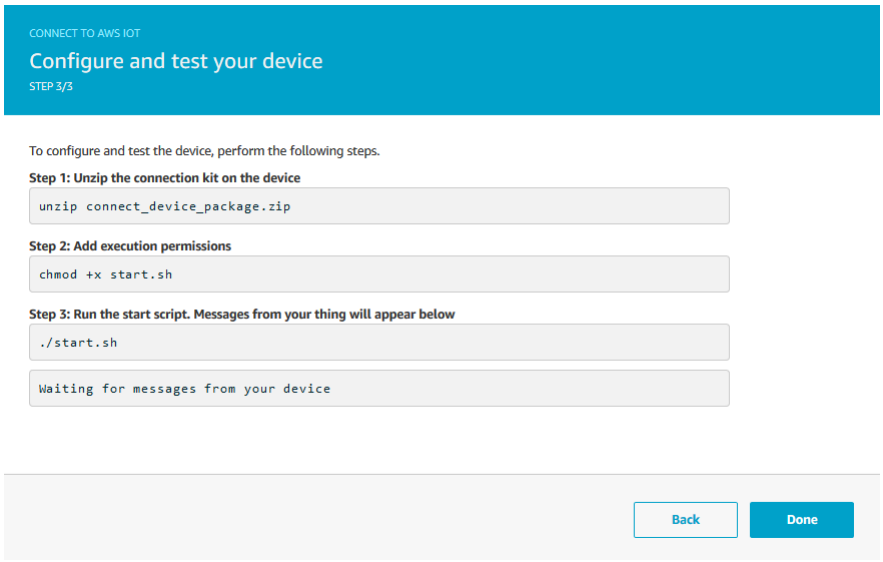


Étape 4. Exécutez l'exemple

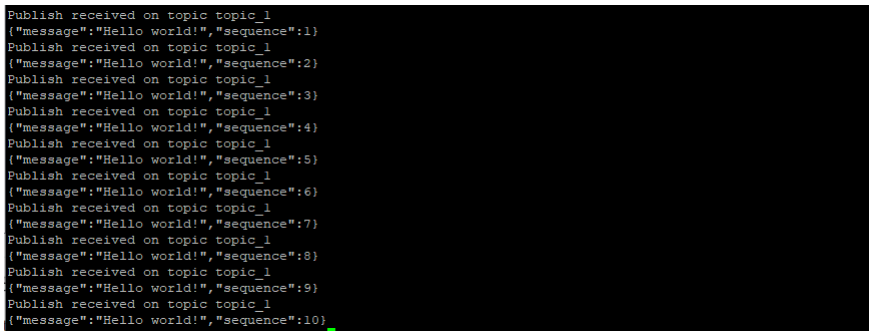
Cette procédure est effectuée dans un terminal ou une fenêtre de commande de votre appareil tout en suivant les instructions affichées dans la console. Les commandes affichées dans la console sont celles du système d'exploitation que vous avez choisi dansthe section called “Étape 2. Créer un objet d'objet” (p. 28). Ceux présentés ici concernent les systèmes d'exploitation Linux/OSX.

1. Dans une fenêtre de terminal ou de commande de votre appareil, dans le répertoire contenant le fichier du kit de connexion, effectuez les étapes indiquées dans laAWS IoTconsole

   Si vous utilisez une fenêtre de commande Windows PowerShell et que la commandeunzipne fonctionne pas, remplacezunzipavecexpand-archiveet réessayez la ligne de commande.



2. Dans le terminal ou la fenêtre de commande de votre appareil, après avoir entré la commande deÉtape 3Dans la console, vous devriez afficher une sortie similaire à ceci. Cette sortie provient des messages que le programme envoie et reçoit ensuite de AWS IoT Core . Pendant que le programme exemple communique avec AWS IoT Core , vous ne verrez aucune activité dans la console. Pour voir l'activité dans la console pendant que vous exécutez l'exemple de programme, consultezÉtape 4de cette procédure. Parfois, au lieu de`topic_1`, vous pouvez voir un message du terminal indiquant qu'il a été reçu à partir de la rubrique**`sdk/test/`*`SDK_programming_language`***, où les*`SDK_Programming_Langage`*peut être Python, JavaScript ou Java.



3. Vous pouvez répéter les commandes deÉtape 3/3dans la console de cette procédure), pour exécuter à nouveau l'exemple de programme.
4. (Facultatif) Si vous souhaitez afficher les messages de votre client IoT dans laAWS IoTconsole, ouvrezClient MQTTsur leTestPage de laAWS IoTconsole DansClient MQTT, abonnez-vous à**`sdk/test/`*`SDK_programming_language`***. Le nom de la rubrique dépend du langage de programmation du SDK que vous avez choisi dansÉtape 1/. Les noms de rubrique possibles sont les suivants et sont sensibles à la casse.

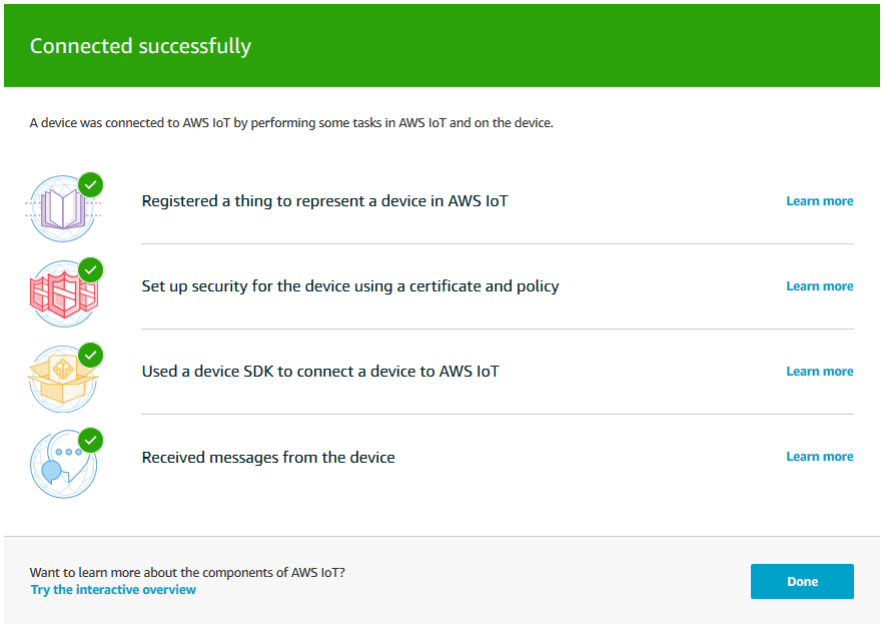
   - Pour le AWS IoT Device SDK , le sujet est**`sdk/test/javascript`**.
   - Pour Python AWS IoT Device SDK , le sujet est**`sdk/test/Python`**.
   - Pour Java AWS IoT Device SDK , le sujet est**`sdk/test/java`**.
5. Après vous être abonné à la rubrique de test, exécutez ce programme sur votre appareil./start.shComme décrit à l'étape précédente. Pour de plus amples informations, veuillez consulterthe section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” (p. 62)Pour plus d'informations, consultez.

Après avoir exécuté./start.sh, des messages s'affichent dans le client MQTT similaires à ce qui suit s'affichent :

```
{
  "message": "Hello World!",
  "sequence": 10
}
```

La `.sequence`nombre incrémente d'un à chaque fois qu'un nouveau`Hello World`est reçu et s'arrête lorsque vous terminez le programme.

6. Une fois que vous avez terminé d'exécuter le programme sur votre appareil, dans leAWS IoTConsole, choisissezTerminépour terminer le didacticiel et voir ce résumé.



Étape 5. Explorer plus loin

Voici quelques idées à explorerAWS IoTplus loin après avoir terminé le démarrage rapide.

- Afficher les messages MQTT dans le client MQTT

  À partir deAWS IoTconsole, vous pouvez ouvrir leClient MQTTsur leTestPage de laAWS IoTconsole DansClient MQTT, abonnez-vous à**#**, puis, sur votre appareil, exécutez le programme./start.shComme décrit à l'étape précédente. Pour plus d'informations, consultez the section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” (p. 62).

- the section called “À vous d'essayer AWS IoT Core Démonstration interactive” (p. 21)

  Pour démarrer le didacticiel interactif, à partir de laProcédurePage de laAWS IoT, dans laVoir commentAWS IoTŒuvresVignette, choisissezLancement du didacticiel.

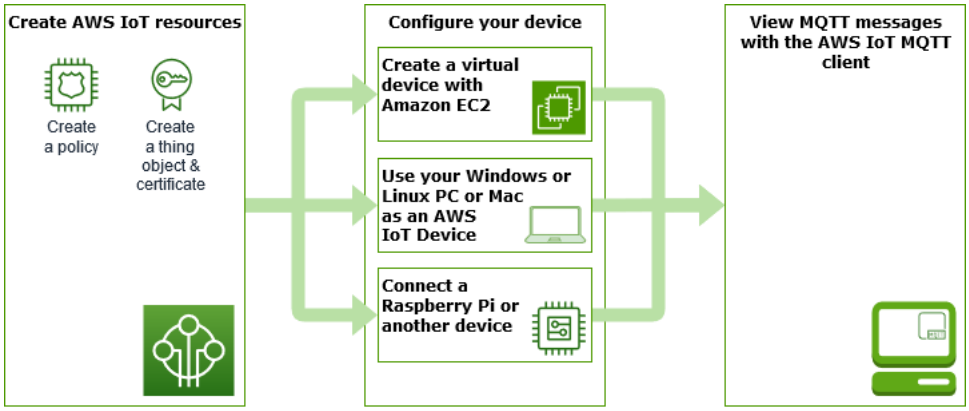
- Préparez-vous à explorer d'autres tutoriels (p. 34)

  Ce démarrage rapide vous donne juste un échantillon deAWS IoT. Si vous voulez explorerAWS IoTet découvrez les fonctionnalités qui en font une plateforme de solution IoT puissante, commencez à préparer votre plateforme de développement enExploration d' AWS IoT Core services dans le tutoriel pratique (p. 34).

Exploration d' AWS IoT Core services dans le tutoriel pratique

Dans ce didacticiel, vous allez installer le logiciel et créer l'AWS IoTressources nécessaires pour connecter un périphérique àAWS IoTafin qu'il puisse envoyer et recevoir des messages MQTT avec AWS IoT Core . Vous verrez les messages dans le client MQTT dans le répertoireAWS IoTconsole

Vous pouvez vous attendre à passer 20-30 minutes sur ce tutoriel. Si vous utilisez un périphérique IoT ou un Raspberry Pi, ce didacticiel peut prendre plus de temps si, par exemple, vous devez installer le système d'exploitation et configurer l'appareil.



Ce tutoriel est le meilleur pour les développeurs qui veulent commencer avecAWS IoTafin qu'ils puissent continuer à explorer des fonctionnalités plus avancées, telles que le moteur de règles et les ombres. Ce tutoriel vous prépare à continuer à apprendre sur AWS IoT Core et comment il interagit avec d'autresAWSen expliquant les étapes de manière plus détaillée que le didacticiel de démarrage rapide. Si vous recherchez juste un rapide,Bonjour le monde, l'expérience, essayez laÀ vous d'essayerAWS IoTConnexion rapide (p. 25).

Après avoir configuré votre Compte AWS andAWS IoT, vous allez suivre ces étapes pour voir comment connecter un appareil et lui faire envoyer des messages àAWS IoT.

Étapes suivantes

- Choisissez l'option d'appareil qui vous convient le mieux (p. 34)
- the section called “CréerAWS IoTresources” (p. 35)si vous n'allez pas créer un appareil virtuel avec Amazon EC2.
- the section called “Configurer votre appareil” (p. 39)
- the section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” (p. 62)

Pour de plus amples informations sur AWS IoT Core , veuillez consulter Qu'est-ce qu’ AWS IoT Core (p. 1) ?

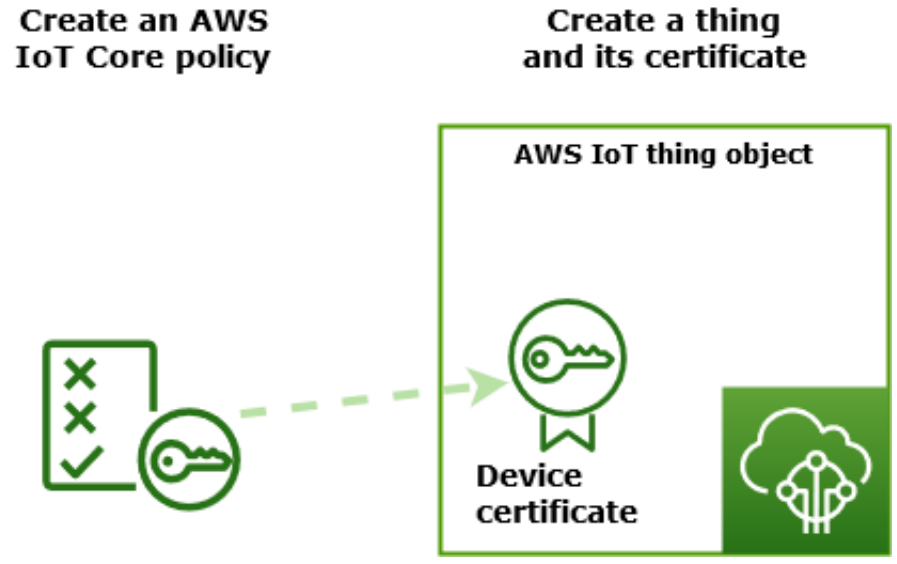
Quelle option d'appareil vous convient le mieux ?

Si vous ne savez pas quelle option choisir, utilisez la liste suivante des avantages et des inconvénients de chaque option pour vous aider à choisir celle qui vous convient le mieux.

Option	Cela pourrait être une bonne option si :	Cela peut ne pas être une bonne option si :
the section called “Créez un appareil virtuel avec Amazon EC2” (p. 39)	• Vous n'avez pas votre propre appareil à tester. • Vous ne souhaitez pas installer de logiciel sur votre propre système. • Vous voulez tester sur un système d'exploitation Linux.	• Vous n'êtes pas à l'aise avec les commandes de ligne de commande. • Vous ne voulez pas engager d'autresAWSFrais. • Vous ne voulez pas tester sur un système d'exploitation Linux.
the section called “Utilisez votre PC Windows ou Linux ou Mac en tant queAWS IoTAppareil” (p. 47)	• Vous ne voulez pas engager d'autresAWSFrais. • Vous ne souhaitez pas configurer d'autres appareils.	• Vous ne souhaitez pas installer de logiciel sur votre ordinateur personnel. • Vous voulez une plateforme de test plus représentative.
the section called “Connect un Raspberry Pi ou un autre appareil” (p. 53)	• Vous voulez testerAWS IoTavec un périphérique réel. • Vous avez déjà un appareil pour tester. • Vous avez de l'expérience dans l'intégration du matériel dans les systèmes.	• Vous ne voulez pas acheter ou configurer un appareil juste pour l'essayer. • Vous voulez testerAWS IoTaussi simplement que possible, pour l'instant.

CréerAWS IoTresources

Dans ce didacticiel, vous allez créer l'objetAWS IoTdont un périphérique a besoin pour se connecter àAWS IoTet échanger des messages.



1. Création d'unAWS IoT, qui autorisera votre appareil à interagir avecAWS IoTServices .
2. Crée un objet objet d'objets dansAWS IoTet son certificat de périphérique X.509, puis joignez le document de stratégie. L'objet chose est la représentation virtuelle de votre appareil dans leAWS

IoTRegistre. Le certificat authentifie votre appareil sur AWS IoT Core , et le document de stratégie autorise votre appareil à interagir avecAWS IoT.

Note

Si vous prévoyez dethe section called “Créez un appareil virtuel avec Amazon EC2” (p. 39)Vous pouvez ignorer cette page et continuer àthe section called “Configurer votre appareil” (p. 39). Vous allez créer ces ressources lorsque vous créez votre objet virtuel.

Ce didacticiel utilise .AWS IoTpour créer la consoleAWS IoTAWS. Si votre appareil prend en charge un navigateur Web, il peut être plus facile d'exécuter cette procédure sur le navigateur Web de l'appareil, car vous pourrez télécharger les fichiers de certificat directement sur votre appareil. Si vous exécutez cette procédure sur un autre ordinateur, vous devez copier les fichiers de certificat sur votre appareil avant qu'ils puissent être utilisés par l'exemple d'application.

Création d'une stratégie AWS IoT

Les certificats X.509 sont utilisés pour authentifier votre appareil avec AWS IoT Core .AWS IoTsont attachées au certificat qui authentifie le périphérique afin de déterminer lesAWS IoTLes opérations, telles que l'abonnement à des rubriques MQTT ou la publication dans ces rubriques, que l'appareil est autorisé à exécuter. Votre appareil présente son certificat lorsqu'il se connecte et envoie des messages à AWS IoT Core .

Au cours de cette procédure, vous créerez une stratégie qui permettra à votre appareil d'exécuter l'AWS IoTnécessaires pour exécuter l'exemple de programme. Vous devez créer leAWS IoTd'abord, afin que vous puissiez l'attacher au certificat de périphérique que vous allez créer ultérieurement.

Pour créer une stratégie AWS IoT

1. Dans le menu de gauche, choisissezSecure, puisStratégies. Sur la page, Vous n'avez pas encore de stratégie, choisissez Créer une stratégie.

   Si votre compte possède des stratégies existantes, choisissezCréer.
2. Sur la page Create a policy (Créer une stratégie) :

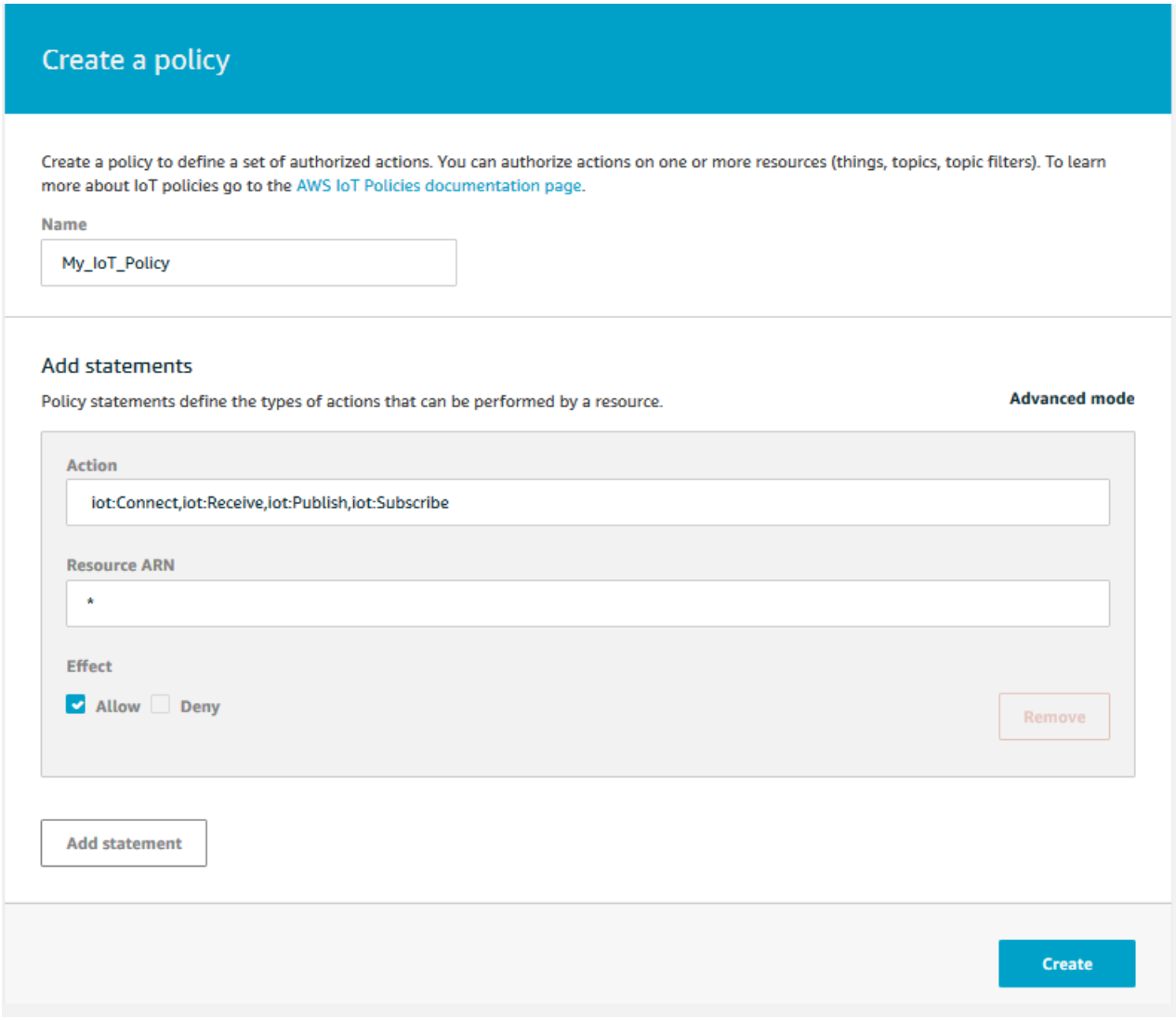
   1. Dans le champ Nom entrez un nom pour la stratégie (par exemple, **`My_Iot_Policy`**). N'utilisez pas d'informations personnelles identifiables dans vos noms de stratégie.
   2. Dans le champ Action, saisissez **`iot:Connect,iot:Receive,iot:Publish,iot:Subscribe`**. Il s'agit des actions que le périphérique aura besoin d'une autorisation pour effectuer lorsqu'il exécute l'exemple de programme à partir du SDK de périphérique.

      Pour plus d'informations sur les stratégies IoT, consultezStratégies AWS IoT Core (p. 270).
   3. Dans le champ ARN de ressource, saisissez **`*`**. Cela sélectionne n'importe quel client (périphérique).

      Note

      Dans ce démarrage rapide, le caractère générique (*) est utilisé pour plus de simplicité. Pour plus de sécurité, vous devez limiter les clients (appareils) autorisés à se connecter et à publier des messages en spécifiant ARN du client (nom de ressource Amazon) au lieu du caractère générique en tant que ressource. Voici le format des ARN de clients : `arn:aws:iot:`*`your-region`*`:`*`your-aws-account`*`:client/`*`my-client-id`* Cependant, vous devez d'abord créer la ressource (périphérique client, shadow de chose, etc.) avant de pouvoir affecter son ARN à une stratégie.
   4. Sélectionnez la case à cocher Autoriser.

      Ces valeurs permettent à tous les clients auxquels cette stratégie est attachée à leur certificat d'exécuter les actions répertoriées dans la sectionActionfield.



3. Une fois que vous avez entré les informations relatives à votre stratégie, choisissez Créer.

Pour plus d'informations, consultez Stratégies IAM (p. 320).

Créer un objet d'objet

Périphériques connectés àAWS IoTsont représentées parobjets d'objetsdans leAWS IoTRegistre. Aobjet d'un objetreprésente un appareil spécifique ou une entité logique. Il peut s'agir dun appareil physique ou un capteur (par exemple, une ampoule ou un interrupteur de lumière sur le mur). Il peut également s'agir dune entité logique, comme une instance d'une application ou une entité physique qui ne se connecte pas àAWS IoT, mais est associée à d'autres dispositifs qui se connectent (par exemple, une voiture dotée de capteurs de moteur ou d'un ordinateur de bord).

Pour créer un objet dans le répertoireAWS IoTconsole

1. DansAWS IoTconsoleDans le menu de gauche, choisissezGérer, puis choisissezObjets.
2. Dans la pageObjetsPage, choisissezCréation d'objets.

3. Dans la pageCréation d'objetsPage, choisissezCréer un objet unique, puis choisissezSuivant.
4. Dans la pageSpécifier les propriétés de chose, pourNom de l'objet, saisissez un nom pour votre objet, tel que**`MyIotThing`**.

   Lorsque vous nommez des objets, choisissez le nom soigneusement, car vous ne pouvez pas changer le nom d'un objet une fois qu'il est créé.

   Pour changer le nom d'un objet, vous devez créer un objet, lui donner un nouveau nom, puis supprimer l'ancien objet.

   > **Note**
   >
   > N'utilisez pas d'informations personnelles identifiables dans votre nom d'objet. Le nom de chose peut apparaître dans les communications et les rapports non chiffrés.

5. Ne remplissez pas les autres champs de cette page. Choisissez Suivant.
6. Dans la pageConfiguration du certificat d'appareil -facultatifPage, choisissezGénérer automatiquement un nouveau certificat (recommandé). Choisissez Suivant.
7. Dans la pageJoindre des stratégies au certificat -facultatif, sélectionnez la stratégie que vous avez créée dans la section précédente. Dans cette section, la politique a été nommée**`My_Iot_Policy`**. ChoisissezCréation d'un objet.
8. Dans la pageTélécharger les certificats et les clésPage :

   1. Téléchargez chacun des fichiers de certificat et de clé et enregistrez-les pour plus tard. Vous devrez installer ces fichiers sur votre appareil.

      Lorsque vous enregistrez vos fichiers de certificat, donnez-leur les noms dans le tableau suivant. Ce sont les noms de fichiers utilisés dans les exemples ultérieurs.

      Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Clé privée	`private.pem.key`
Clé publique	(non utilisé dans ces exemples)
Certificat de l'appareil	`device.pem.crt`
Certificat racine de l'autorité de certification	`Amazon-root-CA-1.pem`

   2. Téléchargez le fichier d'autorité de certification racine pour ces fichiers en sélectionnant l'optionTéléchargementdu fichier de certificat d'autorité de certification racine qui correspond au type de point de terminaison de données et de suite de chiffrement que vous utilisez. Dans ce didacticiel, choisissezTéléchargementà droite deClé RSA 2048 bits : Amazon Root CA 1et téléchargeClé RSA 2048 bits : Amazon Root CA 1Fichier de certificat.

      > **Important**
      >
      > Vous devez enregistrer les fichiers de certificat avant de quitter cette page. Après avoir quitté cette page dans la console, vous n'aurez plus accès aux fichiers de certificat. Si vous avez oublié de télécharger les fichiers de certificat que vous avez créés à cette étape, vous devez quitter cet écran de console, accéder à la liste des éléments de la console, supprimer l'objet objet objet que vous avez créé, puis redémarrer cette procédure dès le début.

   3. Sélectionnez Done (Effectué).

Après avoir terminé cette procédure, vous devriez voir le nouvel objet chose dans votre liste de choses.

Configurer votre appareil

Cette section décrit comment configurer votre appareil pour qu'il se connecte àAWS IoT. Si vous voulez commencer avecAWS IoTmais que vous n'avez pas encore d'appareil, vous pouvez créer un périphérique virtuel à l'aide d'Amazon EC2 ou utiliser votre PC Windows ou Mac comme appareil IoT.

Sélectionnez la meilleure option d'appareil que vous pouvez essayerAWS IoT. Bien sûr, vous pouvez les essayer tous, mais essayez seulement un à la fois. Si vous n'êtes pas sûr de l'option d'appareil qui vous convient le mieux, consultez comment choisirquelle option de périphérique est la meilleure (p. 34), puis revenez à cette page.

Options de l'appareil

- Créez un appareil virtuel avec Amazon EC2 (p. 39)
- Utilisez votre PC Windows ou Linux ou Mac en tant queAWS IoTAppareil (p. 47)
- Connect un Raspberry Pi ou un autre appareil (p. 53)

Créez un appareil virtuel avec Amazon EC2

Dans ce didacticiel, vous allez créer une instance Amazon EC2 qui servira de périphérique virtuel dans le cloud.

Pour suivre ce didacticiel, vous devez disposer d'une commande Compte AWS . Si vous n'en avez pas, suivez les étapes décrites dansConfigurer votre Compte AWS (p. 19)Avant de continuer.

Dans ce didacticiel, vous allez :

- Configurer une instance Amazon EC2 (p. 39)
- Installez Git, Node.js et configurez leAWS CLI (p. 40)
- CréerAWS IoTressources pour votre périphérique virtuel (p. 42)
- Installer le kit SDK des appareils AWS IoT pour Javascript (p. 45)
- Exécuter l'exemple d'application (p. 45)
- Affichez les messages de l'exemple d'application dans leAWS IoTconsole (p. 46)

Configurer une instance Amazon EC2

Les étapes suivantes vous expliquent comment créer une instance Amazon EC2 qui agira en tant que périphérique virtuel à la place d'un périphérique physique.

Pour lancer une instance

1. Ouvrez la console Amazon EC2 sur https://console.aws.amazon.com/ec2/.
2. Sur le tableau de bord de la console, sélectionnez Launch Instance.
3. La .Étape 1 : Sélection d'une Amazon Machine Image (AMI)La page affiche une liste des configurations de base, appeléesAmazon Machine Images (AMI), qui servent de modèles pour votre instance. Sélectionnez une version HVM d'Amazon Linux 2,tels queAMI Amazon Linux 2 (HVM), type de volume SSD. Notez que cette AMI est marquée comme « Éligible à l'offre gratuite ».
4. Sur la page Choisir un type d'instance, vous pouvez sélectionner la configuration matérielle de votre instance. Sélectionnez le type `t2.micro` qui est sélectionné par défaut. Notez que ce type d'instance est éligible pour l'offre gratuite.
5. Sélectionnez Vérifier et lancer afin de laisser l'assistant compléter les autres paramètres de configuration pour vous.

6. Sur la page Review Instance Launch, sélectionnez Launch.
7. Lorsque vous êtes invité à saisir une key pair, sélectionnez.Créer une nouvelle key pair, entrez un nom pour la nouvelle key pair, puis choisissezTélécharger Pair de clés. C'est votre seule occasion d'enregistrer votre fichier clé privé. Veillez donc à télécharger la série de clés. Enregistrez le fichier de clé privée en lieu sûr. Vous devez fournir le nom de votre paire de clés quand vous lancez une instance, ainsi que la clé privée correspondante chaque fois que vous vous connectez à l'instance.

   Warning

   Ne sélectionnez pas l'option Continuer sans paire de clés. Si vous lancez votre instance sans une paire de clés, vous ne pourrez pas vous y connecter.

   Une fois que vous êtes prêt, choisissezLancer les instances.
8. Une page de confirmation indique que l'instance est en cours de lancement. Sélectionnez View Instances pour fermer la page de confirmation et revenir à la console.
9. Sur l'écran Instances, vous pouvez afficher le statut du lancement. Il suffit de peu de temps pour lancer une instance. Lorsque vous lancez une instance, son état initial est `pending`. Une fois que l'instance a démarré, son état devient `running` et elle reçoit un nom DNS public. (Si la colonne DNS public (IPv4) est masquée, sélectionnez l'icône Afficher / Masquer les colonnes (icône en forme d'engrenage) dans le coin supérieur droit de la page, puis sélectionnez DNS public (IPv4).)
10. Cela peut prendre quelques minutes avant que l'instance soit prête pour que vous puissiez vous y connecter. Vérifiez que votre instance a réussi ses contrôles de statut ; vous pouvez voir cette information dans la colonne Status Checks.

    Une fois que votre nouvelle instance a passé ses vérifications d'état, passez à la procédure suivante et connectez-vous à celle-ci.

Pour vous connecter à votre instance

Vous pouvez vous connecter à une instance à l'aide du client basé sur un navigateur en sélectionnant l'instance à partir de la console Amazon EC2 et en choisissant de vous connecter avec Amazon EC2 Instance Connect. Instance Connect gère les autorisations et fournit une connexion réussie.

1. Ouvrez la console Amazon EC2 à l'adresse https://console.aws.amazon.com/ec2/.
2. Dans le menu de gauche, choisissezInstances.
3. Sélectionnez l'instance, puis choisissez Connect (Connexion).
4. ChoisissezAmazon EC2 Instance Connect (connexion basée sur un navigateur),Connexion.

Vous devriez désormais disposer d'un objetAmazon EC2 Instance Connectqui est connectée à votre nouvelle instance Amazon EC2.

Installez Git, Node.js et configurez leAWS CLI

Dans cette section, vous allez installer Git et Node.js, sur votre instance Linux.

Pour installer Git

1. Dans vos dépensesAmazon EC2 Instance Connect, mettez à jour votre instance à l'aide de la commande suivante.

   ```
   sudo yum update -y
   ```
2. Dans vos dépensesAmazon EC2 Instance ConnectInstallez Git à l'aide de la commande suivante.

   ```
   sudo yum install git -y
   ```

Pour installer Node.js

1. Dans vos dépensesAmazon EC2 Instance ConnectInstallez le gestionnaire de version de nœud (nvm) à l'aide de la commande suivante.

   ```
   curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh | bash
   ```

   Nous allons utiliser nvm pour installer Node.js, car il peut installer plusieurs versions de Node.js et vous permettre de passer de l'une à l'autre.
2. Dans vos dépensesAmazon EC2 Instance Connect, activez nvm à l'aide de cette commande.

   ```
   . ~/.nvm/nvm.sh
   ```
3. Dans vos dépensesAmazon EC2 Instance Connect, utilisez nvm pour installer la dernière version de Node.js à l'aide de cette commande.

   ```
   nvm install node
   ```

   L'installation de Node.js installe également le gestionnaire de package de nœud (npm), ce qui vous permet d'installer des modules supplémentaires si besoin.
4. Dans vos dépensesAmazon EC2 Instance ConnectTestez que Node.js est installé et fonctionne correctement à l'aide de cette commande.

   ```
   node -v
   ```

   Ce didacticiel nécessite Node v10.0 ou une version ultérieure.

Pour configurer AWS CLI

Votre instance Amazon EC2 est préchargée avec le répertoireAWS CLI. Cependant, vous devez remplir votreAWS CLI. Pour plus d'informations sur la configuration de votre interface de ligne de commande, consultezConfiguration de l'AWS CLI.

1. L'exemple suivant montre des exemples de valeurs. Remplacez les par vos propres valeurs. Vous trouverez ces valeurs dans votre répertoireAWSdans les informations de votre compte sousMes informations d'identification de sécurité.

   Dans vos dépensesAmazon EC2 Instance ConnectDans la fenêtre, entrez cette commande :

   ```
   aws configure
   ```

   Entrez ensuite les valeurs de votre compte à l'invite affichée.

   ```
   AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
   AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   Default region name [None]: us-west-2
   Default output format [None]: json
   ```
2. Vous pouvez tester votreAWS CLIavec cette commande :

   ```
   aws iot describe-endpoint --endpoint-type iot:Data-ATS
   ```

   Si vos recettesAWS CLIest configuré correctement, la commande doit renvoyer une adresse de point de terminaison à partir de votre Compte AWS .

CréerAWS IoTressources pour votre périphérique virtuel

Cette section décrit comment utiliser l'objetAWS CLIpour créer l'objet chose et ses fichiers de certificat directement sur le périphérique virtuel. Ceci est fait directement sur le périphérique afin d'éviter les complications potentielles qui pourraient résulter de leur copie sur le périphérique à partir d'un autre ordinateur.

Pour créer unAWS IoTobjet de chose dans votre instance Linux

Périphériques connectés àAWS IoTsont représentées parobjets d'objetsdans leAWS IoTRegistre. Aobjet d'un objetreprésente un appareil spécifique ou une entité logique. Dans ce cas, votreobjet d'un objetreprésentera votre appareil virtuel, cette instance Amazon EC2.

1. Dans vos dépensesAmazon EC2 Instance Connect, exécutez la commande suivante pour créer votre objet objet objet objet.

```
aws iot create-thing --thing-name "MyIotThing"
```

2. La réponse JSON doit se présenter comme suit :

```
{
    "thingArn": "arn:aws:iot:your-region:your-aws-account:thing/MyIotThing",
    "thingName": "MyIotThing",
    "thingId": "6cf922a8-d8ea-4136-f3401EXAMPLE"
}
```

Pour créer et joindreAWS IoTclés et certificats dans votre instance Linux

La .create-keys-and-certificatecrée des certificats clients signés par l'autorité de certification racine Amazon. Ce certificat est utilisé pour authentifier l'identité de votre appareil virtuel.

1. Dans vos dépensesAmazon EC2 Instance Connect, créez un répertoire pour stocker vos fichiers de certificat et de clé.

```
mkdir ~/certs
```

2. Dans vos dépensesAmazon EC2 Instance Connect, téléchargez une copie du certificat d'autorité de certification Amazon à l'aide de cette commande.

```
curl -o ~/certs/Amazon-root-CA-1.pem \
    https://www.amazontrust.com/repository/AmazonRootCA1.pem
```

3. Dans vos dépensesAmazon EC2 Instance Connect, exécutez la commande suivante pour créer vos fichiers de clé privée, de clé publique et de certificat X.509. Cette commande enregistre et active également le certificat avecAWS IoT.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile "~/certs/device.pem.crt" \
    --public-key-outfile "~/certs/public.pem.key" \
    --private-key-outfile "~/certs/private.pem.key"
```

   La réponse se présente comme suit. Enregistrez`certificateArn`afin que vous puissiez l'utiliser dans les commandes suivantes. Vous en aurez besoin pour joindre votre certificat à votre objet et pour attacher la stratégie au certificat dans une étape ultérieure.

```
{
```

```
    "certificateArn": "arn:aws:iot:us-
west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificateId":
 "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fbd93adcb51ae54b83dc2",
    "certificatePem": "
-----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uXOjANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMC
VVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6
b24xFDASBgNVBAsTC0lBTSEXAMPLE2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAd
BgkqhkiG9w0BCQEWEG5vb25lQGFtYEXAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhcN
MTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYD
VQQHEwdTZWF0dGxlMQ8wDQYDVQQKEwZBbWF6b24xFDAEXAMPLEsTC0lBTSBDb25z
b2xlMRIwEAYDVQQDEwlUZXN0Q2lsYWMxHzAdBgkqhkiG9w0BCQEXAMPLE25lQGFt
YXpvbi5jb20wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAMaK0dn+aEXAMPLE
EXAMPLEfEvySWtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEIO3IyNoH/f0wYK8m9T
rDHudUZEXAMPLELG5M43q7Wgc/MbQITxOUSQv7c7ugFFDzQGBzZswY6786m86gpE
Ibb3OhjZnzcvQAEXAMPLEWIMm2nrAgMBAAEwDQYJKoZIhvcNAQEFBQADgYEAtCu4
nUhVVxYUntneD9+h8Mg9qEXAMPLEyExzyLwaxlAoo7TJHidbtS4J5iNmZgXL0Fkb
FFBjvSfpJIlJ00zbhNYS5f6GuoEDEXAMPLEBHjJnyp378OD8uTs7fLvjx79LjSTb
NYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaEXAMPLE=
-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC
 KEY-----\nMIIBIjANBgkqhkEXAMPLEQEFAAOCAQ8AMIIBCgKCAQEAEXAMPLE1nnyJwKSMHw4h
\nMMEXAMPLEuuN/dMAS3fyce8DW/4+EXAMPLEyjmoF/YVF/
gHr99VEEXAMPLE5VF13\n59VK7cEXAMPLE67GK+y+jikqXOgHh/xJTwo
+sGpWEXAMPLEDz18xOd2ka4tCzuWEXAMPLEahJbYkCPUBSU8opVkR7qkEXAMPLE1DR6sx2HocliOOLtu6Fkw91swQWEXAMPLE
\GB3ZPrNh0PzQYvjUStZeccyNCx2EXAMPLEvp9mQOUXP6plfgxwKRX2fEXAMPLEDa\nhJLXkX3rHU2xbxJSq7D
+XEXAMPLEcw+LyFhI5mgFRl88eGdsAEXAMPLElnI9EesG\nFQIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security
 reasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

4. Dans vos dépensesAmazon EC2 Instance Connect, attachez votre objet truc au certificat que vous venez de créer à l'aide de la commande suivante et de la commande*certificateArn*dans la réponse de la commande précédente.

```
aws iot attach-thing-principal \
    --thing-name "MyIotThing" \
    --principal "certificateArn"
```

   Si elle aboutit, cette commande n'affiche aucune sortie.

Pour créer et attacher une stratégie

1. Dans vos dépensesAmazon EC2 Instance Connect, créez le fichier de stratégie en copiant et en collant ce document de stratégie dans un fichier nommé**`~/policy.json`**.

   Si vous n'avez pas d'éditeur Linux préféré, vous pouvez ouvrirnano, en utilisant cette commande.

```
nano ~/policy.json
```

   Et coller le document de stratégie pour`policy.json`en elle. Entrez ctrl-x pour quitter le champnanoÉditeur et enregistrez le fichier.

   Contenu du document de politique`policy.json`.

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Subscribe",
                "iot:Receive",
                "iot:Connect"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

2. Dans vos dépensesAmazon EC2 Instance Connect, créez votre stratégie à l'aide de la commande suivante.

```
aws iot create-policy \
    --policy-name "MyIotThingPolicy" \
    --policy-document "file://~/policy.json"
```

File d'attente:

```
{
    "policyName": "MyIotThingPolicy",
    "policyArn": "arn:aws:iot:your-region:your-aws-account:policy/MyIotThingPolicy",
    "policyDocument": "{
        \"Version\": \"2012-10-17\",
        \"Statement\": [
            {
                \"Effect\": \"Allow\",
                \"Action\": [
                    \"iot:Publish\",
                    \"iot:Receive\",
                    \"iot:Subscribe\",
                    \"iot:Connect\"
                ],
                \"Resource\": [
                    \"*\"
                ]
            }
        ]
    }",
    "policyVersionId": "1"
}
```

3. Dans vos dépensesAmazon EC2 Instance Connect, attachez la stratégie au certificat de votre appareil virtuel à l'aide de la commande suivante.

```
aws iot attach-policy \
    --policy-name "MyIotThingPolicy" \
    --target "certificateArn"
```

Si elle aboutit, cette commande n'affiche aucune sortie.

À ce stade, vous avez créé pour votre périphérique virtuel :

- Objet de chose pour représenter votre périphérique virtuel dansAWS IoT.

- Certificat permettant d'authentifier votre périphérique virtuel.
- Document de stratégie permettant d'autoriser votre appareil virtuel à Connect àAWS IoTet pour publier, recevoir et s'abonner aux messages.

Installer le kit SDK des appareils AWS IoT pour Javascript

Dans cette section, vous allez installer l'installationAWS IoTSDK de périphérique pour JavaScript, qui contient le code que les applications peuvent utiliser pour communiquer avecAWS IoTet les exemples de programmes.

Pour installerAWS IoTKit SDK des appareils pour JavaScript sur votre instance Linux

1. Dans vos dépensesAmazon EC2 Instance Connect, cloner leAWS IoTKit SDK des appareils pour JavaScript dans le répertoire`aws-iot-device-sdk-js-v2`de votre répertoire personnel à l'aide de cette commande.

```
cd ~
git clone https://github.com/aws/aws-iot-device-sdk-js-v2.git
```

2. Accédez à `aws-iot-device-sdk-js-v2`que vous avez créé à l'étape précédente.

```
cd aws-iot-device-sdk-js-v2
```

3. Utilisez npm pour installer le SDK.

```
npm install
```

Exécuter l'exemple d'application

Les commandes figurant dans les sections suivantes supposent que vos fichiers de clé et de certificat sont stockés sur votre appareil virtuel, comme indiqué dans ce tableau.

Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Clé privée	`~/certs/private.pem.key`
Certificat de l'appareil	`~/certs/device.pem.crt`
Certificat racine de l'autorité de certification	`~/certs/Amazon-root-CA-1.pem`

Dans cette section, vous allez installer et exécuter le répertoire`pub-sub.js`exemple d'application trouvée dans le`aws-iot-device-sdk-js-v2/samples/node`duAWS IoTKit SDK des périphériques pour JavaScript. Cette application montre comment un appareil, votre instance Amazon EC2, utilise la bibliothèque MQTT pour publier et s'abonner à des messages MQTT. La `pub-sub.js`exemple d'application s'abonne à un sujet,`topic_1`, publie 10 messages dans cette rubrique et affiche les messages tels qu'ils sont reçus du courtier de messages.

Pour installer et exécuter l'exemple d'application

1. Dans vos dépensesAmazon EC2 Instance Connect, accédez à la fenêtre`aws-iot-device-sdk-js-v2/samples/node/pub_sub`que le SDK a créé et installez l'exemple d'application à l'aide de ces commandes.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
npm install
```

2. Dans vos dépensesAmazon EC2 Instance Connectfenêtre, obtenez*votre-iot-point de terminaison*fromAWS IoTen utilisant cette commande.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

3. Dans vos dépensesAmazon EC2 Instance ConnectFenêtre, insérer*votre-iot-point de terminaison*comme indiqué et exécutez cette commande.

```
node dist/index.js --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Exemple d'application :

1. Se connecte àAWS IoTpour votre compte.
2. S'abonne à la rubrique du message,topic_1et affiche les messages qu'il reçoit sur cette rubrique.
3. Publit 10 messages sur le sujet,topic_1.
4. Affiche un résultat semblable à ce qui suit :

```
Publish received on topic topic_1
{"message":"Hello world!","sequence":1}
Publish received on topic topic_1
{"message":"Hello world!","sequence":2}
Publish received on topic topic_1
{"message":"Hello world!","sequence":3}
Publish received on topic topic_1
{"message":"Hello world!","sequence":4}
Publish received on topic topic_1
{"message":"Hello world!","sequence":5}
Publish received on topic topic_1
{"message":"Hello world!","sequence":6}
Publish received on topic topic_1
{"message":"Hello world!","sequence":7}
Publish received on topic topic_1
{"message":"Hello world!","sequence":8}
Publish received on topic topic_1
{"message":"Hello world!","sequence":9}
Publish received on topic topic_1
{"message":"Hello world!","sequence":10}
```

Si vous rencontrez des problèmes dans l'exécution de l'exemple d'application, consultezthe section called “Dépannage des problèmes liés à l'exemple d'application” (p. 61).

Vous pouvez également ajouter le`--verbosity debug`à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'elle fait. Ces informations peuvent vous fournir l'aide dont vous avez besoin pour corriger le problème.

Affichez les messages de l'exemple d'application dans leAWS IoTconsole

Vous pouvez voir les messages de l'application exemple lorsqu'ils passent par le courtier de messages à l'aide de l'outilClient MQTTdans leAWS IoTconsole.

Pour afficher les messages MQTT publiés par l'exemple d'application

1. Vérifiez Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62). Cela vous apprendra à utiliser l'outilClient MQTTdans leAWS IoTconsolePour afficher les messages MQTT au fur et à mesure qu'ils transitent par le courtier de messages.
2. Ouverture d'Client MQTTdans leAWS IoTconsole.
3. Abonnez-vous à la rubrique,topic_1.
4. Dans vos dépensesAmazon EC2 Instance Connect, exécutez à nouveau l'exemple d'application et regardez les messages dans la fenêtreClient MQTTdans leAWS IoTconsole.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
node dist/index.js --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Utilisez votre PC Windows ou Linux ou Mac en tant queAWS IoTAppareil

Dans ce didacticiel, vous allez configurer un ordinateur personnel pour utiliser avecAWS IoT. Ces instructions prennent en charge les PC et Mac Windows et Linux. Pour ce faire, vous devez installer un logiciel sur votre ordinateur. Si vous ne souhaitez pas installer de logiciel sur votre ordinateur, vous pouvez essayerCréez un appareil virtuel avec Amazon EC2 (p. 39), qui installe tous les logiciels sur une machine virtuelle.

Dans ce didacticiel, vous allez :

- Configurer votre ordinateur personnel (p. 47)
- Installez Git, Python et leAWS IoTKit SDK des appareils pour Python (p. 47)
- Configurer la stratégie et exécuter l'exemple d'application (p. 50)
- Affichez les messages de l'exemple d'application dans leAWS IoTconsole (p. 52)

Configurer votre ordinateur personnel

Pour compléter ce didacticiel, vous avez besoin d'un PC Windows ou Linux ou d'un Mac avec une connexion à Internet.

Avant de passer à l'étape suivante, assurez-vous que vous pouvez ouvrir une fenêtre de ligne de commande sur votre ordinateur. Utilisercmd.exesur un PC Windows. Sur un PC Linux ou un Mac, utilisezTerminal.

Installez Git, Python et leAWS IoTKit SDK des appareils pour Python

Dans cette section, vous allez installer Python, et le répertoireAWS IoTKit SDK des appareils pour Python sur votre ordinateur.

Installez la dernière version de Git et Python

Pour télécharger et installer Git et Python sur votre ordinateur

1. Vérifiez si Git est installé sur votre ordinateur. Entrez cette commande dans la ligne de commande.

```
git --version
```

   Si la commande affiche la version de Git, Git est installé et vous pouvez passer à l'étape suivante.

Si la commande affiche une erreur, ouvrezhttps://git-scm.com/downloadet installez Git pour votre ordinateur.

2. Vérifiez si vous avez déjà installé Python. Entrez cette commande dans la ligne de commande.

```
python -V
```

> **Note**
>
> Si cette commande donne une erreur :`Python was not found`, cela peut être dû au fait que votre système d'exploitation appelle l'exécutable Python v3.x en tant que`Python3`. Dans ce cas, remplacez toutes les instances de`python`avec`python3`et continuez le reste de ce didacticiel.

Si la commande affiche la version Python, Python est déjà installé. Ce didacticiel nécessite Python v3.5 ou une version ultérieure.

3. Si Python est installé, vous pouvez ignorer le reste des étapes de cette section. Sinon, poursuivez.
4. Ouvrirhttps://www.python.org/downloads/et téléchargez le programme d'installation de votre ordinateur.
5. Si le téléchargement n'a pas démarré automatiquement, exécutez le programme téléchargé pour installer Python.
6. Vérifiez l'installation de Python.

```
python -V
```

Confirmez que la commande affiche la version Python. Si la version Python n'est pas affichée, essayez de télécharger et d'installer à nouveau Python.

Installer leAWS IoTKit SDK des appareils pour Python

Pour installerAWS IoTKit SDK des appareils pour Python sur votre ordinateur

1. Installez la version 2 duAWS IoTKit SDK des périphériques pour Python.

```
python3 -m pip install awsiotsdk
```

2. CloneAWS IoTDevice SDK pour le dépôt Python dans le répertoire aws-iot-device-sdk-python-v2 de votre répertoire personnel. Cette procédure fait référence au répertoire de base des fichiers que vous installez en tant que*Accueil*.

   L'emplacement réel de la propriété*Accueil*Le répertoire dépend de votre système d'exploitation.

   Linux/macOS

   Dans macOS et Linux, le répertoire*Accueil*Le répertoire est~.

   ```
   cd ~
   git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
   ```

   Windows

   Dans Windows, vous pouvez trouver le répertoire*Accueil*en exécutant cette commande dans le répertoirecmd.

   ```
   echo %USERPROFILE%
   ```

```
cd %USERPROFILE%
git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
```

Note

Si vous utilisez Windows PowerShell par opposition àcmd.exe, puis utilisez la commande suivante.

```
echo $home
```

Se préparer à exécuter les exemples d'applications

Pour préparer votre système à exécuter l'exemple d'application

- Création de l'`certs`. Dans le`certs`, copiez-y la clé privée, le certificat d'appareil et les fichiers de certificat d'autorité de certification racine que vous avez enregistrés lors de la création et l'enregistrement de l'objet objet objet objet objet objet dansthe section called “CréerAWS IoTresources” (p. 35). Les noms de fichiers de chaque fichier du répertoire de destination doivent correspondre à ceux de la table.

  Les commandes figurant dans la section suivante supposent que vos fichiers de clé et de certificat sont stockés sur votre appareil, comme indiqué dans ce tableau.

  Linux/macOS

  Exécutez cette commande pour créer l'objet`certs`que vous allez utiliser lors de l'exécution des exemples d'applications.

  ```
  mkdir ~/certs
  ```

  Dans le nouveau sous-répertoire, copiez les fichiers vers les chemins de fichier de destination indiqués dans le tableau suivant.

  Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Clé privée	`~/certs/private.pem.key`
Certificat de l'appareil	`~/certs/device.pem.crt`
Certificat racine de l'autorité de certification	`~/certs/Amazon-root-CA-1.pem`

  Exécutez cette commande pour répertorier les fichiers dans la boîte de dialogue`certs`et les comparer à ceux répertoriés dans la table.

  ```
  ls -l ~/certs
  ```

  Windows

  Exécutez cette commande pour créer l'objet`certs`que vous allez utiliser lors de l'exécution des exemples d'applications.

```
mkdir %USERPROFILE%\certs
```

Dans le nouveau sous-répertoire, copiez les fichiers vers les chemins de fichier de destination indiqués dans le tableau suivant.

Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Clé privée	`%USERPROFILE%\certs\private.pem.key`
Certificat de l'appareil	`%USERPROFILE%\certs\device.pem.crt`
Certificat racine de l'autorité de certification	`%USERPROFILE%\certs\Amazon-root-CA-1.pem`

Exécutez cette commande pour répertorier les fichiers dans la boîte de dialogue`certs`et les comparer à ceux répertoriés dans la table.

```
dir %USERPROFILE%\certs
```

Configurer la stratégie et exécuter l'exemple d'application

Dans cette section, vous allez configurer votre stratégie et exécuter l'`pubsub.py`trouvé dans l'exemple de script`aws-iot-device-sdk-python-v2/samples`du AWS IoT Device SDK for Python . Ce script indique la façon dont votre appareil utilise la bibliothèque MQTT pour publier et s'abonner à des messages MQTT.

La .`pubsub.py`exemple d'application s'abonne à un sujet,`test/topic`, publie 10 messages dans cette rubrique et affiche les messages tels qu'ils sont reçus du courtier de messages.

Pour exécuter le`pubsub.py`Exemple de script, vous avez besoin des informations suivantes :

Valeurs des paramètres d'application

Paramètre	Où chercher la valeur
votre-iot-point de terminaison	1. DansAWS IoTconsoleDans le menu de gauche, choisissezParamètres. 2. Dans la pageParamètres, votre point de terminaison est affiché dans le répertoirePoint de terminaison des données deSection.

La .*votre-iot-point de terminaison*a un format de :*endpoint_id*-`ats.iot.`*region*`.amazonaws.com`, par exemple,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`.

Avant d'exécuter le script, assurez-vous que la stratégie de votre objet fournit les autorisations permettant à l'exemple de script de se connecter, de s'abonner, de publier et de recevoir. Le JSON de stratégie s'affiche dans l'exemple suivant. Dans`"Resource"`, vous devez remplacer la région et le mot de compte par votre Compte AWS Région et`"arn:aws:iot:`*region*`:`*account*`.:topic/test/topic"`

```
{
"Version": "2012-10-17",
"Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "iot:Publish",
            "iot:Receive"
        ],
        "Resource": [
            "arn:aws:iot:region:account:topic/test/topic"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Subscribe"
        ],
        "Resource": [
            "arn:aws:iot:region:account:topicfilter/test/topic"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Connect"
        ],
        "Resource": [
            "arn:aws:iot:region:account:client/test-*"
        ]
    }
]
}
```

Linux/macOS

Pour exécuter l'exemple de script sous Linux/macOS

1. Dans votre fenêtre de ligne de commande, naviguez jusqu'au répertoire.`~/aws-iot-device-sdk-python-v2/samples/node/pub_sub`que le SDK a créé à l'aide de ces commandes.

   ```
   cd ~/aws-iot-device-sdk-python-v2/samples
   ```

2. Dans votre fenêtre de ligne de commande, remplacez la fenêtre*votre-iot-point de terminaison*comme indiqué et exécutez cette commande.

   ```
   python3 pubsub.py --endpoint your-iot-endpoint --root-ca ~/certs/Amazon-root-
   CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key
   ```

Windows

Pour exécuter l'exemple d'application sur un PC Windows

1. Dans votre fenêtre de ligne de commande, naviguez jusqu'au répertoire.`%USERPROFILE%\aws-iot-device-sdk-python-v2\samples`que le SDK a créé et installez l'exemple d'application à l'aide de ces commandes.

   ```
   cd %USERPROFILE%\aws-iot-device-sdk-python-v2\samples
   ```

2. Dans votre fenêtre de ligne de commande, remplacez la fenêtre*votre-iot-point de terminaison*comme indiqué et exécutez cette commande.

```
python3 pubsub.py --endpoint your-iot-endpoint --root-ca %USERPROFILE%\certs
\Amazon-root-CA-1.pem --cert %USERPROFILE%\certs\device.pem.crt --key %USERPROFILE%
\certs\private.pem.key
```

L'exemple de script :

1. Se connecte àAWS IoTpour votre compte.
2. S'abonne à la rubrique du message,test/sujetet affiche les messages qu'il reçoit sur cette rubrique.
3. Publit 10 messages sur le sujet,test/sujet.
4. Affiche un résultat semblable à ce qui suit :

```
Publish received on topic test/topic
{"message":"Hello world!","sequence":1}
Publish received on topic test/topic
{"message":"Hello world!","sequence":2}
Publish received on topic test/topic
{"message":"Hello world!","sequence":3}
Publish received on topic test/topic
{"message":"Hello world!","sequence":4}
Publish received on topic test/topic
{"message":"Hello world!","sequence":5}
Publish received on topic test/topic
{"message":"Hello world!","sequence":6}
Publish received on topic test/topic
{"message":"Hello world!","sequence":7}
Publish received on topic test/topic
{"message":"Hello world!","sequence":8}
Publish received on topic test/topic
{"message":"Hello world!","sequence":9}
Publish received on topic test/topic
{"message":"Hello world!","sequence":10}
```

Si vous rencontrez des problèmes dans l'exécution de l'exemple d'application, consultezthe section called “Dépannage des problèmes liés à l'exemple d'application” (p. 61).

Vous pouvez également ajouter le`--verbosity debug`à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'elle fait. Ces informations peuvent vous aider à corriger le problème.

Affichez les messages de l'exemple d'application dans leAWS IoTconsole

Vous pouvez voir les messages de l'application exemple lorsqu'ils passent par le courtier de messages à l'aide de l'outilClient MQTTdans leAWS IoTconsole.

Pour afficher les messages MQTT publiés par l'exemple d'application

1. Vérifiez Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62). Cela vous apprendra à utiliser l'outilClient MQTTdans leAWS IoTconsolePour afficher les messages MQTT au fur et à mesure qu'ils transitent par le courtier de messages.
2. Ouverture d'Client MQTTdans leAWS IoTconsole.
3. Abonnez-vous à la rubrique,test/sujet.
4. Dans la fenêtre de votre ligne de commande, exécutez à nouveau l'exemple d'application et regardez les messages dans la fenêtreClient MQTTdans leAWS IoTconsole.

Linux/macOS

```
cd ~/aws-iot-device-sdk-python-v2/samples
python3 pubsub.py --topic test/topic --root-ca ~/certs/Amazon-root-CA-1.pem --cert
 ~/certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Windows

```
cd %USERPROFILE%\aws-iot-device-sdk-python-v2\samples
python3 pubsub.py --topic test/topic --root-ca %USERPROFILE%\certs\Amazon-root-
CA-1.pem --cert %USERPROFILE%\certs\device.pem.crt --key %USERPROFILE%\certs
\private.pem.key --endpoint your-iot-endpoint
```

Connect un Raspberry Pi ou un autre appareil

Dans cette section, nous allons configurer un Raspberry Pi pour une utilisation avecAWS IoT. Si vous avez un autre appareil que vous souhaitez connecter, les instructions du Raspberry Pi incluent des références qui peuvent vous aider à adapter ces instructions à votre appareil.

Cela prend normalement environ 20 minutes, mais cela peut prendre plus de temps si vous avez de nombreuses mises à niveau logicielles système à installer.

Dans ce didacticiel, vous allez :

- Configuration de votre appareil (p. 53)
- Installez les outils et bibliothèques requis pour leAWS IoTKits SDK pour les appareils (p. 54)
- InstallAWS IoTKits SDK pour les appareils (p. 55)
- Installez et exécutez l'exemple d'application (p. 57)
- Affichez les messages de l'exemple d'application dans leAWS IoTconsole (p. 60)

> **Important**
>
> L'adaptation de ces instructions à d'autres périphériques et systèmes d'exploitation peut s'avérer difficile. Vous devez bien comprendre votre appareil pour pouvoir interpréter ces instructions et les appliquer à votre appareil.
> Si vous rencontrez des difficultés lors de la configuration de votre appareil pourAWS IoT, nous ne pouvons pas offrir d'aide au-delà des instructions de cette section. Cependant, vous pouvez essayer l'une des autres options de périphérique comme alternative, par exempleCréez un appareil virtuel avec Amazon EC2 (p. 39)ouUtilisez votre PC Windows ou Linux ou Mac en tant queAWS IoTAppareil (p. 47).

Configuration de votre appareil

L'objectif de cette étape est de collecter ce dont vous aurez besoin pour configurer votre appareil afin qu'il puisse démarrer le système d'exploitation (OS), se connecter à Internet et vous permettre d'interagir avec celui-ci à l'aide d'une interface de ligne de commande.

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

- Un Compte AWS . Si vous n'en avez pas, suivez les étapes décrites dansConfigurer votre Compte AWS (p. 19)Avant de continuer.
- ARaspberry Pi 3 Modèle Bou un modèle plus récent. Cela peut fonctionner sur les versions antérieures du Raspberry Pi, mais elles n'ont pas été testées.

- Raspberry Pi OS (32 bits)ou version ultérieure. Nous vous recommandons d'utiliser la dernière version du système d'exploitation Raspberry Pi. Les versions antérieures du système d'exploitation peuvent fonctionner, mais elles n'ont pas été testées.

  Pour exécuter cet exemple, vous n'avez pas besoin d'installer le bureau avec l'interface utilisateur graphique (GUI) ; toutefois, si vous êtes nouveau dans Raspberry Pi et que votre matériel Raspberry Pi le prend en charge, il peut être plus facile d'utiliser le bureau avec l'interface graphique.
- Une connexion Ethernet ou Wi-Fi.
- Clavier, souris, moniteur, câbles, blocs d'alimentation et tout autre matériel requis par votre appareil.

  Important

  Avant de passer à l'étape suivante, votre appareil doit avoir installé, configuré et fonctionne. L'appareil doit être connecté à Internet et vous devez pouvoir accéder à l'appareil en utilisant son interface de ligne de commande. L'accès en ligne de commande peut se faire par le biais d'un clavier, d'une souris et d'un moniteur directement connectés, ou par l'intermédiaire d'une interface distante du terminal SSH.

Si vous exécutez un système d'exploitation sur votre Raspberry Pi doté d'une interface utilisateur graphique (GUI), ouvrez une fenêtre de terminal sur le périphérique et exécutez les instructions suivantes dans cette fenêtre. Sinon, si vous vous connectez à votre appareil à l'aide d'un terminal distant, tel que PuTTY, ouvrez un terminal distant sur votre appareil et utilisez-le.

Installez les outils et bibliothèques requis pour leAWS IoTKits SDK pour les appareils

Avant d'installer leAWS IoTDevice SDK et exemple de code, assurez-vous que votre système est à jour et dispose des outils et bibliothèques nécessaires pour installer les SDK.

1. Mise à jour du système d'exploitation et installation des bibliothèques nécessaires

   Avant d'installer unAWS IoTDevice SDK, exécutez ces commandes dans une fenêtre de terminal sur votre appareil pour mettre à jour le système d'exploitation et installer les bibliothèques requises.

   ```
   sudo apt-get update
   ```

   ```
   sudo apt-get upgrade
   ```

   ```
   sudo apt-get install cmake
   ```

   ```
   sudo apt-get install libssl-dev
   ```
2. Installez Git

   Si le système d'exploitation de votre appareil n'est pas fourni avec Git installé, vous devrez l'installer pour installer leAWS IoTKit SDK des périphériques pour JavaScript.

   a. Testez pour vérifier si Git est déjà installé en exécutant cette commande.

      ```
      git --version
      ```

   b. Si la commande précédente renvoie la version de Git, Git est déjà installé et vous pouvez passer à l'étape 3.

   c. Si une erreur s'affiche lorsque vous exécutez la commandegit, installez Git en exécutant cette commande.

```
sudo apt-get install git
```

d. Testez à nouveau pour voir si Git est installé en exécutant cette commande.

```
git --version
```

e. Si Git est installé, passez à la section suivante. Si ce n'est pas le cas, résolvez et corrigez l'erreur avant de continuer. Vous avez besoin de Git pour installer leAWS IoTKit SDK des périphériques pour JavaScript.

InstallAWS IoTKits SDK pour les appareils

Installer leAWS IoTKit SDK des appareils.

Python

Dans cette section, vous allez installer Python, ses outils de développement et leAWS IoTKit SDK des appareils pour Python sur votre appareil. Ces instructions s'appliquent à un Raspberry Pi exécutant le dernier système d'exploitation Raspberry Pi. Si vous utilisez un autre appareil ou si vous utilisez un autre système d'exploitation, vous devrez peut-être adapter ces instructions pour votre appareil.

1. Installez Python et ses outils de développement

   La .AWS IoTLe SDK de périphérique pour Python nécessite l'installation de Python 3.5 ou ultérieure sur votre Raspberry Pi.

   Exécutez ces commandes dans une fenêtre de terminal sur votre appareil.

   1. Exécutez cette commande pour déterminer la version de Python installée sur votre appareil.

      ```
      python3 --version
      ```

      Si Python est installé, il affichera sa version.
   2. Si la version affichée est`Python 3.5`ou supérieur, vous pouvez passer à l'étape 2.
   3. Si la version affichée est inférieure à`Python 3.5`, vous pouvez installer la version correcte en exécutant cette commande.

      ```
      sudo apt install python3
      ```
   4. Exécutez cette commande pour confirmer que la version correcte de Python est maintenant installée.

      ```
      python3 --version
      ```
2. Essai pour pip3

   Exécutez ces commandes dans une fenêtre de terminal sur votre appareil.

   1. Exécutez cette commande pour voir sipip3est installé.

      ```
      pip3 --version
      ```
   2. Si la commande renvoie un numéro de version,pip3est installé et vous pouvez passer à l'étape 3.
   3. Si la commande précédente renvoie une erreur, exécutez cette commande pour installerpip3.

```
sudo apt install python3-pip
```

4. Exécutez cette commande pour voir sipip3est installé.

```
pip3 --version
```

3. Installer les actuelsAWS IoTKit SDK des appareils pour Python

   Installer leAWS IoTDevice SDK pour Python et téléchargez les exemples d'applications sur votre appareil.

   Sur votre appareil, exécutez ces commandes.

```
cd ~
python3 -m pip install awsiotsdk
```

```
git clone https://github.com/aws/aws-iot-device-sdk-python-v2.git
```

JavaScript

Dans cette section, vous allez installer Node.js, le gestionnaire de paquets npm et leAWS IoTKit SDK des appareils pour JavaScript sur votre appareil. Ces instructions s'appliquent à un Raspberry Pi exécutant le système d'exploitation Raspberry Pi. Si vous utilisez un autre appareil ou si vous utilisez un autre système d'exploitation, vous devrez peut-être adapter ces instructions pour votre appareil.

1. Installer la dernière version de Node.js

   La .AWS IoTLe kit SDK pour les appareils pour JavaScript nécessite l'installation de Node.js et du gestionnaire de package npm sur votre appareil Raspberry Pi.

   a. Téléchargez la version Node la plus récente en saisissant cette commande.

```
cd ~
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -
```

   b. Installez Node et npm.

```
sudo apt-get install -y nodejs
```

   c. Vérifiez l'installation de Node.

```
node -v
```

      Confirmez que la commande affiche la version du nœud. Ce didacticiel nécessite Node v10.0 ou une version ultérieure. Si la version Node n'est pas affichée, essayez de télécharger à nouveau le référentiel Node.

   d. Vérifier l'installation de npm.

```
npm -v
```

      Confirmez que la commande affiche la version npm. Si la version npm n'est pas affichée, réessayez d'installer Node et npm.

   e. Redémarrez le périphérique.

```
sudo shutdown -r 0
```

Continuez après le redémarrage de l'appareil.

2. Installer le kit SDK des appareils AWS IoT pour Javascript

   Installez le kit SDK des appareils AWS IoT pour JavaScript sur votre Raspberry Pi.

   a. Installez aws-crt, la bibliothèque d'exécution commune.

   ```
   cd ~
   npm install aws-crt
   ```

   b. Installez la version 2 duAWS IoTKit SDK des périphériques pour JavaScript.

   ```
   npm install aws-iot-device-sdk-v2
   ```

   c. CloneAWS IoTKit SDK des appareils pour JavaScript dans le répertoire`aws-iot-device-sdk-js-v2`de votre annuaire*`Accueil`*directory. Sur le Raspberry Pi, le*`Accueil`*Le répertoire est`~/`, qui est utilisée comme*`Accueil`*dans les commandes suivantes. Si votre appareil utilise un chemin différent pour le*`Accueil`*, vous devez remplacer`~/`avec le chemin correct pour votre appareil dans les commandes suivantes.

   Ces commandes créent l'objet`~/aws-iot-device-sdk-js-v2`et copiez le code SDK dedans.

   ```
   cd ~
   git clone https://github.com/aws/aws-iot-device-sdk-js-v2.git
   ```

   d. Remplacez par`aws-iot-device-sdk-js-v2`que vous avez créé à l'étape précédente et installez le Kit de développement logiciel (SDK).

   ```
   cd ~/aws-iot-device-sdk-js-v2
   npm install
   ```

Installez et exécutez l'exemple d'application

Dans cette section, vous allez installer et exécuter le répertoire`pubsub`exemple d'application trouvée dans leAWS IoTKit SDK des appareils. Cette application indique la façon dont votre appareil utilise la bibliothèque MQTT pour publier et vous abonner à des messages MQTT. L'exemple d'application s'abonne à un sujet,`topic_1`, publie 10 messages dans cette rubrique et affiche les messages tels qu'ils sont reçus du courtier de messages.

Installation des fichiers de certificat

L'exemple d'application nécessite les fichiers de certificat qui authentifient le périphérique à installer sur l'appareil.

Pour installer les fichiers de certificat de périphérique pour l'exemple d'application

1. Création d'un`certs`dans votre répertoire*`Accueil`*en exécutant ces commandes.

   ```
   cd ~
   mkdir certs
   ```

2. Dans le`~/certs`, copiez-y la clé privée, le certificat d'appareil et le certificat d'autorité de certification racine que vous avez créé précédemment dansthe section called “CréerAWS IoTresources” (p. 35).

La façon dont vous copiez les fichiers de certificat sur votre appareil dépend du périphérique et du système d'exploitation et n'est pas décrite ici. Toutefois, si votre appareil prend en charge une interface utilisateur graphique (GUI) et dispose d'un navigateur Web, vous pouvez effectuer la procédure décrite dansthe section called “CréerAWS IoTresources” (p. 35)depuis le navigateur Web de votre appareil pour télécharger les fichiers résultants directement sur votre appareil.

Les commandes figurant dans la section suivante supposent que vos fichiers de clé et de certificat sont stockés sur l'appareil, comme indiqué dans ce tableau.

Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Certificat racine de l'autorité de certification	`~/certs/Amazon-root-CA-1.pem`
Certificat de l'appareil	`~/certs/device.pem.crt`
Clé privée	`~/certs/private.pem.key`

Pour exécuter cet exemple d'application, vous avez besoin des informations suivantes :

Valeurs des paramètres d'application

Paramètre	Où chercher la valeur
votre-iot-point de terminaison	DansAWS IoTconsole, choisissezGérer, puisObjets. Choisissez l'objet IoT que vous avez créé pour votre appareil,MyIoThingLe nom utilisé précédemment, puis choisissezInteragir. Sur la page de détails de l'objet, votre point de terminaison s'affiche dans laHTTPSSection.

La .*votre-iot-point de terminaison*a un format de :*endpoint_id*-`ats.iot.`*region*`.amazonaws.com`, par exemple,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`.

Python

Pour installer et exécuter l'exemple d'application

1. Accédez au répertoire d'exemple d'application.

```
cd ~/aws-iot-device-sdk-python-v2/samples
```

2. Dans la fenêtre de ligne de commande, remplacez la fenêtre*votre-iot-point de terminaison*comme indiqué et exécutez cette commande.

```
python3 pubsub.py --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

3. Observez que l'exemple d'application :

   1. Se connecte àAWS IoTpour votre compte.

2. S'abonne à la rubrique du message,topic_1et affiche les messages qu'il reçoit sur cette rubrique.
3. Publit 10 messages sur le sujet,topic_1.
4. Affiche un résultat semblable à ce qui suit :

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to topic 'topic_1'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 10 message(s)
Publishing message to topic 'topic_1': Hello World! [1]
Received message from topic 'topic_1': b'Hello World! [1]'
Publishing message to topic 'topic_1': Hello World! [2]
Received message from topic 'topic_1': b'Hello World! [2]'
Publishing message to topic 'topic_1': Hello World! [3]
Received message from topic 'topic_1': b'Hello World! [3]'
Publishing message to topic 'topic_1': Hello World! [4]
Received message from topic 'topic_1': b'Hello World! [4]'
Publishing message to topic 'topic_1': Hello World! [5]
Received message from topic 'topic_1': b'Hello World! [5]'
Publishing message to topic 'topic_1': Hello World! [6]
Received message from topic 'topic_1': b'Hello World! [6]'
Publishing message to topic 'topic_1': Hello World! [7]
Received message from topic 'topic_1': b'Hello World! [7]'
Publishing message to topic 'topic_1': Hello World! [8]
Received message from topic 'topic_1': b'Hello World! [8]'
Publishing message to topic 'topic_1': Hello World! [9]
Received message from topic 'topic_1': b'Hello World! [9]'
Publishing message to topic 'topic_1': Hello World! [10]
Received message from topic 'topic_1': b'Hello World! [10]'
10 message(s) received.
Disconnecting...
Disconnected!
```

Si vous rencontrez des problèmes dans l'exécution de l'exemple d'application, consultezthe section called “Dépannage des problèmes liés à l'exemple d'application” (p. 61).

Vous pouvez également ajouter le`--verbosity debug`à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'elle fait. Ces informations peuvent vous fournir l'aide dont vous avez besoin pour corriger le problème.

JavaScript

Pour installer et exécuter l'exemple d'application

1. Dans votre fenêtre de ligne de commande, naviguez jusqu'au répertoire.`~/aws-iot-device-sdk-js-v2/samples/node/pub_sub`que le SDK a créé et installez l'exemple d'application à l'aide de ces commandes.

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
npm install
```

2. Dans la fenêtre de ligne de commande, remplacez la fenêtre*votre-iot-point de terminaison*comme indiqué et exécutez cette commande.

```
node dist/index.js --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

3. Observez que l'exemple d'application :

   1. Se connecte àAWS IoTpour votre compte.
   2. S'abonne à la rubrique du message,topic_1et affiche les messages qu'il reçoit sur cette rubrique.
   3. Publit 10 messages sur le sujet,topic_1.
   4. Affiche un résultat semblable à ce qui suit :

```
Publish received on topic topic_1
{"message":"Hello world!","sequence":1}
Publish received on topic topic_1
{"message":"Hello world!","sequence":2}
Publish received on topic topic_1
{"message":"Hello world!","sequence":3}
Publish received on topic topic_1
{"message":"Hello world!","sequence":4}
Publish received on topic topic_1
{"message":"Hello world!","sequence":5}
Publish received on topic topic_1
{"message":"Hello world!","sequence":6}
Publish received on topic topic_1
{"message":"Hello world!","sequence":7}
Publish received on topic topic_1
{"message":"Hello world!","sequence":8}
Publish received on topic topic_1
{"message":"Hello world!","sequence":9}
Publish received on topic topic_1
{"message":"Hello world!","sequence":10}
```

   Si vous rencontrez des problèmes dans l'exécution de l'exemple d'application, consultezthe section called “Dépannage des problèmes liés à l'exemple d'application” (p. 61).

   Vous pouvez également ajouter le`--verbosity debug`à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'elle fait. Ces informations peuvent vous fournir l'aide dont vous avez besoin pour corriger le problème.

Affichez les messages de l'exemple d'application dans leAWS IoTconsole

Vous pouvez voir les messages de l'application exemple lorsqu'ils passent par le courtier de messages à l'aide de l'outilClient MQTTdans leAWS IoTconsole.

Pour afficher les messages MQTT publiés par l'exemple d'application

1. Vérifiez Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62). Cela vous apprendra à utiliser l'outilClient MQTTdans leAWS IoTconsolePour afficher les messages MQTT au fur et à mesure qu'ils transitent par le courtier de messages.
2. Ouverture d'Client MQTTdans leAWS IoTconsole.
3. Abonnez-vous à la rubrique,topic_1.
4. Dans la fenêtre de votre ligne de commande, exécutez à nouveau l'exemple d'application et regardez les messages dans la fenêtreClient MQTTdans leAWS IoTconsole.

   Python

```
cd ~/aws-iot-device-sdk-python-v2/samples
python3 pubsub.py --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

JavaScript

```
cd ~/aws-iot-device-sdk-js-v2/samples/node/pub_sub
node dist/index.js --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Dépannage des problèmes liés à l'exemple d'application

Si vous rencontrez une erreur lorsque vous essayez d'exécuter l'exemple d'application, voici quelques éléments à vérifier.

Vérifier le certificat

Si le certificat n'est pas actif,AWS IoTn'acceptera aucune tentative de connexion qui l'utilise à des fins d'autorisation. Lors de la création de votre certificat, il est facile de négliger leActivate. Heureusement, vous pouvez activer votre certificat à partir duAWS IoTconsole.

Pour vérifier l'activation de votre certificat

1. DansAWS IoTconsoleDans le menu de gauche, choisissezSecure, puisCertificats.
2. Dans la liste des certificats, recherchez le certificat que vous avez créé pour l'exercice et vérifiez son état dans la listeÉtatcolumn.

   Si vous ne vous souvenez pas du nom du certificat, vérifiez ceux qui sontInactifpour voir s'ils peuvent être ceux que vous utilisez.

   Choisissez le certificat dans la liste pour ouvrir sa page de détails. Dans la page détaillée, vous pouvez voir sonCréation d'une datePour vous aider à identifier le certificat.
3. Pour activer un certificat inactifSur la page des détails du certificat, choisissezActions, puisActivate.

Si vous avez trouvé le certificat correct et son actif, mais que vous rencontrez toujours des problèmes lors de l'exécution de l'exemple d'application, vérifiez sa stratégie comme décrit l'étape suivante.

Vous pouvez également essayer de créer une nouvelle chose et un nouveau certificat en suivant les étapes décrites dansthe section called “Créer un objet d'objet” (p. 37). Si vous créez une nouvelle chose, vous devrez lui donner un nouveau nom de chose et télécharger les nouveaux fichiers de certificat sur votre appareil.

Vérifiez la stratégie attachée au certificat

Les stratégies autorisent les actions dansAWS IoT. Si le certificat utilisé pour la connexion àAWS IoTn'a pas de stratégie ou n'a pas de stratégie lui permettant de se connecter, la connexion sera refusée, même si le certificat est actif.

Pour vérifier les stratégies attachées à un certificat

1. Recherchez le certificat comme décrit dans l'élément précédent et ouvrez sa page de détails.
2. Dans le menu de gauche de la page de détails du certificat, choisissezStratégiesPour afficher les stratégies attachées au certificat
3. S'il n'y a pas de stratégie attachée au certificat, ajoutez-en une en choisissant l'optionActions, puis en choisissantAttacher une stratégie.

   Choisissez la stratégie que vous avez créée précédemment dansthe section called “CréerAWS IoTresources” (p. 35).

4. Si une stratégie est attachée, choisissez la vignette de stratégie pour ouvrir sa page de détails.

   Dans la page de détails, consultez la pageDocument de stratégiepour vous assurer qu'il contient les mêmes informations que celles que vous avez créées dansthe section called “Création d'une stratégie AWS IoT” (p. 36).

Vérifiez la ligne de commande.

Assurez-vous que vous avez utilisé la ligne de commande correcte pour votre système. Les commandes utilisées sur les systèmes Linux/macOS sont souvent différentes de celles utilisées sur les systèmes Windows.

Vérifiez l'adresse du point de terminaison

Passez en revue la commande que vous avez saisie et vérifiez l'adresse du point de terminaison dans votre commande à celle de votreAWS IoTconsole.

Vérifiez les noms de fichiers des fichiers de certificat

Comparez les noms de fichiers de la commande que vous avez entrée aux noms de fichiers des fichiers de certificats dans la boîte de dialogue`certs`directory.

Certains systèmes peuvent exiger que les noms de fichiers soient entre guillemets pour fonctionner correctement.

Vérifiez l'installation du kit SDK

Assurez-vous que l'installation de votre SDK est complète et correcte.

En cas de doute, réinstallez le SDK sur votre appareil. Dans la plupart des cas, il s'agit de trouver la section du tutoriel intituléInstaller leAWS IoTKit SDK des appareils pour***Langage SDK***et en suivant à nouveau la procédure.

Si vous utilisez leAWS IoTKit SDK des appareils pour JavaScript, n'oubliez pas d'installer les exemples d'applications avant d'essayer de les exécuter. L'installation du SDK n'installe pas automatiquement les exemples d'applications. Les exemples d'applications doivent être installés manuellement après l'installation du SDK.

Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT

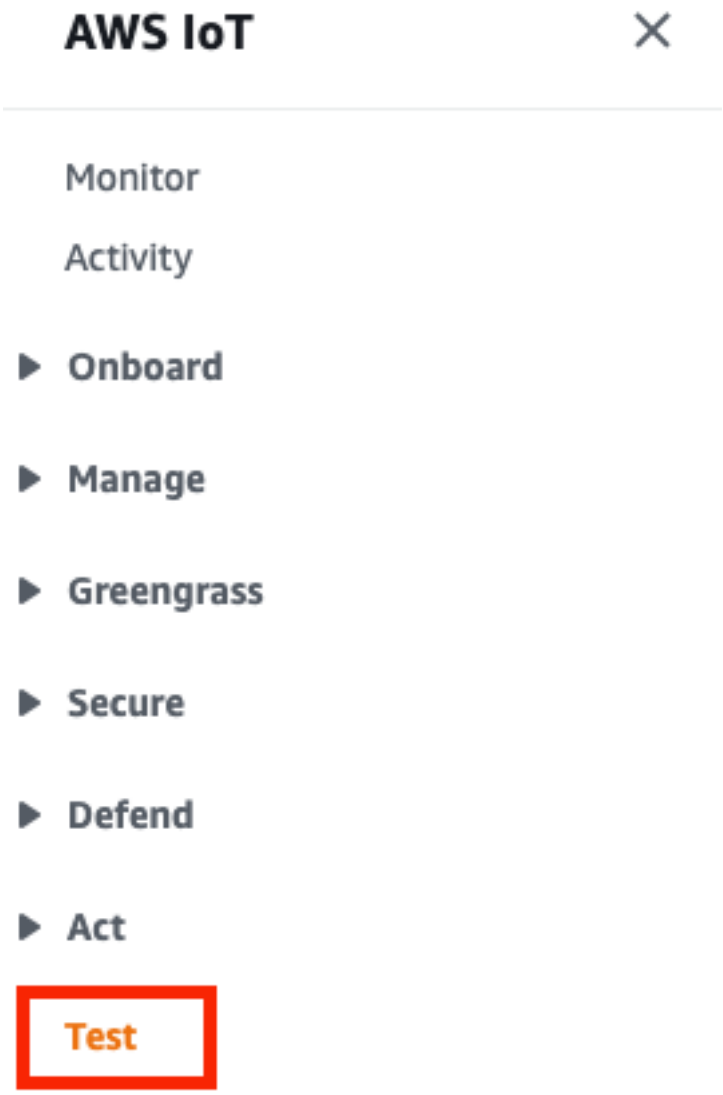
Cette section décrit comment utiliser l'objetAWS IoTClient MQTT dans leAWS IoTconsolePour regarder les messages MQTT envoyés et reçus parAWS IoT. L'exemple utilisé dans cette section concerne les exemples utilisés dansMise en route avec AWS IoT Core (p. 18) ; vous pouvez toutefois remplacer l'objet***topicName***utilisé dans les exemples avec n'importe quelnom de rubrique ou filtre de rubrique (p. 88)utilisé par votre solution IoT.

Les appareils publient des messages MQTT qui sont identifiés parRubriques (p. 88)pour communiquer leur état àAWS IoT, etAWS IoTpublie des messages MQTT pour informer les appareils et les applications des changements et des événements. Vous pouvez utiliser le client MQTT pour vous abonner à ces rubriques et consulter les messages au fur et à mesure qu'ils apparaissent. Vous pouvez également utiliser le client MQTT pour publier des messages MQTT sur des appareils et services abonnés dans votre Compte AWS .

Affichage des messages MQTT dans le client MQTT

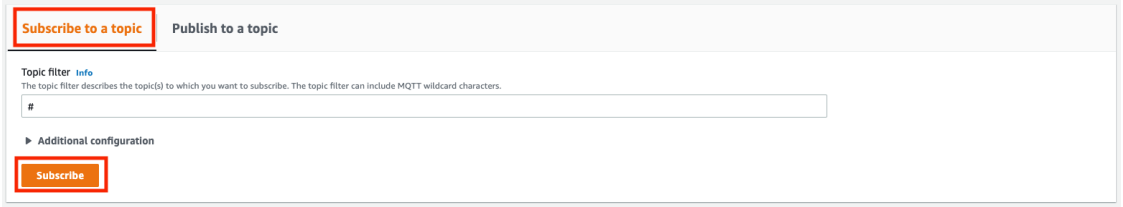
Pour afficher les messages MQTT dans le client MQTT

1. DansAWS IoTconsoleDans le menu de gauche, choisissezTest.

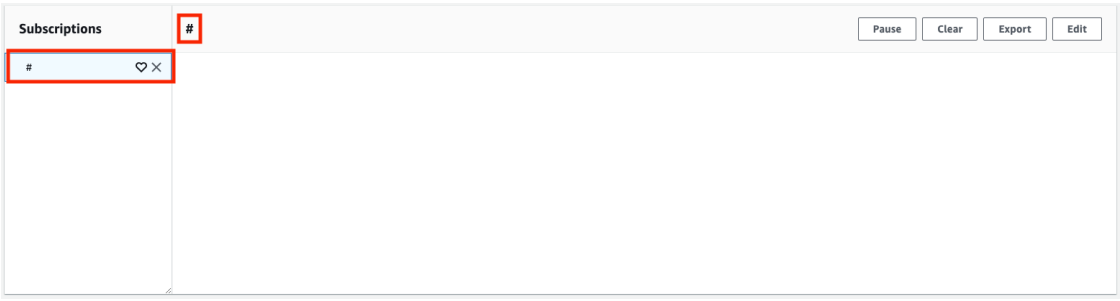


2. S'abonner à*topicName*sur lequel votre appareil publie. Pour l'exemple d'application de démarrage, abonnez-vous à**#**, qui s'abonne à toutes les rubriques des messages.

   Continuant avec l'exemple de démarrage, sur leS'abonner à une rubrique, dans l'ongletFiltre de rubriquesChamp, entrez**#**, puisS'abonner.



La page du journal des messages de rubrique,#s'ouvre et#s'affiche dans laSubscriptions.



3. Si le périphérique que vous avez configuré dansthe section called “Configurer votre appareil” (p. 39)exécute l'exemple de programme, vous devriez voir les messages qu'il envoie àAWS IoTdans le#Journal des messages. Les entrées du journal des messages apparaîtront sous lePublierlorsque les messages avec la rubrique abonnée sont reçus parAWS IoT.
4. Dans la page#, vous pouvez également publier des messages dans une rubrique, mais vous devez spécifier le nom de la rubrique. Vous ne pouvez pas publier sur le#.

   Les messages publiés sur les rubriques abonnées apparaissent dans le journal des messages à mesure qu'ils sont reçus, avec le message le plus récent en premier.

Messages de résolution des problèmes

Utiliser le filtre de rubrique générique

Si vos messages ne s'affichent pas dans le journal des messages comme prévu, essayez de vous abonner à un filtre de rubrique générique, comme décrit à la sectionFiltres de rubrique (p. 90). Le filtre de rubrique générique multi-niveaux MQTT est le signe de hachage ou de livre (#) et peut être utilisé comme filtre de rubrique dans larubrique abonnementfield.

Abonnement aux#s'abonne à chaque rubrique reçue par le courtier de messages. Vous pouvez affiner le filtre en remplaçant les éléments du chemin de filtre de rubrique par un#caractère Wild-card à plusieurs niveaux ou le caractère « + » à un seul niveau.

Lorsque vous utilisez des caractères génériques dans un filtre de rubrique

- Le caractère générique à plusieurs niveaux doit être le dernier caractère du filtre de rubrique.
- Le chemin de filtre de rubrique ne peut comporter qu'un seul caractère générique de niveau par niveau de rubrique.

Exemples :

Filtre de rubriques	Affiche les messages avec
`#`	Un nom de la rubrique
`topic_1/#`	Un nom de rubrique qui commence par`topic_1/`
`topic_1/level_2/#`	Un nom de rubrique qui commence par`topic_1/level_2/`
`topic_1/+/level_3`	Un nom de rubrique qui commence par`topic_1/`, se termine par`/level_3`, et a un élément de n'importe quelle valeur entre les deux.

Pour plus d'informations sur les filtres de rubrique, consultezFiltres de rubrique (p. 90).

Rechercher les erreurs de nom de rubrique

Les noms de rubrique MQTT et les filtres de rubrique sont sensibles à la casse. Si, par exemple, votre appareil publie des messages sur`Topic_1`(avec un capitalT) au lieu de`topic_1`, rubrique à laquelle vous vous êtes abonné, ses messages n'apparaissent pas dans le client MQTT. L'abonnement au filtre de rubrique générique indique toutefois que le périphérique publie des messages et que vous pouvez voir qu'il utilise un nom de rubrique qui n'est pas celui que vous attendiez.

Publication de messages MQTT à partir du client MQTT

Pour publier un message dans une rubrique MQTT

1. Sur la page du client MQTT, dans la listePublier dans une rubrique, dans l'ongletNom de la rubrique, entrez le*`topicName`*de votre message. Pour cet exemple, utilisez **`my/topic`**.

   Note

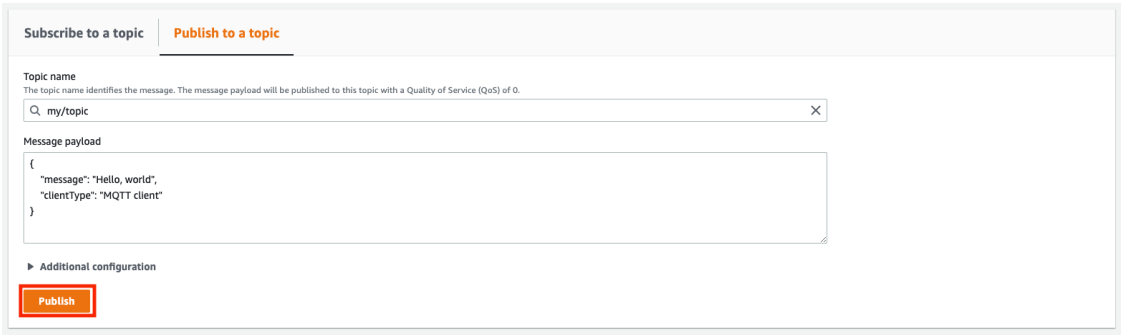
   N'utilisez pas d'informations personnelles identifiables dans les noms de rubriques, que ce soit dans le client MQTT ou dans l'implémentation de votre système. Les noms de rubrique peuvent apparaître dans les communications et les rapports non chiffrés.
2. Dans la fenêtre de charge utile du message, saisissez le contenu JSON suivant :

```
{
    "message": "Hello, world",
    "clientType": "MQTT client"
}
```

3. ChoisissezPublierPour publier votre message dansAWS IoT.

   Note

   Vérifiez que vous êtes abonné au programmemon/thèmeavant de publier votre message.

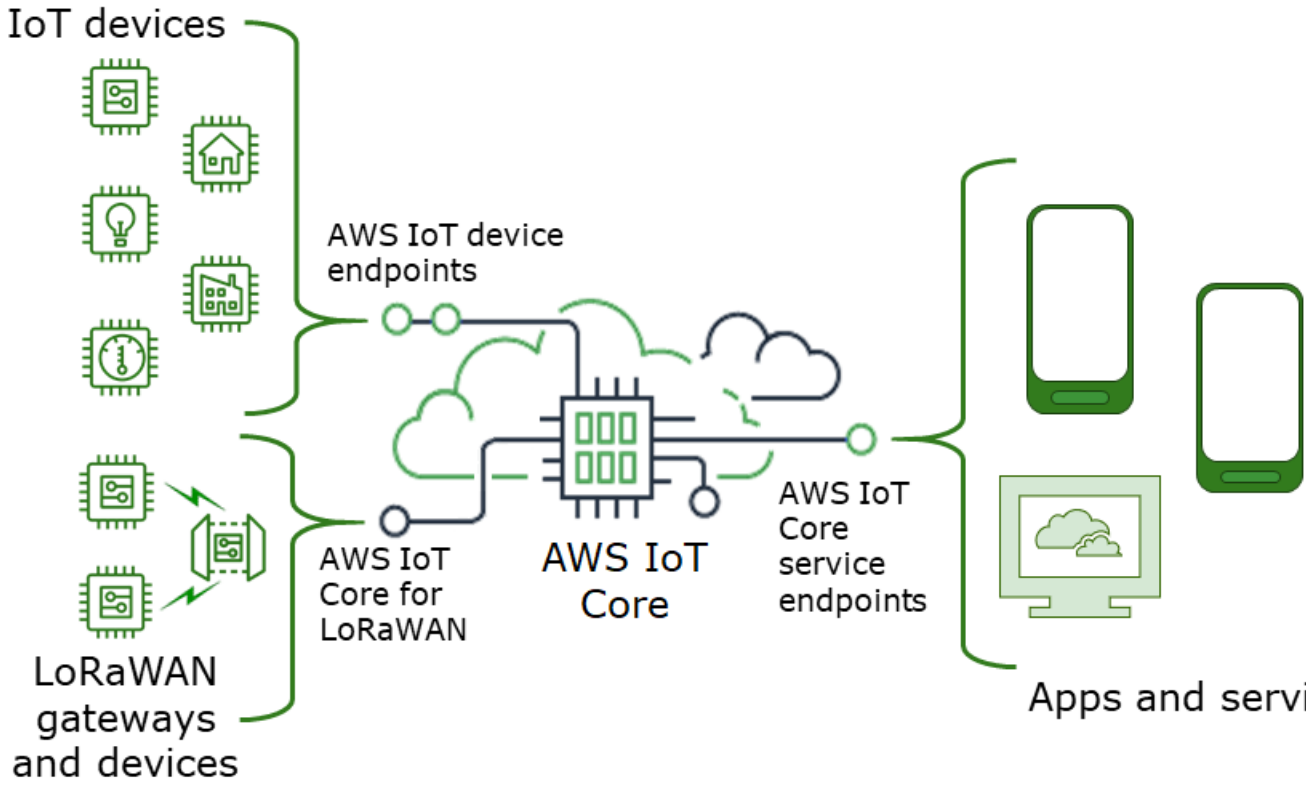


4. DansSubscriptionsListe, choisissezmon/thèmepour voir le message. Le message suivant devrait s'afficher dans le client MQTT sous la fenêtre de charge utile du message de publication.

Vous pouvez publier des messages MQTT dans d'autres rubriques en modifiant la propriété*topicName*dans leNom de la rubriqueet en choisissant le champPublier.

Connexion à AWS IoT Core

AWS IoT Core prend en charge les connexions avec les appareils IoT, les passerelles sans fil, les services et les applications. Les périphériques se connectent au AWS IoT Core Pour qu'ils puissent envoyer des données à destination et en provenance d'AWS IoTservices et autres appareils. Les applications et autres services se connectent également à AWS IoT Core pour contrôler et gérer les appareils IoT et traiter les données de votre solution IoT. Cette section décrit comment choisir le meilleur moyen de se connecter et de communiquer avec AWS IoT Core pour chaque aspect de votre solution IoT.



Il existe plusieurs méthodes pour interagir avecAWS IoT. Les applications et les services peuvent utiliserPoints de terminaison de service AWS IoT Core (p. 67)et les appareils peuvent se connecter à AWS IoT Core à l'aide du kitAWS IoTpoints de terminaison de périphérique (p. 68)ou AWS IoT Core pour passerelles et appareils LoRaWan (p. 69).

Points de terminaison de service AWS IoT Core

La . AWS IoT Core offrent un accès aux fonctions qui contrôlent et gèrent vosAWS IoTsolution.

- Endpoints

  La . AWS IoT Core les points de terminaison du service sont spécifiques à la région et sont répertoriés dans AWS IoT Core Points de terminaison et quotas. Les formats de la AWS IoT Core sont les suivantes.

Usage du point de terminaison	Format de point de terminaison	Serve
AWS IoT Core Contrôle	`iot.aws-region.amazonaws.com`	AWS IoT Plan de contrôle API
Données AWS IoT Core	Voir ??? (p. 76).	AWS IoT API de plan de données
AWS IoT Core données sur les tâches	Voir ??? (p. 76).	AWS IoT API de plan de données d'emploi
Tunneling sécurisé AWS IoT Core	`api.tunneling.iot.aws-region.amazonaws.com`	AWS IoT API Tunneling sécurisé

- Kits de développement logiciel et outils

  La .AWSKits SDKfournissent une prise en charge spécifique à la langue AWS IoT Core API et les API d'autresAWSServices . La .AWSKits SDK Mobilefournissent aux développeurs d'applications une prise en charge spécifique à la plate-forme pour AWS IoT Core API, et d'autresAWSServices sur les appareils mobiles.

  La .AWS CLIfournit un accès en ligne de commande aux fonctions fournies par leAWS IoTPoints de terminaison du service.AWS Tools for PowerShellfournit des outils pour gérerAWSservices et ressources dans l'environnement de script PowerShell.

- Authentication

  Les points de terminaison de service utilisent les utilisateurs IAM etAWSinformations d'identification pour authentifier les utilisateurs.

- En savoir plus

  Pour plus d'informations et des liens vers les références du SDK, voirthe section called “Connexion à AWS IoT Core Points de terminaison de service” (p. 69).

AWS IoTpoints de terminaison de périphérique

La .AWS IoTprennent en charge la communication entre vos appareils IoT etAWS IoT.

- Endpoints

  Les points de terminaison de l'appareil sont spécifiques à votre compte et vous pouvez voir ce qu'ils sont à l'aide de ladescribe-endpointCommande de.

  Pour plus d'informations sur ces points de terminaison et les fonctions qu'ils prennent en charge, consultez.the section called “AWS IoTdonnées de périphérique et points de terminaison de service” (p. 76).

- SDKs

  La .AWS IoTKits SDK pour les appareils (p. 78)fournissent une prise en charge spécifique à la langue des protocoles MQTT (Message Queueing Telemetry Transport) et WebSocket Secure (WSS), que les périphériques utilisent pour communiquer avecAWS IoT.Kits SDK AWS Mobile (p. 74)fournissent également un support pour les communications des appareils MQTT,AWS IoTAPI et les API d'autresAWSServices sur les appareils mobiles.

- Authentication

Les points de terminaison du périphérique utilisent des certificats X.509 ouAWSUtilisateurs IAM avec des informations d'identification pour authentifier les utilisateurs.

- En savoir plus

  Pour plus d'informations et des liens vers les références du SDK, voirthe section called “Kits SDK pour les appareils AWS IoT” (p. 78).

AWS IoT Core pour passerelles et appareils LoRaWan

AWS IoT Core pour LoRaWan connecte des passerelles et des appareils sans fil à AWS IoT Core .

- Endpoints

  AWS IoT Core pour LoRaWan gère les connexions de passerelle au compte et à la région AWS IoT Core Points de terminaison . Les passerelles peuvent se connecter au point de terminaison CUPS (Configuration and Update Server) de votre compte AWS IoT Core pour LoRaWAN fournit.

Usage du point de terminaison	Format de point de terminaison	Serve
Serveur de configuration et de mise à jour (CUPS)	*account-specific-prefix*`.cups.lorawan.`*aws-region*`.amazonaws.com:443`	Communication de passerelle avec le serveur de configuration et de mise à jour fourni par AWS IoT Core pour LoRaWAN
Serveur réseau LoRaWan (LNS)	*account-specific-prefix*`.gateway.lorawan.`*aws-region*`.amazonaws.com:443`	Communication de passerelle avec le serveur réseau LoRaWan fourni par AWS IoT Core pour LoRaWAN

- SDKs

  La .AWS IoTAPI sans fil qui AWS IoT Core pour LoRaWan est basé sur est pris en charge par leAWSKIT SDK. Pour de plus amples informations, veuillez consulterAWSKits SDK et outils.

- Authentication

  AWS IoT Core Pour les communications des appareils LoRawan utilisent les certificats X.509 pour sécuriser les communications avecAWS IoT.

- En savoir plus

  Pour plus d'informations sur la configuration et la connexion de périphériques sans fil, consultez. AWS IoT Core pour LoRaWAN (p. 1062).

Connexion à AWS IoT Core Points de terminaison de service

Vous pouvez accéder aux fonctions que l' AWS IoT Core fournissent à l'aide de laAWS CLI, leAWSSDK pour votre langue préférée, ou en appelant directement l'API REST. Nous vous recommandons d'utiliser le kitAWS CLIou unAWSSDK pour interagir avec AWS IoT Core car ils intègrent les bonnes pratiques pour

appelerAWSServices . L'appel direct des API REST est une option, mais vous devez fournirinformations d'identification de sécurité nécessairesqui permettent d'accéder à l'API.

Note

Les appareils IoT doivent utiliserKits SDK pour les appareils AWS IoT (p. 78). Les SDK de périphérique sont optimisés pour une utilisation sur les périphériques, prennent en charge la communication MQTT avecAWS IoT, et supporte leAWS IoTAPI les plus utilisées par les appareils. Pour plus d'informations sur les kits SDK pour les appareils et les fonctions qu'ils fournissent, consultez.Kits SDK pour les appareils AWS IoT (p. 78).
Les appareils mobiles doivent utiliserKits SDK AWS Mobile (p. 74). Les SDK mobiles prennent en charge lesAWS IoTLes API, les communications de périphériques MQTT et les API d'autresAWSServices sur les appareils mobiles. Pour plus d'informations sur les kits SDK Mobile et les fonctions qu'ils fournissent, consultez.Kits SDK AWS Mobile (p. 74).

Les sections suivantes décrivent les outils et les kits SDK que vous pouvez utiliser pour développer et interagir avecAWS IoTet autresAWSServices . Pour obtenir la liste complète desAWSoutils et kits de développement disponibles pour créer et gérer des applications surAWS, voirOutils sur lesquels s'appuyerAWS.

AWS CLI pour AWS IoT Core

La .AWS CLIfournit un accès en ligne de commande àAWSAPI.

- Installation

  Pour plus d'informations sur l'installation duAWS CLI, voirInstallation du kitAWS CLI.
- Authentication

  La .AWS CLIutilise les informations d'identification de votre Compte AWS .
- Reference

  Pour de plus amples informations sur l'AWS CLICommandes de l'pour ces AWS IoT Core services, voir :
  - AWS CLI Référence des commandes pour l'IoT
  - AWS CLI Référence des commandes pour les données IoT
  - AWS CLI Référence de commande pour les données de travaux IoT
  - AWS CLI Référence de commande pour le tunnel sécurisé IoT

Pour les outils à gérerAWSservices et ressources dans l'environnement de script PowerShell, consultezAWSTools for PowerShell.

Kits SDK AWS

avecAWSLes SDK, vos applications et appareils compatibles peuvent appelerAWS IoTAPI et les API d'autresAWSServices . Cette section fournit des liens vers leAWSLes kits SDK et la documentation de référence de l'API pour les API de l'objet AWS IoT Core Services .

La .AWSKits SDK pris en charge AWS IoT Core API

- AWS IoT
- AWS IoT Plan de données
- AWS IoT Plan de données des jobs
- AWS IoT Tunneling sécurisé

- AWS IoT Sans fil

C++

Pour installerAWS SDK for C++et utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionPremiers pas avec l'AWSKit SDK pour C++

   Ces instructions décrivent comment :
   - Installation et génération du kit SDK à partir des fichiers source
   - Fournissez des informations d'identification pour utiliser le SDK avec votre Compte AWS
   - Initialiser et arrêter le SDK dans votre application ou service
   - Créer un projet CMake pour créer votre application ou service
2. Créez et exécutez un exemple d'application. Pour les exemples d'applications qui utilisent leAWSSDK for C++, consultezAWS SDK for C++Exemples de code.

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for C++prend en charge

- Documentation de référence AWS# IoT# IotClient
- Aws# IotDataPlane# Documentation de référence IotDataPlanneClient
- Aws# IotJobsDataPlane# Documentation de référence IotJobsDataPlaneClient
- Aws# IotSecureTunneling# Documentation de référence IOTSecureTunnelingClient

Go

Pour installerAWS SDK for Goet utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionMise en route avec lesAWS SDK for Go

   Ces instructions décrivent comment :
   - Installation de la AWS SDK for Go
   - Obtenez des clés d'accès pour le SDK pour accéder à votre Compte AWS
   - Importer des paquets dans le code source de nos applications ou services
2. Créez et exécutez un exemple d'application. Pour les exemples d'applications qui utilisent leAWS SDK for Go, voirAWS SDK for GoExemples de code.

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for Goprend en charge

- Documentation de références IoT
- Documentation de référence IotDataPlane
- Documentation de référence IotJobsDataPlane
- Documentation de référence IOSecureTunneling

Java

Pour installerAWS SDK for Javaet utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionDémarrer avecAWS SDK for Java 2.x

   Ces instructions décrivent comment :
   - S'inscrire àAWSet créez un utilisateur IAM

- Téléchargement du kit SDK
- ConfigurationAWSInformations d'identification et région
- Utilisation du kit SDK avec Apache Maven
- Utilisez le kit SDK avec Gradle

2. Créez et exécutez un exemple d'application à l'aide de l'un desAWS SDK for Java 2.xExemples de code.
3. Vérifiez laDocumentation de références d'API SDK

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for Javaprend en charge

- Documentation de références IOTClient
- Documentation de référence IotDataPlaneClient
- Documentation de référence IotJobsDataPlanceClient
- Documentation de référence IotSecureTunnelingClient

JavaScript

Pour installerAWS SDK for JavaScriptet utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionConfiguration de l'AWS SDK for JavaScript. Ces instructions s'appliquent à l'utilisation duAWS SDK for JavaScriptdans le navigateur et avec Node.JS. Assurez-vous de suivre les instructions qui s'appliquent à votre installation.

   Ces instructions décrivent comment :
   - Vérifiez les prérequis
   - Installation du kit SDK pour JavaScript
   - Chargez le kit SDK pour JavaScript
2. Créez et exécutez un exemple d'application pour commencer à utiliser le SDK comme décrit l'option de démarrage de votre environnement.
   - Commencez à utiliser le kitAWSKit SDK for JavaScript in the Browser, ou
   - Commencez à utiliser le kitAWSKit SDK for JavaScript in Node.js Node.js

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for JavaScriptprend en charge

- `AWS.Iot reference documentation`
- `AWS.IotData reference documentation`
- `AWS.IotJobsDataPlane reference documentation`
- `AWS.IotSecureTunneling reference documentation`

.NET

Pour installerAWS SDK for .NETet utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionConfiguration de votreAWS SDK for .NETenvironment
2. Suivez les instructions de la sectionConfiguration de votreAWS SDK for .NETproject

   Ces instructions décrivent comment :
   - Démarrer un projet
   - Obtenir et configurerAWScredentials

- InstallAWSPaquets SDK

3. Créez et exécutez l'un des exemples de programmes dansUtilisation deAWSServices dansAWSKit SDK pour .NET
4. Vérifiez laDocumentation de références d'API SDK

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for .NETprend en charge

- Documentation de référence Amazon.iot.model
- Documentation de référence Amazon.iotData.model
- Documentation de référence Amazon.iotJobsDataPlane.model
- Documentation de référence Amazon.iotSecureTunneling.model

PHP

Pour installerAWS SDK for PHPet utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionMise en route avec lesAWS SDK for PHPVersion 3

   Ces instructions décrivent comment :
   - Vérifiez les prérequis
   - Installer le SDK
   - Appliquer le SDK à un script PHP
2. Créez et exécutez un exemple d'application à l'aide de l'un desAWS SDK for PHPExemples de code de version 3

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for PHPprend en charge

- Documentation de références IOTClient
- Documentation de référence IotDataPlaneClient
- Documentation de référence IotJobsDataPlanceClient
- Documentation de référence IotSecureTunnelingClient

Python

Pour installerAWS SDK for Python (Boto3)et utilisez-le pour vous connecter àAWS IoT :

1. Suivez les instructions de la sectionAWS SDK for Python (Boto3)Quickstart

   Ces instructions décrivent comment :
   - Installer le SDK
   - Configuration du kit SDK
   - Utilisez le kit SDK dans votre code
2. Créez et exécutez un exemple de programme qui utilise la fonctionAWS SDK for Python (Boto3)

   Ce programme affiche les options de journalisation actuellement configurées du compte. Une fois que vous avez installé le kit SDK et que vous avez configuré pour votre compte, vous devez pouvoir exécuter ce programme.

```
import boto3
import json
```

```
# initialize client
iot = boto3.client('iot')

# get current logging levels, format them as JSON, and write them to stdout
response = iot.get_v2_logging_options()
print(json.dumps(response, indent=4))
```

Pour plus d'informations sur la fonction utilisée dans cet exemple, consultez.the section called “Configurer la journalisation AWS IoT” (p. 353).

Documentation sur l'pour AWS IoT Core Services que le kitAWS SDK for Python (Boto3)prend en charge

- Documentation de références IoT
- Documentation de référence IotDataPlane
- Documentation de référence IotJobsDataPlane
- Documentation de référence IOSecureTunneling

Ruby

Pour installerAWS SDK for Rubyet utilisez-le pour vous connecter àAWS IoT :

- Suivez les instructions de la sectionMise en route avec lesAWS SDK for Ruby

  Ces instructions décrivent comment :
  - Installer le SDK
  - Configuration du kit SDK
- Crée et exécute le kitDidacticiel Hello World

Documentation sur l'pour AWS IoT Core Services que le kitAWSPrise en charge du SDK for Ruby

- Documentation de référence du client Aws# IoT# Client
- Aws# IotDataPlane# Documentation de référence client
- Aws# IotJobsDataPlane# Documentation de référence client
- Aws# IOTSecureTunneling# Documentation de référence du client

Kits SDK AWS Mobile

La .AWSLes SDK mobiles fournissent aux développeurs d'applications mobiles une prise en charge spécifique de la plate-forme pour les API du AWS IoT Core , la communication des appareils IoT à l'aide de MQTT et les API d'autresAWSServices .

Android

AWS Mobile SDK for Android

La .AWS Mobile SDK for Androidcontient une bibliothèque, des exemples et une documentation pour développeurs pour développer des applications mobiles connectées à l'aide d'AWS. Ce kit SDK comprend également la prise en charge des communications des appareils MQTT et l'appel des API de l' AWS IoT Core Services . Pour de plus amples informations, consultez les ressources suivantes :

- AWS Kit SDK Mobile pour Android sur GitHub
- AWS Kit SDK Mobile pour Android — Readme
- AWS Kit SDK Mobile pour Android — Echantillons
- AWS Référence des kits SDK for Android
- Documentation de référence de classe AWSIOTClient

iOS

AWS Mobile SDK for iOS

La .AWS Mobile SDK for iOSest un kit de développement logiciel open source, distribué sous une licence open source Apache. Le SDK for iOS fournit une bibliothèque, des exemples de code et une documentation pour aider les développeurs à créer des applications mobiles connectées à l'aide d'AWS. Ce kit SDK comprend également la prise en charge des communications des appareils MQTT et l'appel des API de l' AWS IoT Core Services . Pour de plus amples informations, consultez les ressources suivantes :

- AWS Mobile SDK for iOS sur GitHub
- AWS SDK for iOS Readme
- AWS SDK for iOS — Exemples
- AWS IoT Docs de référence de classe dans leAWSSDK pour iOS

API REST du kit AWS IoT Core services

Les API REST du AWS IoT Core peuvent être appelés directement à l'aide de requêtes HTTP.

- URL de point de terminaison

  Les points de terminaison de service qui exposent les API REST du AWS IoT Core varient d'une région à l'autre et sont répertoriés dans AWS IoT Core Points de terminaison et quotas. Vous devez utiliser le point de terminaison pour la région qui possède leAWS IoTLes ressources auxquelles vous souhaitez accéder, carAWS IoTLes ressources sont spécifiques à la région.
- Authentication

  Les API REST du AWS IoT Core Utilisation des servicesAWSInformations d'identification IAM pour l'authentification. Pour de plus amples informations, veuillez consulterSignatureAWSDemandes d'APIdans leAWSRéférence générale.
- Référence d'API

  Pour plus d'informations sur les fonctions spécifiques fournies par les API REST du AWS IoT Core services, voir :
  - Référence d'API pour l'IoT.
  - Référence API pour les données IoT.
  - Référence API pour les données des travaux IoT.
  - Référence API pour le tunnel sécurisé IoT.

Connexion d'appareils àAWS IoT

Les appareils se connectent àAWS IoTet les autres services via AWS IoT Core . à travers AWS IoT Core , les appareils envoient et reçoivent des messages à l'aide de points de terminaison spécifiques à votre compte. La .the section called “Kits SDK pour les appareils AWS IoT” (p. 78)prennent en charge les

communications de périphériques à l'aide des protocoles MQTT et WSS. Pour plus d'informations sur les protocoles que les appareils peuvent utiliser, consultez.the section called “Protocoles de communication de périphérique” (p. 79).

Agent de messages

AWS IoTgère la communication de l'appareil via un courtier de messages. Les appareils et les clients publient des messages au courtier de messages et s'abonnent également aux messages publiés par le courtier de messages. Les messages sont identifiés par une application définietopic (p. 88). Lorsque l'agent de messages reçoit un message publié par un appareil ou un client, il le republie sur les appareils et les clients abonnés à la rubrique du message. Le courtier de messages transmet également les messages à laAWS IoT Règles (p. 394), qui peut agir sur le contenu du message.

AWS IoTSécurité des messages

Connexions d'appareils àAWS IoTusethe section called “Certificats client X.509” (p. 236)andAWSSignature V4pour l'authentification. Les communications des appareils sont sécurisées par TLS version 1.2 etAWS IoTnécessite que les périphériques envoient leExtension SNI (Server Name Indication)lorsqu'ils se connectent. Pour de plus amples informations, veuillez consulter Sécurité du transport dans AWS IoT.

AWS IoTdonnées de périphérique et points de terminaison de service

Chaque compte dispose de plusieurs terminaux d'appareil qui sont uniques au compte et prennent en charge des fonctions IoT spécifiques. La .AWS IoTLes terminaux de données de périphérique prennent en charge un protocole de publication/abonnement conçu pour les besoins de communication des appareils IoT. Toutefois, d'autres clients, tels que les applications et les services, peuvent également utiliser cette interface si leur application nécessite les fonctionnalités spécialisées fournies par ces terminaux. La .AWS IoTles terminaux de service de périphériques prennent en charge l'accès centré sur les périphériques aux services de sécurité et de gestion.

Pour connaître le point de terminaison de données de l'appareil de votre compte, vous pouvez le trouver dans leParamètresPage de votre AWS IoT Core console

Pour connaître le point de terminaison de l'appareil de votre compte à une fin spécifique, y compris le point de terminaison de données de l'appareil, utilisez ledescribe-endpointCommande CLI affichée ici, ou la commande`DescribeEndpoint`API REST et fournissez l'interface*`endpointType`*Valeur du paramètre à partir du tableau suivant.

```
aws iot describe-endpoint --endpoint-type endpointType
```

Cette commande renvoie un*`iot-endpoint`*Dans le format suivant :*`account-specific-prefix`*`.iot.`*`aws-region`*`.amazonaws.com`.

La .`DescribeEndpoint`L'API n'a pas besoin d'être interrogée chaque fois qu'un nouveau périphérique est connecté. Les points de terminaison que vous créez persistent pour toujours et ne changent pas une fois qu'ils sont créés.

Chaque client dispose d'un`iot:Data-ATS`et un`iot:Data`. Chaque point de terminaison utilise un certificat X.509 pour authentifier le client. Nous recommandons vivement aux clients d'utiliser le type de point de terminaison `iot:Data-ATS` le plus récent pour éviter les problèmes liés à la méfiance généralisée à l'égard des autorités de certification Symantec. Nous fournissons le`iot:Data`Pour les appareils pour récupérer les données des anciens points de terminaison qui utilisent des certificats VeriSign pour assurer la compatibilité ascendante. Pour de plus amples informations, veuillez consulter Authentification du serveur.

AWS IoTpoints de terminaison pour les périphériques

Usage du point de terminaison	***endpointType***valeur	Description
Données IoT	`iot:Data-ATS`	Utilisé pour envoyer et recevoir des données depuis le courtier de messages,Device Shadow (p. 547), etMoteur de règles (p. 394)Composants d'AWS IoT. `iot:Data-ATS`renvoie un point de terminaison de données signées ATS.
Données IoT (hérité)	`iot:Data`	`iot:Data`renvoie un point de terminaison de données signées VeriSign fourni pour la rétrocompatibilité.
Accès aux informations d'identification IoT	`iot:CredentialProvider`	Utilisé pour échanger le certificat X.509 intégré d'un appareil contre des informations d'identification temporaires permettant de se connecter directement à d'autresAWSServices . Pour de plus amples informations sur la connexion à d'autres services AWS, veuillez consulter Autorisation des appels directs vers les services AWS (p. 307).
Gestion des tâches IoT	`iot:Jobs`	Utilisé pour permettre aux appareils d'interagir avec leAWS IoTService Jobs utilisant le kitOffres d'emploi Device HTTPS API (p. 683).
Conseiller sur les appareils IoT (aperçu)	`iot:DeviceAdvisor`	Type de point de terminaison de test utilisé pour tester des périphériques avec Device Advisor. Pour plus d'informations, consultez ??? (p. 998).
Bêta des données IoT (aperçu)	`iot:Data-Beta`	Type de point de terminaison réservé aux versions bêta. Pour plus d'informations sur son utilisation actuelle, consultez.??? (p. 106).

Vous pouvez également utiliser votre propre nom de domaine complet (FQDN), tel que*`example.com`*et le certificat de serveur associé pour connecter des périphériques àAWS IoTen utilisantthe section called “Les points de terminaison configurables” (p. 106), qui est actuellement en version bêta publique.

Kits SDK pour les appareils AWS IoT

La .AWS IoTLes kits SDK pour appareils vous aident à connecter vos appareils IoT à AWS IoT Core et ils prennent en charge les protocoles MQTT et MQTT sur WSS.

La .AWS IoTLes kits SDK de périphérique diffèrent desAWSSDK en ce que leAWS IoTLes kits SDK pour appareils prennent en charge les besoins de communication spécialisés des appareils IoT, mais ne prennent pas en charge tous les services pris en charge par leAWSKits SDK. La .AWS IoTLes kits SDK de périphérique sont compatibles avec leAWSLes kits SDK qui prennent en charge tous lesAWS ; cependant, ils utilisent des méthodes d'authentification différentes et se connectent à différents points de terminaison, ce qui pourrait rendre l'utilisation deAWSSDK peu pratique sur un appareil IoT.

Appareils mobiles

La .the section called “Kits SDK AWS Mobile” (p. 74)prennent en charge les communications de périphériques MQTT, certains desAWS IoTAPI de service et les API d'autresAWSServices . Si vous développez sur un appareil mobile pris en charge, consultez son SDK pour voir s'il s'agit de la meilleure option pour développer votre solution IoT.

C++

AWS IoT Kit SDK des appareils C++

La .AWS IoTC++ Device SDK permet aux développeurs de générer des applications connectées à l'aide deAWSet les API du AWS IoT Core Services . Ce kit SDK a été conçu en particulier pour les appareils qui ne sont pas limités en ressources et qui nécessitent des fonctions avancées, telles que la mise en file d'attente des messages, la prise en charge du multithreading et les dernières fonctions de langue. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT SDK de périphérique C++ v2 sur GitHub
- AWS IoT Lisez-moi du SDK de périphérique C ++ v2
- AWS IoT Exemples SDK pour les appareils C++

Python

AWS IoT Kit SDK des appareils pour Python

La .AWS IoTLe kit SDK des appareils pour Python permet aux développeurs d'écrire des scripts Python afin d'utiliser leurs appareils pour accéder auAWS IoTvia MQTT ou MQTT via le protocole WebSocket Secure (WSS). En connectant leurs appareils aux API du AWS IoT Core Les utilisateurs peuvent travailler de façon sécurisée avec l'agent de messages, les règles et le service Device Shadow qui AWS IoT Core fournit et avec d'autresAWSServices tels queAWS Lambda, Amazon Kinesis et Amazon S3, et bien plus encore.

- AWS IoT Kit SDK des appareils pour Python v2 sur GitHub
- AWS IoT Kit SDK des appareils pour Python v2 - Readme
- AWS IoT Kit SDK des appareils pour Python v2
- AWS IoT Documentation du kit SDK des appareils pour Python v2

JavaScript

AWS IoT Kit SDK des appareils pour JavaScript

La .AWS IoTLe kit SDK des appareils pour JavaScript permet aux développeurs d'écrire des applications JavaScript qui ont accès aux API de l' AWS IoT Core À l'aide de MQTT ou MQTT via le protocole WebSocket. Il peut être utilisé dans des environnements Node.js et des applications de navigateur. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour JavaScript v2 sur GitHub
- AWS IoT Kit SDK des périphériques pour JavaScript v2 – Readme
- AWS IoT Kit SDK des appareils pour JavaScript v2
- AWS IoT Documentation du kit SDK des appareils pour JavaScript v2

Java

AWS IoT Kit SDK des appareils pour Java

La .AWS IoTLe kit SDK des appareils pour Java permet aux développeurs Java d'accéder aux API du AWS IoT Core via MQTT ou MQTT via le protocole WebSocket. Le kit SDK prend en charge le service Device Shadow. Vous pouvez accéder au service Shadows à l'aide des méthodes HTTP, notamment GET, UPDATE et DELETE. Le kit SDK prend également en charge un modèle d'accès aux shadows simplifié, qui permet aux développeurs d'échanger des données avec des shadows d'objets en utilisant des méthodes getter et setter, sans avoir à sérialiser ou désérialiser des documents JSON. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour Java v2 sur GitHub
- AWS IoT Kit SDK des périphériques pour Java v2 – Readme
- AWS IoT Kit SDK des appareils pour Java v2 — Echantillons

Embedded C

AWS IoT Kit SDK des appareils pour Embedded C

> **Important**
>
> Ce kit SDK est destiné à être utilisé par les développeurs de logiciels intégrés expérimentés.

La . AWS IoT Device SDK for Embedded C (C-SDK) est un ensemble de fichiers source C sous la licence open source MIT qui peuvent être utilisés dans des applications incorporées pour connecter en toute sécurité des appareils IoT àAWSCore IoT. Il comprend un MQTT, JSON Parser etAWS IoTBibliothèque Device Shadow. Il est distribué sous forme source et est conçu pour être intégré dans un microprogramme client avec un code d'application, d'autres bibliothèques et, éventuellement, un système d'exploitation temps réel (RTOS).

Pour le provisionnement de flotte, utilisez le`v4_beta_deprecated`Version du kit AWS IoT Device SDK for Embedded C àhttps://github.com/aws/aws-iot-device-sdk-embedded-C/tree/v4_beta_deprecated.

Le AWS IoT Device SDK for Embedded C est généralement destiné aux appareils à ressources limitées qui nécessitent un moteur d'exécution optimisé en langage C. Vous pouvez utiliser le kit SDK sur n'importe quel système d'exploitation et l'héberger sur n'importe quel type de processeur (par exemple, microcontrôleurs et MPU). Si votre appareil dispose de suffisamment de mémoire et de ressources de traitement disponibles, nous vous conseillons d'utiliser l'un des autresAWS IoTLes kits SDK des appareils et mobiles, tels que leAWS IoTKit de développement logiciel (SDK) de l'appareil pour C++, Java, JavaScript ou Python.

Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour Embedded C GitHub
- AWS IoT Kit SDK des appareils pour Embedded C – Readme
- AWS IoT Kit SDK des appareils pour Embedded C

Protocoles de communication de périphérique

AWS IoT Core prend en charge les appareils et les clients qui utilisent les protocoles MQTT et MQTT sur WebSocket Secure (WSS) pour publier et s'abonner aux messages, ainsi que les appareils et les clients qui utilisent le protocole HTTPS pour publier des messages. Tous les protocoles prennent en charge les protocoles IPv4 et IPv6. Cette section décrit les différentes options de connexion pour les appareils et les clients.

TLS v1.2

AWS IoT Core utilise le traitementTLS Version 1.2pour chiffrer toutes les communications. Les clients doivent également envoyer leExtension SNI (Server Name Indication). Les tentatives de connexion qui n'incluent pas le SNI sont refusées. Pour de plus amples informations, veuillez consulter Sécurité du transport dans AWS IoT.

La .Kits SDK pour les appareils AWS IoT (p. 78)prennent en charge MQTT et MQTT sur WSS et prennent en charge les exigences de sécurité des connexions client. Nous vous recommandons d'utiliser le kitKits SDK pour les appareils AWS IoT (p. 78)pour connecter des clients àAWS IoT.

Protocoles, mappages de ports et authentification

La façon dont un périphérique ou un client se connecte au courtier de messages à l'aide d'un point de terminaison de périphérique dépend du protocole qu'il utilise. Le tableau suivant répertorie les protocoles que l'objetAWS IoTet les méthodes d'authentification et les ports qu'ils utilisent.

Protocoles, authentification et mappages de port

Protocole	Activités prises en charge	Authentification	Port	Nom du protocole ALPN
MQTT via WebSocket	Publier/s'abonner	Signature Version 4	443	S/O
MQTT via WebSocket	Publier/s'abonner	Authentification personnalisée	443	S/O
MQTT	Publier/s'abonner	Certificat de client X.509	443†	`x-amzn-mqtt-ca`
MQTT	Publier/s'abonner	Certificat de client X.509	8883	S/O
MQTT	Publier/s'abonner	Authentification personnalisée	443†	`mqtt`
HTTPS	Publiez uniquement	Signature Version 4	443	S/O
HTTPS	Publiez uniquement	Certificat de client X.509	443†	`x-amzn-http-ca`
HTTPS	Publiez uniquement	Certificat de client X.509	8443	S/O
HTTPS	Publiez uniquement	Authentification personnalisée	443	S/O

Négociation du protocole de couche d'application (ALPN)

†Les clients qui se connectent sur le port 443 avec l'authentification de certificat client X.509 doivent implémenter leNégociation du protocole de couche d'application (ALPN)et utilisez

l'extension TLSNom du protocole ALPNrépertorié dans la liste ALPN ProtocolNameList envoyée par le client dans le cadre de`ClientHello`Message.
Sur le port 443, leIOT : données ATS (p. 77)prend en charge ALPN x-amzn-http-ca HTTP, mais leIOT : Emplois (p. 77)n'est pas le point de terminaison.
Sur le port 8443 HTTPS et le port 443 MQTT avec ALPN x-amzn-mqtt-ca,authentification personnalisée (p. 256)ne peut pas être utilisé.

Les clients se connectent à leur Compte AWS points de terminaison de l'appareil. Voirthe section called "AWS IoTdonnées de périphérique et points de terminaison de service" (p. 76)pour plus d'informations sur la recherche de points de terminaison d'appareil de votre compte.

Connexion à AWS IoT Core

Protocole	Point de terminaison ou URL
MQTT	*iot-endpoint*
MQTT via WSS	`wss://`*iot-endpoint*`/mqtt`
HTTPS	`https://`*iot-endpoint*`/topics`

Choisir un protocole pour la communication de votre appareil

Pour la plupart des communications de périphériques IoT via les terminaux de périphériques, vous devez utiliser les protocoles MQTT ou MQTT sur WSS ; toutefois, les points de terminaison de périphériques prennent également en charge HTTPS. Le tableau suivant compare comment AWS IoT Core utilise les deux protocoles pour la communication des périphériques.

AWS IoTProtocoles des périphériques côte à côte

Fonction	MQTT (p. 82)	HTTPS (p. 86)
Support de publication/abonnement	Publiez et vous abonner	Publiez uniquement
Prise en charge de SDK	AWSKits SDK pour les appareils (p. 78)prend en charge les protocoles MQTT et WSS	Pas de prise en charge du SDK, mais vous pouvez utiliser des méthodes spécifiques à la langue pour faire des requêtes HTTPS
Soutien de la qualité de service	MQTT QoS niveaux 0 et 1 (p. 83)	Aucune prise en charge QoS
Peut recevoir des messages manqués alors que l'appareil était hors connexion	Oui	Non
`clientId`appui sur le terrain	Oui	Non
Détection de déconnexion de périphérique	Oui	Non
Communications sécurisées	Oui. Voir Protocoles, mappages de ports et authentification (p. 80)	Oui. Voir Protocoles, mappages de ports et authentification (p. 80)
Durée de connexion	Jusqu'à plusieurs semaines	Jusqu'à 24 heures
Définitions de rubrique	Application définie	Application définie

Fonction	MQTT (p. 82)	HTTPS (p. 86)
Format des données du message	Application définie	Application définie
Frais généraux du protocole	LOWER	Plus élevé
Consommation d'énergie	LOWER	Plus élevé

MQTT

MQTTest un protocole de messagerie léger largement utilisé, conçu pour les appareils limités.AWS IoTla prise en charge de MQTT est basée sur leSpécification MQTT v3.1.1, avec quelques différences. Pour obtenir des informations sur l'utilisationAWS IoTdiffère de la spécification MQTT v3.1.1, consultez.the section called “AWS IoTDifférences par rapport à la spécification MQTT version 3.1.1” (p. 85).

AWS IoT Core prend en charge les connexions de périphériques qui utilisent le protocole MQTT et le protocole MQTT sur WSS. La .Kits SDK pour les appareils AWS IoT (p. 78)prennent en charge les deux protocoles et sont les moyens recommandés pour connecter des périphériques àAWS IoT. La .AWS IoTLes kits SDK de périphériques prennent en charge les fonctions nécessaires pour que les périphériques et les clients puissent se connecter et y accéder AWS IoT Core et ils prennent en charge les protocoles d'authentification que leAWS IoTles services exigent. Pour plus d'informations sur la façon de se connecter àAWS IoTà l'aide deAWSKit de développement logiciel (SDK) et liens vers des exemples deAWS IoTdans les langages pris en charge, consultez la sectionthe section called “Connexion avec MQTT à l'aide de l'outilAWS IoTKits SDK pour les appareils” (p. 82). Pour plus d'informations sur les méthodes d'authentification et les mappages de ports pour les messages MQTT, consultezProtocoles, mappages de ports et authentification (p. 80).

Bien que nous vous recommandons d'utiliser leAWS IoTKit SDK de périphérique pour se connecterAWS IoT, ils ne sont pas obligatoires. Si vous n'utilisez pas le kitAWS IoTLes kits SDK de périphérique, cependant, vous devez fournir la sécurité de connexion et de communication nécessaire. Les clients doivent envoyer leExtension SNI (Server Name Indication)dans la demande de connexion. Les tentatives de connexion qui n'incluent pas le SNI sont refusées. Pour de plus amples informations, veuillez consulter Sécurité du transport dans AWS IoT. Les clients qui utilisent les utilisateurs IAM etAWSles informations d'identification pour authentifier les clients doivent fournir lesSignature Version 4L'authentification.

Connexion avec MQTT à l'aide de l'outilAWS IoTKits SDK pour les appareils

Cette section contient des liens vers leAWS IoTKit SDK de périphérique et au code source des exemples de programmes qui illustrent comment connecter un périphérique àAWS IoT. Les exemples d'applications liées ici montrent comment se connecter àAWS IoTÀ l'aide du protocole MQTT et MQTT via WSS.

C++

Utilisation du kitAWS IoTKit SDK des appareils C++ pour connecter des appareils

- Code source d'un exemple d'application qui montre un exemple de connexion MQTT en C ++
- AWS IoT SDK de périphérique C++ v2 sur GitHub

Python

Utilisation du kitAWS IoTKit SDK des appareils pour Python pour connecter des appareils

- Code source d'un exemple d'application qui montre un exemple de connexion MQTT en Python
- AWS IoT Kit SDK des appareils pour Python v2 sur GitHub

JavaScript

Utilisation du kitAWS IoTKit SDK des appareils pour JavaScript pour connecter des appareils

- Code source d'un exemple d'application qui montre un exemple de connexion MQTT en JavaScript
- AWS IoT Kit SDK des périphériques pour JavaScript v2 sur GitHub

Java

Utilisation du kitAWS IoTKit SDK des appareils pour Java pour connecter des appareils

- Code source d'un exemple d'application qui affiche un exemple de connexion MQTT en Java
- AWS IoT Kit SDK des périphériques pour Java v2 sur GitHub

Embedded C

Utilisation du kitAWS IoTKit SDK des appareils pour Embedded C pour connecter des appareils

> **Important**
>
> Ce kit SDK est destiné à être utilisé par les développeurs de logiciels intégrés expérimentés.

- Code source d'un exemple d'application qui affiche un exemple de connexion MQTT dans Embedded C
- AWS IoT Kit SDK des périphériques pour Embedded C GitHub

Options de qualité de service (QoS) MQTT

AWS IoTet l'AWS IoTLes SDK pour les appareils prennent en chargeNiveaux de qualité de service (QoS) MQTT`0`and`1`. Le protocole MQTT définit un troisième niveau de QoS, niveau`2`, maisAWS IoTne prend pas en charge. Seul le protocole MQTT prend en charge la fonctionnalité QoS. HTTPS ne prend pas en charge la QoS service.

Ce tableau décrit comment chaque niveau QoS affecte les messages publiés par et par le courtier de messages.

Avec un niveau QoS de...	Le message est...	Commentaires
QoS niveau 0	Envoyé zéro ou plusieurs fois	Ce niveau doit être utilisé pour les messages qui sont envoyés via des liens de communication fiables ou qui peuvent être manqués sans problème.
QoS de niveau 1	Envoyé au moins une fois, puis à plusieurs reprises jusqu'à ce qu'un`PUBACK`la réponse est reçue	Le message n'est pas considéré comme complet tant que l'expéditeur n'a pas reçu un`PUBACK`réponse pour indiquer la livraison réussie.

Utilisation des sessions permanentes MQTT

Les sessions persistantes stockent les abonnements et les messages d'un client, avec une qualité de service (QoS) de 1, qui n'ont pas été reconnus par le client. Lorsqu'un périphérique déconnecté se

reconnecte à une session persistante, la session reprend, ses abonnements sont rétablis et les messages abonnés reçus avant la reconnexion et qui n'ont pas été reconnus par le client sont envoyés au client.

Création d'une session persistante

Vous créez une session permanente MQTT en envoyant un`CONNECT`et en définissant le`cleanSession`Indicateur sur`0`. Si aucune session n'existe pour le client qui envoie le fichier`CONNECT`Une nouvelle session permanente est créée. Si une session existe déjà pour le client, ce dernier reprend la session existante.

Opérations au cours d'une session persistante

Les clients utilisent`sessionPresent`dans la connexion reconnue (`CONNACK`) afin de déterminer si une session permanente est présente. Si`sessionPresent`est`1`, une session permanente est présente et tous les messages stockés pour le client sont remis au client immédiatement après que ce dernier reçoit la`CONNACK`, comme décrit dansTrafic de messages après reconnexion à une session persistante (p. 84). Si`sessionPresent`est`1`, il n'est pas nécessaire que le client se réabonne. Cependant, si`sessionPresent`est`0`, aucune session permanente n'est présente et le client doit se réabonner à ses filtres de rubrique.

Une fois que le client rejoint une session permanente, il peut publier des messages et s'abonner aux filtres de rubrique sans aucun indicateur supplémentaire sur chaque opération.

Trafic de messages après reconnexion à une session persistante

Une session permanente représente une connexion continue entre un client et un courtier de messages MQTT. Lorsqu'un client se connecte au courtier de messages à l'aide d'une session permanente, ce dernier enregistre tous les abonnements souscrits par le client pendant la connexion. Lorsque le client se déconnecte, le courtier de messages conserve les messages d'une QoS 1 et les nouveaux messages d'une QoS 1 publiés sur les rubriques auxquelles le client s'est abonné. Les messages sont stockés selon la limite de compte, les messages qui dépassent cette limite seront supprimés. Pour plus d'informations sur les limites de messages persistants, consultez. AWS IoT Core Points de terminaison et quotas. Lorsque le client se reconnecte à sa session permanente, tous les abonnements sont réactivés et tous les messages stockés sont envoyés au client à un rythme maximum de 10 messages par seconde.

Après la reconnexion, les messages stockés sont envoyés au client, à une vitesse limitée à 10 messages stockés par seconde, ainsi que tout trafic de messages en cours jusqu'à ce que le`Publish requests per second per connection`Limite est atteinte. Étant donné que le taux de remise des messages stockés est limité, il faudra plusieurs secondes pour remettre tous les messages stockés si une session a plus de 10 messages stockés à livrer après la reconnexion.

Mettre fin à une session persistante

Les conditions suivantes décrivent la fin des sessions persistantes.

- Lorsque le délai d'expiration de la session persistante s'écoule. Le minuteur d'expiration de la session permanente démarre lorsque l'agent de messages détecte la déconnexion d'un client, soit par la déconnexion du client, soit par la temporisation de la connexion.
- Lorsque le client envoie un`CONNECT`qui définit le`cleanSession`Indicateur sur`1`.

> **Note**
>
> Les messages stockés en attente d'être envoyés au client à la fin d'une session sont ignorés ; cependant, ils sont toujours facturés au tarif de messagerie standard, même s'ils n'ont pas pu être envoyés. Pour plus d'informations sur la tarification des messages, consultez. AWS IoT Core Tarification. Vous pouvez configurer l'intervalle de temps d'expiration.

Reconnexion après l'expiration d'une session persistante

Si un client ne se reconnecte pas à sa session persistante avant son expiration, la session se termine et ses messages stockés sont ignorés. Lorsqu'un client se reconnecte après l'expiration de la session avec un`cleanSession`Indicateur sur`0`, le service crée une nouvelle session persistante. Les abonnements ou messages de la session précédente ne sont pas disponibles pour cette session car ils ont été ignorés à l'expiration de la session précédente.

Frais de message de session persistante

Les messages sont facturés à votre Compte AWS lorsque le courtier de messages envoie un message à un client ou à une session persistante hors connexion. Lorsqu'un appareil hors connexion avec une session persistante se reconnecte et reprend sa session, les messages stockés sont remis à l'appareil et débités à nouveau sur votre compte. Pour plus d'informations sur la tarification des messages, consultez. AWS IoT Core Tarification - Messagerie.

La durée d'expiration de la session permanente par défaut d'une heure peut être augmentée à l'aide du processus d'augmentation de la limite standard. Notez que l'augmentation de la durée d'expiration de la session peut augmenter les frais de vos messages, car cette durée supplémentaire pourrait permettre de stocker davantage de messages pour l'appareil hors connexion et ces messages supplémentaires seraient facturés à votre compte au tarif de messagerie standard. La durée d'expiration de la session est approximative et une session peut durer jusqu'à 30 minutes de plus que la limite du compte ; toutefois, une session ne sera pas plus courte que la limite du compte. Pour plus d'informations sur les limites de session, consultez. AWSQuotas de service.

Utilisation de ConnectAttributs

`ConnectAttributes`vous permettent de spécifier les attributs que vous souhaitez utiliser dans votre message de connexion dans vos stratégies IAM, telles que`PersistentConnect`and`LastWill`. avec`ConnectAttributes`, vous pouvez créer des stratégies qui ne donnent pas aux appareils l'accès à de nouvelles fonctionnalités par défaut, ce qui peut être utile si un périphérique est compromis.

`connectAttributes` prend en charge les fonctions suivantes :

`PersistentConnect`

Utilisation de l'`PersistentConnect`Pour enregistrer tous les abonnements souscrits par le client pendant la connexion lorsque la connexion entre le client et le courtier est interrompue.

`LastWill`

Utilisation de l'`LastWill`Pour publier un message dans le répertoire`LastWillTopic`lorsqu'un client se déconnecte de façon inattendue.

Par défaut, votre stratégie a une connexion non persistante et aucun attribut n'est transmis pour cette connexion. Vous devez spécifier une connexion persistante dans votre stratégie IAM si vous voulez en avoir une.

Pour`ConnectAttributes`Exemples, consultezExemples de stratégies de connexion (p. 280).

AWS IoTDifférences par rapport à la spécification MQTT version 3.1.1

L'implémentation du courtier de messages est basée sur laSpécification MQTT v3.1.1, mais diffère de la spécification de ces manières :

- AWS IoTprend en charge la qualité de service (QoS) de MQTT 0 et 1 uniquement.AWS IoTne prend pas en charge la publication ou l'abonnement avec QoS niveau 2. Lorsque QoS 2 est demandé, le agent de messages n'envoie pas de PUBACK ou de SUBACK.
- Dans AWS IoT, l'abonnement à une rubrique avec QoS niveau 0 signifie qu'un message est distribué zéro fois ou plus. Un message peut être remis plusieurs fois. Les messages remis plusieurs fois peuvent être envoyés avec un ID de paquet différent. Dans ce cas, l'indicateur DUP n'est pas défini.

- Lorsqu'il répond à une demande de connexion, l'agent de messages envoie un message CONNACK. Ce message contient un indicateur précisant si la connexion reprend une session précédente.
- Avant d'envoyer des paquets de contrôle supplémentaires ou une demande de déconnexion, le client doit attendre que le message CONNACK soit reçu sur son appareil à partir duAWS IoTAgent de messages.
- Lorsqu'un client s'abonne à une rubrique, il peut y avoir un délai entre le moment où l'agent de messages envoie un SUBACK et le moment où le client commence à recevoir de nouveaux messages correspondants.
- La spécification MQTT fournit une disposition permettant à l'éditeur de demander au courtier de conserver le dernier message envoyé à une rubrique et de l'envoyer à tous les abonnés à venir de cette rubrique.AWS IoTne prend pas en charge les messages conservés. Si une demande est faite de conserver les messages, la connexion est interrompue.
- L'agent de messages utilise l'ID de client pour identifier chaque client. L'ID de client est transmis depuis le client à l'agent de messages dans le cadre de la charge utile MQTT. Deux clients possédant le même ID de client ne peuvent pas être connectés simultanément à l'agent de messages. Lorsqu'un client se connecte à l'agent de messages à l'aide d'un ID de client qu'un autre client utilise, la nouvelle connexion client est acceptée et le client connecté précédemment est déconnecté.
- À de rares occasions, l'agent de messages peut renvoyer le même message PUBLISH logique avec un ID de paquet différent.
- L'agent de messages ne garantit pas l'ordre dans lequel les messages et les ACK sont reçus.

HTTPS

Les clients peuvent publier des messages en adressant des demandes à l'API REST à l'aide des protocoles HTTP 1.0 ou 1.1. Pour connaître l'authentification et les mappages de port utilisés par les demandes HTTP, veuillez consulter Protocoles, mappages de ports et authentification (p. 80).

> Note
>
> Contrairement à MQTT, HTTPS ne prend pas en charge de valeur `clientId`. Donc, alors qu'un`clientId`est disponible lors de l'utilisation de MQTT, il n'est pas disponible lors de l'utilisation de HTTPS.

URL du message HTTPS

Les appareils et les clients publient leurs messages en effectuant des demandes POST à un point de terminaison spécifique au client et une URL spécifique à une rubrique :

```
https://IoT_data_endpoint/topics/url_encoded_topic_name?qos=1"
```

- *`IOT_Data_Endpoint`*est leAWS IoTPoint de terminaison de données de (p. 76). Vous pouvez trouver le point de terminaison dans la sectionAWS IoTSur la page de détails de l'objet ou sur le client à l'aide de l'outilAWS CLICommande de l' :

  aws iot describe-endpoint --endpoint-type iot:Data-ATS

  Le point de terminaison doit ressembler à ceci : `a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com`
- *`<url_encoded_topic_name>`* est le nom de rubrique (p. 89) complet du message envoyé.

Exemples de code de message HTTPS

Voici quelques exemples montrant la façon d'envoyer un message HTTPS àAWS IoT.

Python

```
import requests
import argparse

# define command-line parameters
parser = argparse.ArgumentParser(description="Send messages through an HTTPS
 connection.")
parser.add_argument('--endpoint', required=True, help="Your AWS IoT data custom
 endpoint, not including a port. " +
                                                       "Ex: \"abcdEXAMPLExyz-ats.iot.us-
east-1.amazonaws.com\"")
parser.add_argument('--cert', required=True, help="File path to your client
 certificate, in PEM format.")
parser.add_argument('--key', required=True, help="File path to your private key, in PEM
 format.")
parser.add_argument('--topic', required=True, default="test/topic", help="Topic to
 publish messages to.")
parser.add_argument('--message', default="Hello World!", help="Message to publish. " +
                                                       "Specify empty string to publish
 nothing.")

# parse and load command-line parameter values
args = parser.parse_args()

# create and format values for HTTPS request
publish_url = 'https://' + args.endpoint + ':8443/topics/' + args.topic + '?qos=1'
publish_msg = args.message.encode('utf-8')

# make request
publish = requests.request('POST',
            publish_url,
            data=publish_msg,
            cert=[args.cert, args.key])

# print results
print("Response status: ", str(publish.status_code))
if publish.status_code == 200:
        print("Response body:", publish.text)
```

CURL

Vous pouvez utiliser curl à partir d'un client ou d'un appareil pour envoyer un message à AWS IoT.

Pour utiliser curl pour envoyer un message à partir d'un appareil client AWS IoT

1. Vérifiez les pages ou curl Version d'.

   a. Sur votre client, exécutez cette commande à partir d'une invite de commande.

      curl --help

      Dans le texte d'aide, recherchez les options TLS. Vous devriez voir l'option `--tlsv1.2`.

   b. Si vous voyez l' option `--tlsv1.2`, continuez.

   c. Si vous ne voyez pas le kit `--tlsv1.2` ou vous obtenez un `command not found`, vous devrez peut-être mettre à jour ou installer curl sur votre client ou installer `openssl` Avant de continuer.

2. Installez les certificats sur votre client.

   Copiez les fichiers de certificat que vous avez créés lorsque vous avez enregistré votre client (objet) dans la console AWS IoT. Assurez-vous d'avoir ces trois fichiers de certificat sur votre client avant de continuer.

- Le fichier de certificat CA (*Amazon-root-CA-1.pem*dans cet exemple).
- Fichier de certificat du client (*périphérique.pem.crt*dans cet exemple).
- Fichier de clé privée du client (*private.pem.key*dans cet exemple).

3. Création de l'curl, en remplaçant les valeurs remplaçables pour celles de votre compte et de votre système.

```
curl --tlsv1.2 \
    --cacert Amazon-root-CA-1.pem \
    --cert device.pem.crt \
    --key private.pem.key \
    --request POST \
    --data "{ \"message\": \"Hello, world\" }" \
    "https://IoT_data_endpoint:8443/topics/topic?qos=1"
```

--tlsv1.2

Utilisez TLS 1.2 (SSL).

—cacert*Amazon-root-CA-1.pem*

Nom et chemin d'accès du fichier de certificat d'autorité de certification, si nécessaire, pour vérifier l'appairage.

--cert*périphérique.pem.crt*

Nom et chemin d'accès du fichier de certificat du client, si nécessaire.

—key*private.pem.key*

Nom et chemin d'accès du fichier de clé privée du client, si nécessaire.

—request POST

Type de demande HTTP (dans le cas présent, POST).

—data »*{ \ "message \ » : \ "Bonjour, monde \ »}*"

Données POST HTTP que vous souhaitez publier. Dans ce cas, il s'agit d'une chaîne JSON, avec les guillemets internes échappés à l'aide du caractère de barre oblique inverse (\).

« https ://*IOT_Data_Endpoint*:8443/Sujets/*topic*? qos=1"

URL de votre clientAWS IoTpoint de terminaison de données de périphérique, suivi du port HTTPS,`:8443`, qui est ensuite suivi du mot-clé,`/topics/`et le nom de la rubrique,`topic`, dans ce cas. Spécifiez la qualité de service en tant que paramètre de requête,`?qos=1`.

4. Ouvrez le client de test MQTT dans la boîte de dialogueAWS IoTconsole

   Suivez les instructions de la sectionAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62)et configurez la console pour vous abonner aux messages portant le nom de rubrique de*topic*utilisé dans votrecurl, ou utilisez le filtre de rubrique générique de#.

5. Testez la commande.

   Lors de la surveillance de la rubrique dans le client de test de la console AWS IoT, accédez à votre client et émettez la ligne de commande curl que vous avez créée à l'étape 3. Vous devriez voir les messages de votre client dans la console.

Rubriques MQTT

Rubriques MQTT identifierAWS IoTMessages.AWS IoTLes clients identifient les messages qu'ils publient en donnant aux messages des noms de rubrique. Les clients identifient les messages auxquels ils

souhaitent s'abonner (réception) en enregistrant un filtre de rubrique avec AWS IoT Core . L'agent de messages utilise des noms de rubrique et des filtres de rubrique pour acheminer les messages des clients publiant vers les clients abonnés.

L'agent de messages utilise des rubriques pour identifier les messages envoyés à l'aide de MQTT et envoyés à l'aide du protocole HTTP à l'URL du message HTTPS (p. 86).

Bien que AWS IoT prend en charge certaines rubriques système réservées (p. 91), la plupart des rubriques MQTT sont créées et gérées par vous, le concepteur du système. AWS IoT utilise des rubriques pour identifier les messages reçus de la part des clients de publication et sélectionner les messages à envoyer aux clients abonnés, comme décrit dans les sections suivantes. Avant de créer un espace de nom de rubrique pour votre système, passez en revue les caractéristiques des rubriques MQTT pour créer la hiérarchie des noms de rubrique qui fonctionne le mieux pour votre système IoT.

Noms de rubrique

Les noms de rubrique et les filtres de rubrique sont des chaînes codées en UTF-8. Ils peuvent représenter une hiérarchie d'informations en utilisant la barre oblique (/) pour séparer les niveaux de la hiérarchie. Par exemple, ce nom de rubrique peut faire référence à un capteur de température dans la salle 1 :

- `sensor/temperature/room1`

Dans cet exemple, il peut également y avoir d'autres types de capteur dans d'autres pièces avec des noms de rubrique tels que :

- `sensor/temperature/room2`
- `sensor/humidity/room1`
- `sensor/humidity/room2`

> **Note**
>
> Lorsque vous considérez les noms de rubrique pour les messages de votre système, gardez à l'esprit les points suivants :
>
> - Les noms de rubrique et les filtres de rubrique sont sensibles à la casse.
> - Les noms de rubrique ne doivent pas contenir d'informations personnelles identifiables.
> - Les noms de rubrique commençant par $ sont des rubriques réservées (p. 91) utilisées uniquement par AWS IoT Core .
> - AWS IoT Core ne peut pas envoyer ou recevoir de messages entre Compte AWS s ou régions.

Pour plus d'informations sur la conception de vos noms de rubrique et de votre espace de noms, consultez notre livre blancConception de rubriques MQTT pourAWS IoTCore.

Pour obtenir des exemples de la façon dont les applications peuvent publier et s'abonner à des messages, commencez parMise en route avec AWS IoT Core (p. 18)andAWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil (p. 1140).

> **Important**
>
> L'espace de noms de rubrique est limité à une extension Compte AWS Région et. Par exemple, les recettes`sensor/temp/room1`utilisé par un Compte AWS dans une région est distincte de la`sensor/temp/room1`utilisé par le mêmeAWSdans une autre Région ou utilisé par tout autre Compte AWS dans n'importe quelle région.

ARN de la rubrique

Tous les ARN de rubrique (Amazon Resource Names) ont la forme suivante :

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Par exemple,`arn:aws:iot:us-west-2:123EXAMPLE456:topic/application/topic/device/sensor`est un ARN pour le sujet `application/topic/device/sensor`.

Filtres de rubrique

Les clients abonnés enregistrent des filtres de rubrique avec l'agent de messages pour spécifier les rubriques de message que l'agent de messages doit leur envoyer. Un filtre de rubrique peut être composé d'un nom de rubrique unique pour s'abonner à un seul nom de rubrique ou il peut inclure des caractères génériques pour s'abonner à plusieurs noms de rubrique en même temps.

Les clients publiant ne peuvent pas utiliser de caractères génériques dans les noms de rubrique qu'ils publient.

Le tableau suivant répertorie les caractères génériques pouvant être utilisés dans un filtre de rubrique.

Caractères génériques de rubrique

Caractère générique	Correspondance	Remarques
#	Toutes les chaînes au niveau et au-dessous dans la hiérarchie des rubriques.	Doit être le dernier caractère du filtre de rubrique. Doit être le seul caractère dans son niveau de hiérarchie des rubriques. Peut être utilisé dans un filtre de rubrique contenant également le caractère générique +.
+	Toute chaîne du niveau qui contient le caractère.	Doit être le seul caractère dans son niveau de hiérarchie des rubriques. Peut être utilisé dans plusieurs niveaux d'un filtre de rubrique.

Utilisation de caractères génériques avec les exemples de nom de rubrique de capteur précédents :

- Un abonnement à `sensor/#` reçoit les messages publiés dans `sensor/`, `sensor/temperature`, `sensor/temperature/room1`, mais pas les messages publiés dans `Sensor`.
- Un abonnement à `sensor/+/room1` reçoit les messages publiés dans `sensor/temperature/room1` et `sensor/humidity/room1`, mais pas les messages envoyés à `sensor/temperature/room2` ou `sensor/humidity/room2`.

ARN de filtre de rubriques

Tous les ARN de filtre de rubrique (Amazon Resource Names) ont la forme suivante :

```
arn:aws:iot:aws-region:AWS-account-ID:topicfilter/TopicFilter
```

Par exemple,`arn:aws:iot:us-west-2:123EXAMPLE456:topicfilter/application/topic/#/sensor`est un ARN pour le filtre de rubriques `application/topic/#/sensor`.

Charge utile des messages MQTT

La charge utile du message envoyée dans vos messages MQTT n'est pas spécifiée parAWS IoT, sauf si la charge utile du message est pour l'un desthe section called “Rubriques réservées” (p. 91). Au contraire, vous définissez la charge utile des messages pour vos rubriques afin de mieux répondre aux besoins de votre application, dans les limites des AWS IoT Core Service Quotas pour les protocoles.

L'utilisation d'un format JSON pour votre charge utile de message active l'optionAWS IoTpour analyser vos messages et lui appliquer des requêtes SQL. Si votre application ne nécessite pas le moteur Règles pour appliquer des requêtes SQL à vos charges utiles de messages, vous pouvez utiliser n'importe quel format de données dont votre application a besoin. Pour plus d'informations sur les limitations et les caractères réservés dans un document JSON utilisé dans les requêtes SQL, consultezExtensions JSON (p. 540).

Pour plus d'informations sur la conception de vos rubriques MQTT et leurs charges utiles de messages correspondantes, consultezConception de rubriques MQTT pour AWS IoT Core .

Rubriques réservées

Les rubriques qui commencent par un signe dollar ($) sont réservés à l'utilisation par AWS IoT. Vous pouvez vous abonner à ces rubriques réservées et y publier lorsqu'elles le permettent. Toutefois, vous ne pouvez pas créer de nouvelles rubriques commençant par un signe dollar. Les opérations de publication ou d'abonnement à des rubriques réservées qui ne sont pas prises en charge peuvent entraîner la fin de la connexion.

Rubriques de modèle de ressource

Sujet	Opérations autorisées du client	Description
$aws/sitewise/asset-models/*assetModelId*/assets/*assetId*/properties/*propertyId*	S'abonner	AWS IoT SiteWise publie des notifications sur les propriétés de ressource pour cette rubrique. Pour de plus amples informations, veuillez consulterInteraction avec d'autresAWSservicesdans leAWS IoT SiteWiseGuide de l'utilisateur.

Rubriques Device Defender

Ces messages prennent en charge les tampons de réponse au format CBOR (Concise Binary Object Représentation) et JSON (JavaScript Object Notation), selon le *payload-format* de la rubrique.

payload-format	Type de données du format de réponse
CBOR	CBOR (Concise Binary Object Representation, représentation concise d'objets binaires)
json	JavaScript Object Notation (JSON)

Pour plus d'informations, consultez Envoi de métriques à partir d'appareils (p. 956).

Sujet	Opérations autorisées	Description
$aws/things/*thingName*/defender/metrics/*payload-format*	Publier	Les agents Device Defender publient des métriques dans cette rubrique. Pour plus d'informations, consultez Envoi de métriques à partir d'appareils (p. 956).
$aws/things/*thingName*/defender/metrics/*payload-format*/accepted	S'abonner	AWS IoT publie dans cette rubrique après qu'un agent Device Defender publie un message réussi dans $aws/things/*thingName*/defender/metrics/*payload-format*. Pour plus d'informations, consultez Envoi de métriques à partir d'appareils (p. 956).
$aws/things/*thingName*/defender/metrics/*payload-format*/rejected	S'abonner	AWS IoT publie dans cette rubrique après qu'un agent Device Defender publie un message infructueux dans $aws/things/*thingName*/defender/metrics/*payload-format*. Pour plus d'informations, consultez Envoi de métriques à partir d'appareils (p. 956).

Rubriques d'événement

Sujet	Opérations autorisées du client	Description
$aws/events/certificates/registered/*caCertificateId*	S'abonner	AWS IoT publie ce message lorsque AWS IoT enregistre automatiquement un certificat et lorsqu'un client présente un certificat avec l'état `PENDING_ACTIVATION`. Pour plus d'informations, consultez the section called “Configuration de la première connexion par un client pour l'enregistrement automatique” (p. 248).
$aws/events/job//*JobId*/annulé	S'abonner	AWS IoTPubliez ce message lorsqu'une tâche est annulée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job//*JobId*/cancellation_in_progress	S'abonner	AWS IoTpublie ce message lorsqu'une tâche est annulée. Pour plus d'informations,

Sujet	Opérations autorisées du client	Description
		consultez Événements Jobs (p. 1055).
$aws/events/job//*JobId*/terminé	S'abonner	AWS IoTpublie ce message lorsqu'une tâche est terminée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job//*JobId*/supprimé	S'abonner	AWS IoTpublie ce message lors de la suppression d'une tâche Pour plus d'informations, consultez.Événements Jobs (p. 1055).
$aws/events/job//*JobId*/deletion_in_progress	S'abonner	AWS IoTpublie ce message lorsqu'une tâche est en cours de suppression. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/annulé	S'abonner	AWS IoTpublie ce message lorsque l'exécution d'une tâche est annulée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/supprimé	S'abonner	AWS IoTpublie ce message lorsque l'exécution d'une tâche est supprimée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/a échoué	S'abonner	AWS IoTpublie ce message lorsqu'une exécution de tâche a échoué. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/rejetée	S'abonner	AWS IoTpublie ce message lorsqu'une exécution de tâche a été rejetée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/supprimé	S'abonner	AWS IoTpublie ce message lorsqu'une exécution de tâche a été supprimée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution//*JobId*/réussi	S'abonner	AWS IoTpublie ce message lorsqu'une exécution de tâche a réussi. Pour plus d'informations, consultez Événements Jobs (p. 1055).

Sujet	Opérations autorisées du client	Description
$aws/events/jobExecution//*JobId*/timed_out	S'abonner	AWS IoTpublie ce message lorsqu'une exécution de tâche a expiré. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/presence/connected/*clientId*	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un client MQTT avec l'ID client spécifié se connecte à AWS IoT. Pour plus d'informations, consultez Événements de connexion/déconnexion (p. 1058).
$aws/events/presence/disconnected/*clientId*	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un client MQTT avec l'ID client spécifié se déconnecte de AWS IoT. Pour plus d'informations, consultez Événements de connexion/déconnexion (p. 1058).
$aws/events/subscriptions/subscribed/*clientId*	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un client MQTT avec l'ID client spécifié s'abonne à une rubrique MQTT. Pour plus d'informations, consultez Événements d'abonnement/désabonnement (p. 1060).
$aws/events/subscriptions/unsubscribed/*clientId*	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un client MQTT avec l'ID client spécifié se désabonne d'une rubrique MQTT. Pour plus d'informations, consultez Événements d'abonnement/désabonnement (p. 1060).
$aws/events/thing/*thingName*/created	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet *thingName* est créé. Pour plus d'informations, consultez the section called “Événements de registre” (p. 1048).
$aws/events/thing/*thingName*/updated	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet *thingName* est mis à jour. Pour plus d'informations, consultez the section called “Événements de registre” (p. 1048).

Sujet	Opérations autorisées du client	Description
$aws/events/thing/*thingName*/deleted	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet *thingName* est supprimé. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingGroup/*thingGroupName*/created	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, *thingGroupName*, est créé. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingGroup/*thingGroupName*/updated	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, *thingGroupName*, est mis à jour. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingGroup/*thingGroupName*/deleted	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, *thingGroupName*, est supprimé. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingType/*thingTypeName*/created	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d'objet *thingTypeName* est créé. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingType/*thingTypeName*/updated	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d'objet *thingTypeName* est mis à jour. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/thingType/*thingTypeName*/deleted	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d'objet *thingTypeName* est supprimé. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).

Sujet	Opérations autorisées du client	Description
$aws/events/ thingTypeAssociation/ thing/*thingName*/ *thingTypeName*	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, *thingName*, est associé ou dissocié du type d'objet, *thingTypeName*. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/ thingGroupMembership/ thingGroup/ *thingGroupName*/ thing/*thingName*/added	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, *thingName*, est ajouté au groupe d'objets, *thingGroupName*. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/ thingGroupMembership/ thingGroup/ *thingGroupName*/ thing/*thingName*/removed	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, *thingName*, est supprimé du groupe d'objets, *thingGroupName*. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/ thingGroupHierarchy/ thingGroup/ *parentThingGroupName*/ childThingGroup/ *childThingGroupName*/ added	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, *childThingGroupName*, est ajouté au groupe d'objets, *parentThingGroupName*. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).
$aws/events/ thingGroupHierarchy/ thingGroup/ *parentThingGroupName*/ childThingGroup/ *childThingGroupName*/ removed	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, *childThingGroupName*, est supprimé du groupe d'objets, *parentThingGroupName*. Pour plus d'informations, consultez the section called "Événements de registre" (p. 1048).

Rubriques de mise en service d'une flotte

Ces messages prennent en charge les tampons de réponse au format CBOR (Concise Binary Object Représentation) et JSON (JavaScript Object Notation), selon le *payload-format* de la rubrique.

payload-format	Type de données du format de réponse
CBOR	CBOR (Concise Binary Object Representation, représentation concise d'objets binaires)
json	JavaScript Object Notation (JSON)

Pour plus d'informations, consultez API MQTT de mise en service des appareils (p. 751).

Sujet	Opérations autorisées du client	Description
$aws/certificates/create/*payload-format*	Publier	Pour créer un certificat à partir d'une demande de signature de certificat (CSR), publiez sur cette rubrique.
$aws/certificates/create/*payload-format*/accepted	S'abonner	AWS IoT publie dans cette rubrique après un appel réussi à $aws/certificates/create/*payload-format*.
$aws/certificates/create/*payload-format*/rejected	S'abonner	AWS IoT publie dans cette rubrique après un appel infructueux à $aws/certificates/create/*payload-format*.
$aws/certificates/create-from-csr/*payload-format*	Publier	Publie dans cette rubrique pour créer un certificat à partir d'un CSR.
$aws/certificates/create-from-csr/*payload-format*/accepted	S'abonner	AWS IoT publie dans cette rubrique après un appel réussi à $aws/certificates/create-from-csr/*payload-format*.
$aws/certificates/create-from-csr/*payload-format*/rejected	S'abonner	AWS IoT publie dans cette rubrique après un appel réussi à $aws/certificates/create-from-csr/*payload-format*.
$aws/events/presence/connected/*clientId*	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un client MQTT avec l'ID client spécifié se connecte à AWS IoT. Pour plus d'informations, consultez Événements de connexion/déconnexion (p. 1058).
$aws/provisioning-templates/*templateName*/provision/*payload-format*	Publier	Publiez dans cette rubrique pour enregistrer un objet.
$aws/provisioning-templates/*templateName*/provision/*payload-format*/accepted	S'abonner	AWS IoT publie dans cette rubrique après un appel réussi à $aws/provisioning-templates/*templateName*/provision/*payload-format*.

Sujet	Opérations autorisées du client	Description
$aws/provisioning-templates/*templateName*/provision/*payload-format*/rejected	S'abonner	AWS IoT publie dans cette rubrique après un appel infructueux à $aws/provisioning-templates/*templateName*/provision/*payload-format*.

Rubriques de tâche

Note

Les opérations du client notées commeRecevoirdans ce tableau indiquent les rubriquesAWS IoTpublie directement au client qui l'a demandé, que le client ait souscrit ou non au sujet. Les clients devraient également s'attendre à recevoir ces messages de réponse même s'ils ne s'y sont pas abonnés.
Ces messages de réponse ne passent pas par le courtier de messages et ils ne peuvent pas être abonnés par d'autres clients ou règles. Pour vous abonner à des messages liés à l'activité de travail, utilisez la`notify`and`notify-next`Rubriques.
Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).

Sujet	Opérations autorisées du client	Description
$aws/things/*thingName*/jobs/get	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande `GetPendingJobExecutions`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/get/accepted	S'abonner, recevoir	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande `GetPendingJobExecutions`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/get/rejected	S'abonner, recevoir	Les appareils s'abonnent à cette rubrique lorsqu'une demande `GetPendingJobExecutions` est rejetée. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/start-next	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande `StartNextPendingJobExecution`. Pour plus d'informations,

Sujet	Opérations autorisées du client	Description
		consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/start-next/accepted	S'abonner, recevoir	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande `StartNextPendingJobExecution`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/start-next/rejected	S'abonner, recevoir	Les appareils s'abonnent à cette rubrique lorsqu'une demande `StartNextPendingJobExecution` est rejetée. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/*jobId*/get	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande `DescribeJobExecution`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/*jobId*/get/accepted	S'abonner, recevoir	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande `DescribeJobExecution`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/*jobId*/get/rejected	S'abonner, recevoir	Les appareils s'abonnent à cette rubrique lorsqu'une demande `DescribeJobExecution` est rejetée. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).

Sujet	Opérations autorisées du client	Description
$aws/things/*thingName*/jobs/*jobId*/update	Publier	Les appareils publient un message dans cette rubrique pour envoyer une demande `UpdateJobExecution`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/things/*thingName*/jobs/*jobId*/update/accepted	S'abonner, recevoir	Les appareils s'abonnent à cette rubrique pour recevoir des réponses positives à une demande `UpdateJobExecution`. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683). Note Seul l'appareil qui publie sur $aws/things/*thingName*/jobs/*jobId*/update reçoit les messages sur cette rubrique.
$aws/things/*thingName*/jobs/*jobId*/update/rejected	S'abonner, recevoir	Les appareils s'abonnent à cette rubrique lorsqu'une demande `UpdateJobExecution` est rejetée. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683). Note Seul l'appareil qui publie sur $aws/things/*thingName*/jobs/*jobId*/update reçoit les messages sur cette rubrique.
$aws/things/*thingName*/jobs/notify	S'abonner	Les périphériques s'abonnent à cette rubrique pour recevoir des notifications lorsque l'exécution d'une tâche est ajoutée ou supprimée de la liste des exécutions en attente pour un objet. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).

Sujet	Opérations autorisées du client	Description
$aws/things/*thingName*/jobs/notify-next	S'abonner	Les périphériques s'abonnent à cette rubrique pour recevoir des notifications lorsque l'exécution de la tâche en attente suivante pour l'objet est modifiée. Pour plus d'informations, consultez API MQTT et HTTPS pour les appareils Jobs (p. 683).
$aws/events/job/*jobId*/completed	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est terminée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job/*jobId*/canceled	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est annulée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job/*jobId*/deleted	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est supprimée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job/*jobId*/cancellation_in_progress	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'annulation d'une tâche commence. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/job/*jobId*/deletion_in_progress	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque la suppression d'une tâche commence. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/succeeded	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche aboutit. Pour plus d'informations, consultez Événements Jobs (p. 1055).

Sujet	Opérations autorisées du client	Description
$aws/events/jobExecution/*jobId*/failed	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche échoue. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/rejected	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est rejetée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/canceled	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est annulée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/timed_out	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche arrive à expiration. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/removed	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est retirée. Pour plus d'informations, consultez Événements Jobs (p. 1055).
$aws/events/jobExecution/*jobId*/deleted	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est supprimée. Pour plus d'informations, consultez Événements Jobs (p. 1055).

Rubriques de règle

Sujet	Opérations autorisées du client	Description
$aws/rules/*ruleName*	Publier	Un appareil ou une application publie dans cette rubrique pour déclencher des règles directement. Pour plus d'informations, consultez Réduction des coûts de messagerie avec Basic Ingest (p. 480).

Rubriques liées au tunneling sécurisé

Sujet	Opérations autorisées du client	Description
$aws/things/*thing-name*/tunnels/notify	S'abonner	AWS IoT publie ce message pour qu'un agent IoT démarre un proxy local sur l'appareil distant. Pour plus d'informations, consultez the section called “Extrait de l'agent IoT” (p. 726).

Rubriques de shadow

Les rubriques de cette section sont utilisées par les shadows nommés et non nommés. Les rubriques utilisées par chacun d'eux ne diffèrent que par le préfixe de rubrique. Ce tableau indique le préfixe de rubrique utilisé par chaque type de shadow.

Valeur ***ShadowTopicPrefix***	Type de shadow
$aws/things/*thingName*/shadow	Shadow non nommé (classique)
$aws/things/*thingName*/shadow/name/*shadowName*	Shadow nommé

Pour créer une rubrique complète, sélectionnez la case*ShadowTopicPrefix*pour le type d'ombre auquel vous souhaitez faire référence, remplacez*thingName*et, le cas échéant,*shadowName*, avec leurs valeurs correspondantes, puis ajoutez cela au stub de rubrique comme indiqué dans le tableau suivant. N'oubliez pas que les rubriques sont sensibles à la casse.

Sujet	Opérations autorisées du client	Description
ShadowTopicPrefix/delete	Publier/s'abonner	Un appareil ou une application publie dans cette rubrique pour supprimer un shadow. Pour de plus amples informations, veuillez consulter /delete (p. 583).
ShadowTopicPrefix/delete/accepted	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'un shadow est supprimé. Pour de plus amples informations, veuillez consulter /delete/accepted (p. 584).
ShadowTopicPrefix/delete/rejected	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'une demande de suppression d'un shadow est rejetée. Pour de plus amples informations,

Sujet	Opérations autorisées du client	Description
		veuillez consulter /delete/rejected (p. 584).
ShadowTopicPrefix/get	Publier/s'abonner	Une application ou un objet publie un message vide dans cette rubrique pour obtenir un shadow. Pour plus d'informations, consultez Rubriques MQTT de Device Shadow (p. 577).
ShadowTopicPrefix/get/accepted	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'une demande de shadow aboutit. Pour de plus amples informations, veuillez consulter /get/accepted (p. 578).
ShadowTopicPrefix/get/rejected	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'une demande de shadow est rejetée. Pour de plus amples informations, veuillez consulter /get/rejected (p. 579).
ShadowTopicPrefix/update	Publier/s'abonner	Un objet ou une application publie dans cette rubrique pour mettre à jour un shadow. Pour de plus amples informations, veuillez consulter /update (p. 579).
ShadowTopicPrefix/update/accepted	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lors de la réussite d'une mise à jour dans un shadow. Pour de plus amples informations, veuillez consulter /update/accepted (p. 581).
ShadowTopicPrefix/update/rejected	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'une mise à jour d'un shadow est rejetée. Pour de plus amples informations, veuillez consulter /update/rejected (p. 583).

Sujet	Opérations autorisées du client	Description
ShadowTopicPrefix/update/delta	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'un écart est constaté entre les sections déclarées et les sections souhaitées d'un shadow. Pour de plus amples informations, veuillez consulter /update/delta (p. 580).
ShadowTopicPrefix/update/documents	S'abonner	AWS IoT publie un document d'état dans cette rubrique chaque fois qu'une mise à jour du shadow est effectuée avec succès. Pour de plus amples informations, veuillez consulter /update/documents (p. 582).

Rubriques de remise de fichiers basées sur MQT

Ces messages prennent en charge les tampons de réponse au format CBOR (Concise Binary Object Représentation) et JSON (JavaScript Object Notation), selon le *payload-format* de la rubrique.

payload-format	Type de données du format de réponse
CBOR	CBOR (Concise Binary Object Representation, représentation concise d'objets binaires)
json	JavaScript Object Notation (JSON)

Sujet	Opérations autorisées du client	Description
$aws/things/*ThingName*/streams/*StreamId*/data/*payload-format*	S'abonner	AWSLa remise de fichiers basée sur MQT est publiée dans cette rubrique si la requête « GetStream » à partir d'un périphérique est acceptée. La charge utile contient les données du flux. Pour plus d'informations, consultez UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques (p. 782).
$aws/things/*ThingName*/streams/*StreamId*/get/*payload-format*	Publier	Un appareil publie dans cette rubrique pour effectuer une requête « GetStream ». Pour plus d'informations, consultez UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques (p. 782).

Sujet	Opérations autorisées du client	Description
$aws/things/*ThingName*/streams/*StreamId*/description/*payload-format*	S'abonner	AWSLa remise de fichiers basée sur MQT est publiée dans cette rubrique si la requête « DescribeStream » à partir d'un périphérique est acceptée. La charge utile contient la description du flux. Pour plus d'informations, consultez UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques (p. 782).
$aws/things/*ThingName*/streams/*StreamId*/describe/*payload-format*	Publier	Un appareil publie dans cette rubrique pour effectuer une demande « DescribeStream ». Pour plus d'informations, consultez UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques (p. 782).
$aws/things/*ThingName*/streams/*StreamId*/rejet/*payload-format*	S'abonner	AWSLa remise de fichiers basée sur MQT est publiée dans cette rubrique si une requête « DescribeStream » ou « GetStream » d'un périphérique est rejetée. Pour plus d'informations, consultez UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques (p. 782).

ARN de rubrique réservée

Tous les ARN de rubrique réservée (Amazon Resource Names) ont la forme suivante :

```
arn:aws:iot:aws-region:AWS-account-ID:topic/Topic
```

Par exemple,`arn:aws:iot:us-west-2:123EXAMPLE456:topic/$aws/things/thingName/jobs/get/accepted`est un ARN pour le sujet réservé`$aws/things/thingName/jobs/get/accepted`.

Les points de terminaison configurables

> **Note**
>
> Cette fonctionnalité n'est actuellement pas disponible dans les régions Chine et GovCloud.

avec AWS IoT Core , vous pouvez configurer et gérer les comportements de vos points de terminaison de données à l'aide de configurations de domaine. Vous pouvez générer plusieurs AWS IoT Core et personnalisez également les points de terminaison de données à l'aide de vos propres noms de domaine entièrement qualifiés (et certificats de serveur associés) et d'autorisations. Pour plus d'informations sur les autorisations personnalisées, consultez.the section called “Authentification personnalisée” (p. 256).

AWS IoT Core utilise l'extension TLS d'indication de nom de serveur (SNI) pour appliquer des configurations de domaine. Les périphériques doivent utiliser cette extension lorsqu'ils se connectent. Ils doivent également transmettre un nom de serveur identique au nom de domaine que vous spécifiez dans la configuration de domaine.* Pour tester ce service, utilisez la version v2 de l'outilAWS IoTKits SDK pour les appareilsdans GitHub.

> **Note**
>
> Si vous créez plusieurs points de terminaison de données dans votre compte, ils partageront AWS IoT Core telles que les rubriques MQTT, les ombres de périphériques et les règles.

Vous pouvez utiliser des configurations de domaine pour simplifier certaines tâches, comme les tâches suivantes.

- Migrez des appareils vers AWS IoT.
- Prenez en charge des parcs d'appareils hétérogènes en maintenant des configurations de domaine distinctes pour des types d'appareils distincts
- Préservez l'identité de la marque (par exemple, via un nom de domaine) lors de la migration de l'infrastructure d'application vers AWS IoT.

Vous pouvez configurer un nom de domaine complet (FQSN) et le certificat de serveur associé. Vous pouvez également associer un mécanisme d'autorisation personnalisée. Pour plus d'informations, consultez Authentification personnalisée (p. 256).

> **Note**
>
> AWS IoT utilise l'extension TLS d'indication de nom de serveur (SNI) pour appliquer des configurations de domaine. Les appareils doivent utiliser cette extension lors de la connexion et transmettre un nom de serveur identique au nom de domaine spécifié dans la configuration de domaine. Pour tester ce service, utilisez la version v2 de chaque kit SDK pour appareils AWS IoT dans GitHub.

Rubriques

Création et configuration de domaines gérés par AWS

Vous créez un point de terminaison configurable sur un domaine géré par AWS à l'aide de l'API CreateDomainConfiguration. Une configuration de domaine pour un domaine géré par AWS comprend les éléments suivants :

- `domainConfigurationName`— Nom défini par l'utilisateur qui identifie la configuration du domaine.

  > **Note**
  >
  > Les noms de configuration de domaine commençant par `IoT:` sont réservés aux points de terminaison par défaut et ne peuvent pas être utilisés. En outre, cette valeur doit être unique à votre région.

- `defaultAuthorizerName`(facultatif) — Nom du mécanisme d'autorisation personnalisée à utiliser sur le point de terminaison.
- `allowAuthorizerOverride`— Valeur booléenne qui spécifie si les appareils peuvent remplacer l'agent d'autorisation par défaut en spécifiant un autre mécanisme dans l'en-tête HTTP de la demande. Cette valeur est requise si une valeur est spécifiée pour `defaultAuthorizerName`.

- `serviceType`–AWS IoTne prend actuellement en charge que le`DATA`Type de service. Lorsque vous spécifiez`DATA`,AWS IoTrenvoie un type de point de terminaison`iot:Data-ATS`. Vous ne pouvez pas créer un point de terminaison configurable `iot:Data` (VeriSign).

La commande AWS CLI suivante crée la configuration de domaine pour un point de terminaison `Data`.

```
aws iot create-domain-configuration --domain-configuration-name "myDomainConfigurationName"
 --service-type "DATA"
```

Création et configuration de domaines personnalisés

Les configurations de domaine vous permettent de spécifier un nom de domaine complet personnalisé (FQDN) pour vous connecter à AWS IoT. Les domaines personnalisés permettent de gérer vos propres certificats de serveur afin de gérer les détails, comme l'autorité de certification racine utilisée pour signer le certificat, l'algorithme de signature, les niveaux de la chaîne de certificats et le cycle de vie du certificat.

Le flux de travail pour définir une configuration de domaine avec un domaine personnalisé se compose des trois étapes suivantes.

1. Enregistrement des certificats de serveur dans AWS Certificate Manager (p. 108)
2. Création d'une configuration de domaine (p. 109)
3. Création d'enregistrements DNS (p. 110)

Enregistrement des certificats de serveur dansAWSgestionnaire de certificats

Avant de créer une configuration de domaine avec un domaine personnalisé, vous devez enregistrer votre chaîne de certificats de serveur dansAWS Certificate Manager(ACM). Vous pouvez utiliser trois types de certificats de serveur.

- Certificats publics générés par ACM (p. 109)
- Certificats externes signés par une autorité de certification publique (p. 109)
- Certificats externes signés par une autorité de certification privée (p. 109)

> **Note**
>
> AWS IoT considère qu'un certificat est signé par une autorité de certification publique s'il est inclus dans le groupe d'autorités de certification approuvées de Mozilla.

Conditions relatives aux certificats

VoirConditions préalables à l'importation de certificatspour connaître les exigences relatives à l'importation de certificats dans ACM. Outre ces exigences, AWS IoT Core ajoute les critères suivants.

- Le certificat feuille doit inclure leUtilisation de la clé étenduex509 v3 avec une valeur deServerAuth(Authentification du serveur Web TLS). Si vous demandez le certificat à ACM, cette extension est automatiquement ajoutée.
- La profondeur maximale de la chaîne de certificats est de 5 certificats.
- La taille maximale de la chaîne de certificat est de 16 Ko.

Utilisation d'un certificat pour plusieurs domaines

Si vous envisagez d'utiliser un certificat pour couvrir plusieurs sous-domaines, utilisez un domaine générique dans le champ CN (Common Name) ou SAN (Subject Alternative Names). Par exemple, utilisez

`*.iot.example.com` pour couvrir dev.iot.example.com, qa.iot.example.com et prod.iot.example.com. Chaque nom de domaine complet nécessite sa propre configuration de domaine, mais plusieurs configurations de domaine peuvent utiliser la même valeur générique. Les valeurs CN ou SAN doivent couvrir le nom de domaine complet que vous souhaitez utiliser en tant que domaine personnalisé. Cette couverture peut être une correspondance exacte ou une correspondance générique.

Les sections suivantes décrivent comment obtenir chaque type de certificat. Chaque ressource de certificat nécessite un nom de ressource Amazon (ARN) enregistré auprès d'ACM que vous utilisez lorsque vous créez votre configuration de domaine.

Certificats publics générés par ACM

Vous pouvez générer un certificat public pour votre domaine personnalisé à l'aide de l'API RequestCertificate. Lorsque vous générez un certificat de cette manière, ACM valide votre propriété du domaine personnalisé. Pour de plus amples informations, veuillez consulterDemander un certificat publicdans leAWS Certificate ManagerGuide de l'utilisateur.

Certificats externes signés par une autorité de certification publique

Si vous disposez déjà d'un certificat de serveur signé par une autorité de certification publique (une autorité de certification incluse dans le groupe d'autorités de certification approuvées de Mozilla), vous pouvez importer la chaîne de certificats directement dans ACM à l'aide de l'outilImportCertificateAPI. Pour en savoir plus sur cette tâche et sur les conditions préalables et les exigences en matière de format de certificat, consultez Importation de certificats.

Certificats externes signés par une autorité de certification privée

Si vous disposez déjà d'un certificat de serveur signé par une autorité de certification privée ou auto-signé, vous pouvez utiliser ce certificat pour créer votre configuration de domaine, mais vous devez également créer un certificat public supplémentaire dans ACM pour valider la propriété de votre domaine. Pour ce faire, enregistrez votre chaîne de certificats de serveur dans ACM à l'aide de l'outilImportCertificateAPI. Pour en savoir plus sur cette tâche et sur les conditions préalables et les exigences en matière de format de certificat, consultez Importation de certificats.

Une fois que vous avez importé votre certificat dans ACM, générez un certificat public pour votre domaine personnalisé à l'aide de l'outilRequestCertificateAPI. Lorsque vous générez un certificat de cette manière, ACM valide votre propriété du domaine personnalisé. Pour de plus amples informations, veuillez consulter Demander un certificat public. Lorsque vous créez votre configuration de domaine, utilisez ce certificat public comme certificat de validation.

Création d'une configuration de domaine

Vous créez un point de terminaison configurable sur un domaine personnalisé à l'aide de l'outilCreateDomainConfigurationAPI. Une configuration de domaine pour un domaine personnalisé se compose des éléments suivants :

- `domainConfigurationName`— Nom défini par l'utilisateur qui identifie la configuration du domaine.

  Note

  Les noms de configuration de domaine commençant par `IoT:` sont réservés aux points de terminaison par défaut et ne peuvent pas être utilisés. En outre, cette valeur doit être unique à votre région.
- `domainName`— Le nom de domaine complet que vos appareils utilisent pour se connecter àAWS IoT.

  Note

  AWS IoT tire parti de l'extension TLS d'indication de nom de serveur (SNI) pour appliquer des configurations de domaine. Les appareils doivent utiliser cette extension lors de la connexion et

transmettre un nom de serveur identique au nom de domaine spécifié dans la configuration de domaine.

- `serverCertificateArns`— ARN de la chaîne de certificats du serveur que vous avez enregistrée auprès d'ACM. AWS IoT Core Actuellement, un seul certificat de serveur.
- `validationCertificateArn`— ARN du certificat public que vous avez généré dans ACM pour valider la propriété de votre domaine personnalisé. Cet argument n'est pas requis si vous utilisez un certificat de serveur signé publiquement ou généré par ACM.
- `defaultAuthorizerName`(facultatif) — Nom du mécanisme d'autorisation personnalisée à utiliser sur le point de terminaison.
- `allowAuthorizerOverride`— Valeur booléenne qui spécifie si les appareils peuvent remplacer l'agent d'autorisation par défaut en spécifiant un autre mécanisme dans l'en-tête HTTP de la demande. Cette valeur est requise si une valeur est spécifiée pour `defaultAuthorizerName`.
- `serviceType`–AWS IoTne prend actuellement en charge que le`DATA`Type de service. Lorsque vous spécifiez`DATA`,AWS IoTrenvoie un type de point de terminaison`iot:Data-ATS`.

La commande AWS CLI suivante crée une configuration de domaine pour iot.example.com.

```
aws iot create-domain-configuration --domain-configuration-name "myDomainConfigurationName"
 --service-type "DATA"
--domain-name "iot.example.com" --server-certificate-arns serverCertARN --validation-
certificate-arn validationCertArn
```

Note

Après avoir créé votre configuration de domaine, jusqu'à 60 minutes peuvent s'écouler avant que`AWS IoT`sert vos certificats de serveur personnalisés.

Création d'enregistrements DNS

Après avoir enregistré votre chaîne de certificats de serveur et créé votre configuration de domaine, créez un enregistrement DNS afin que votre domaine personnalisé pointe vers un domaine AWS IoT. Cet enregistrement doit pointer vers un point de terminaison AWS IoT de type `iot:Data-ATS`. Vous pouvez obtenir votre point de terminaison à l'aide de l'outilDescribeEndpointAPI.

Procédez comme suit :AWS CLILa commande montre comment obtenir votre point de terminaison.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

Après avoir obtenu votre point de terminaison `iot:Data-ATS`, créez un enregistrement `CNAME` à partir de votre domaine personnalisé vers ce point de terminaison AWS IoT. Si vous créez plusieurs domaines personnalisés dans le même compte, créez un alias sur ce même point de terminaison `iot:Data-ATS`.

Troubleshooting

Si vous rencontrez des difficultés pour connecter des périphériques à un domaine personnalisé, assurez-vous que AWS IoT Core a accepté et appliqué votre certificat de serveur. Vous pouvez vérifier que AWS IoT Core a accepté votre certificat à l'aide de la méthode AWS IoT Core ou sur la console`AWS CLI`.

Pour utiliser AWS IoT Core , accédez à la consoleParamètreset sélectionnez le nom de configuration du domaine. DansDétails du certificat de serveur, vérifiez les détails de l'état et de l'état. Si le certificat n'est pas valide, remplacez-le dans ACM par un certificat qui répond auxexigences relatives au certificat (p. 108)répertoriés dans la section précédente. Si le certificat a le même ARN, AWS IoT Core sera ramasser et l'appliquer automatiquement.

Pour vérifier l'état du certificat à l'aide de l'outil`AWS CLI`, appelez l'intentionDescribeDomainConfigurationet spécifiez votre nom de configuration de domaine.

Note

Si votre certificat n'est pas valide, AWS IoT Core continuera de servir le dernier certificat valide.

Vous pouvez vérifier quel certificat est servi sur votre point de terminaison à l'aide de la commande openssl suivante.

```
openssl s_client -connect custom-domain-name:8883 -showcerts -servername
custom-domain-name
```

Gestion des configurations de domaine

Vous pouvez gérer les cycles de vie des configurations existantes à l'aide des API suivantes.

- ListDomainConfigurations
- DescribeDomainConfiguration
- UpdateDomainConfiguration
- DeleteDomainConfiguration

Affichage des configurations de domaine

Utilisez l'API ListDomainConfigurations pour renvoyer une liste paginée de toutes les configurations de domaine de votre compte. Vous pouvez voir les détails d'une configuration de domaine particulière à l'aide de l'API DescribeDomainConfiguration. Cette API prend un seul paramètre `domainConfigurationName` et renvoie les détails de la configuration spécifiée.

Mise à jour des configurations de

Pour mettre à jour l'état ou le mécanisme d'autorisation personnalisée de votre configuration de domaine, utilisez l'API UpdateDomainConfiguration. Vous pouvez définir l'état sur `ENABLED` ou sur `DISABLED`. Si vous désactivez la configuration du domaine, les appareils connectés à ce domaine reçoivent une erreur d'authentification.

Note

Actuellement, vous ne pouvez pas mettre à jour le certificat de serveur dans la configuration de votre domaine. Pour modifier le certificat d'une configuration de domaine, vous devez le supprimer, puis le recréer.

Suppression de configurations de domaine

Avant de supprimer une configuration de domaine, utilisez l'API UpdateDomainConfiguration pour définir l'état sur `DISABLED`. Vous évitez ainsi de supprimer accidentellement le point de terminaison. Après avoir désactivé la configuration du domaine, supprimez-la à l'aide de l'API DeleteDomainConfiguration .

Note

Vous devez placer les domaines gérés par AWS dans`DISABLED`Pour 7 jours, vous pouvez les supprimer. Vous pouvez placer des domaines personnalisés dans`DISABLED`, puis supprimez-les immédiatement.

Après avoir supprimé une configuration de domaine, AWS IoT Core ne dessert plus le certificat de serveur associé à ce domaine personnalisé.

Rotation des certificats dans des domaines personnalisés

Vous devrez peut-être remplacer périodiquement votre certificat de serveur par un certificat mis à jour. Le taux auquel vous effectuez cette opération dépend de la période de validité de votre certificat. Si vous

avez généré votre certificat de serveur en utilisantAWS Certificate Manager(ACM), vous pouvez configurer le certificat pour qu'il soit renouvelé automatiquement. Lorsque ACM renouvelle votre certificat, AWS IoT Core Retrait automatiquement le nouveau certificat. Vous n'avez pas à effectuer d'action supplémentaire. Si vous avez importé votre certificat de serveur à partir d'une autre source, vous pouvez le faire pivoter en le reportant sur ACM. Pour plus d'informations sur la remise en état des certificats, consultez.Réimportation d'un certificat.

Note

AWS IoT Core ne récupère les mises à jour de certificats que dans les conditions suivantes.

- Le nouveau certificat a le même ARN que l'ancien.
- Le nouveau certificat a le même algorithme de signature, nom commun ou autre nom de sujet que l'ancien.

Didacticiels AWS IoT

These tutorials can help you learn how to apply a specific aspect of AWS IoT to your solution.

Rubriques

Connect d'un appareil à AWS IoT Core en utilisant laAWS IoTKits SDK pour les appareils

Ce didacticiel explique comment connecter un appareil à AWS IoT Core afin qu'il puisse envoyer et recevoir des données depuis et versAWS IoT. Après avoir terminé ce didacticiel, votre appareil sera configuré pour se connecter à AWS IoT Core et vous comprendrez comment les appareils communiquent avecAWS IoT.

Dans le cadre de ce didacticiel, vous allez :

1. the section called “Préparez votre appareil pourAWS IoT” (p. 114)
2. the section called “Vérifiez le protocole MQTT” (p. 114)
3. the section called “Passez en revue l'exemple d'application pubsub.py Device SDK” (p. 115)
4. the section called “Connect votre appareil et communiquez avec AWS IoT Core ” (p. 121)
5. the section called “Vérifiez les résultats” (p. 126)

Ce didacticiel vous prendra environ une heure.

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- TerminéMise en route avec AWS IoT Core (p. 18)

  Dans la section de ce tutoriel où vous devezthe section called “Configurer votre appareil” (p. 39), sélectionnez lathe section called “Connect un Raspberry Pi ou un autre appareil” (p. 53)pour votre appareil et utilisez les options de langage Python pour configurer votre appareil.

  Veillez à garder ouverte la fenêtre du terminal que vous utilisez dans ce didacticiel, car vous l'utiliserez également dans ce didacticiel.
- Un périphérique qui peut exécuter leAWS IoTSDK de périphérique version 2 pour Python.

  Ce didacticiel explique comment connecter un appareil à AWS IoT Core en utilisant des exemples de code Python, qui nécessitent un périphérique relativement puissant, à mesure que vont l'IoT et les périphériques embarqués.

  Si vous travaillez avec des périphériques à ressources limitées, ces exemples de code peuvent ne pas fonctionner sur eux. Dans ce cas, vous pouvez avoir plus de succès enthe section called “Utilisation de l' AWS IoT Device SDK for Embedded C ” (p. 126)Didacticiel.

Préparez votre appareil pourAWS IoT

DansMise en route avec AWS IoT Core (p. 18), vous avez préparé votre appareil etAWSafin qu'ils puissent communiquer. Cette section passe en revue les aspects de cette préparation qui s'appliquent à toute connexion de périphérique avec AWS IoT Core .

Pour qu'un appareil se connecte à AWS IoT Core :

1. Vous devez avoir un Compte AWS .

   La procédure dansConfigurer votre Compte AWS (p. 19)Décrit comment créer un Compte AWS Si vous n'en avez pas encore.
2. Dans ce compte, vous devez disposer des éléments suivants :AWS IoTresourcesdéfini pour l'appareil dans votre Compte AWS et Région.

   La procédure dansCréerAWS IoTresources (p. 35)décrit comment créer ces ressources pour l'appareil dans votre Compte AWS et Région.
   - Acertificat de l'appareilenregistré avecAWS IoTet activé pour authentifier l'appareil.

     Le certificat est souvent créé avec et joint à unAWS IoTObjet d'objets. Alors qu'un objet chose n'est pas nécessaire pour qu'un périphérique se connecte àAWS IoT, il rendAWS IoTdisponibles pour l'appareil.
   - AStratégieattaché au certificat de périphérique qui l'autorise à se connecter à AWS IoT Core et effectuez toutes les actions que vous voulez.
3. UnConnexion Internetqui peut accéder à votre Compte AWS les points de terminaison de l'appareil.

   Les points de terminaison du périphérique sont décrits dansAWS IoTdonnées de périphérique et points de terminaison de service (p. 76)et peut être vu dans lePage des paramètres de l'AWS IoTconsole.
4. Logiciel de communicationtels que leAWS IoTLes kits SDK de périphérique fournissent. Ce didacticiel utilise . AWS IoTSDK de périphérique version 2 pour Python.

Vérifiez le protocole MQTT

Avant de parler de l'exemple d'application, il aide à comprendre le protocole MQTT. Le protocole MQTT offre certains avantages par rapport aux autres protocoles de communication réseau, tels que HTTP, ce qui en fait un choix populaire pour les appareils IoT. Cette section passe en revue les principaux aspects de MQTT qui s'appliquent à ce didacticiel. Pour plus d'informations sur la comparaison de MQTT à HTTP, consultez.Choisir un protocole pour la communication de votre appareil (p. 81).

MQTT utilise un modèle de communication publier/abonné

Le protocole MQTT utilise un modèle de communication de publication et d'abonnement avec son hôte. Ce modèle diffère du modèle de demande/réponse que HTTP utilise. Avec MQTT, les périphériques établissent une session avec l'hôte identifié par un ID client unique. Pour envoyer des données, les périphériques publient des messages identifiés par rubriques à un courtier de messages dans l'hôte. Pour recevoir des messages du courtier de messages, les appareils s'abonnent aux rubriques qu'ils recevront en envoyant des filtres de rubriques dans les demandes d'abonnement au courtier de messages.

MQTT

Le courtier de messages reçoit des messages provenant de périphériques et publie des messages sur des appareils qui y sont abonnés. avecSessions persistantes (p. 83)(sessions qui restent actives même lorsque le périphérique initiateur est déconnecté), les périphériques peuvent récupérer les messages qui ont été publiés alors qu'ils étaient déconnectés. Côté appareil, MQTT prend en charge les niveaux de qualité de service (QoS (p. 83)) qui garantissent que l'hôte reçoit les messages envoyés par le périphérique.

Passez en revue l'exemple d'application pubsub.py Device SDK

Cette section passe en revue les`pubsub.py`Exemple d'application depuis l'AWS IoTSDK de périphérique version 2 pour Pythonutilisé dans le cadre de ce didacticiel. Ici, nous allons examiner comment il se connecte à AWS IoT Core pour publier et s'abonner aux messages MQTT. La section suivante présente quelques exercices pour vous aider à explorer comment un appareil se connecte et communique avec AWS IoT Core .

La .pubsub.pyillustre ces aspects d'une connexion MQTT avec AWS IoT Core :

- Protocoles de communication (p. 115)
- Sessions permanentes (p. 118)
- Qualité de service (p. 118)
- Publication des messages (p. 119)
- Abonnement aux messages (p. 119)
- Déconnexion et reconnexion de l'appareil (p. 120)

Protocoles de communication

La .`pubsub.py`illustre une connexion MQTT à l'aide des protocoles MQTT et MQTT sur WSS.
La .AWSExécution commune (AWSCRT)fournit la prise en charge du protocole de communication de bas niveau et est incluse dans la bibliothèqueAWS IoTSDK de périphérique version 2 pour Python.

MQTT

La .`pubsub.py`Exemples d'appel`mtls_from_path`(illustré ici) dans la`mqtt_connection_builder`Pour établir une connexion avec AWS IoT Core En utilisant le protocole MQTT.`mtls_from_path`utilise les certificats X.509 et TLS v1.2 pour authentifier le périphérique. La .AWSLa bibliothèque CRT gère les détails de niveau inférieur de cette connexion.

```
mqtt_connection = mqtt_connection_builder.mtls_from_path(
    endpoint=args.endpoint,
    cert_filepath=args.cert,
    pri_key_filepath=args.key,
    ca_filepath=args.root_ca,
    client_bootstrap=client_bootstrap,
    on_connection_interrupted=on_connection_interrupted,
    on_connection_resumed=on_connection_resumed,
    client_id=args.client_id,
    clean_session=False,
    keep_alive_secs=6
)
```

`endpoint`

Vos Compte AWS point de terminaison de l'appareil IoT

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`cert_filepath`

Chemin d'accès au fichier de certificat du périphérique

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`pri_key_filepath`

Chemin d'accès au fichier de clé privée du périphérique qui a été créé avec son fichier de certificat

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`ca_filepath`

Chemin d'accès au fichier de l'autorité de certification racine Obligatoire uniquement si le serveur MQTT utilise un certificat qui ne se trouve pas déjà dans votre magasin d'approbation.

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`client_bootstrap`

Objet d'exécution commun qui gère les activités de communication de socket

Dans l'exemple d'application, cet objet est instancié juste avant l'appel à`mqtt_connection_builder.mtls_from_path`.

`on_connection_interrupted`, `on_connection_resumed`

Fonctions de rappel à appeler lorsque la connexion du périphérique est interrompue et reprise

`client_id`

L'ID qui identifie ce périphérique de manière unique dans le champ Région AWS

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`clean_session`

Que ce soit pour démarrer une nouvelle session persistante ou, si une session est présente, se reconnecter à une session existante

`keep_alive_secs`

La valeur « keep alive », en secondes, à envoyer dans la`CONNECT`de la demande. Un ping sera automatiquement envoyé à cet intervalle. Le serveur suppose que la connexion est perdue s'il ne reçoit pas de ping après 1,5 fois cette valeur.

MQTT via WSS

La `pubsub.py`Exemples d'appel`websockets_with_default_aws_signing`(illustré ici) dans la`mqtt_connection_builder`Pour établir une connexion avec AWS IoT Core utilisation du protocole MQTT sur WSS.`websockets_with_default_aws_signing`crée une connexion MQTT sur WSS en utilisantSignature V4pour authentifier le périphérique.

```
mqtt_connection = mqtt_connection_builder.websockets_with_default_aws_signing(
    endpoint=args.endpoint,
    client_bootstrap=client_bootstrap,
    region=args.signing_region,
    credentials_provider=credentials_provider,
    websocket_proxy_options=proxy_options,
    ca_filepath=args.root_ca,
    on_connection_interrupted=on_connection_interrupted,
    on_connection_resumed=on_connection_resumed,
    client_id=args.client_id,
    clean_session=False,
    keep_alive_secs=6
)
```

`endpoint`

Vos Compte AWS point de terminaison de l'appareil IoT

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`client_bootstrap`

Objet d'exécution commun qui gère les activités de communication de socket

Dans l'exemple d'application, cet objet est instancié juste avant l'appel à`mqtt_connection_builder.websockets_with_default_aws_signing`.

`region`

La .AWSsignature Région utilisée par l'authentification Signature V4. Dans`pubsub.py`, il transmet le paramètre entré dans la ligne de commande.

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`credentials_provider`

La .AWSinformations d'identification fournies à utiliser pour l'authentification

Dans l'exemple d'application, cet objet est instancié juste avant l'appel à`mqtt_connection_builder.websockets_with_default_aws_signing`.

`websocket_proxy_options`

Options de proxy HTTP, si vous utilisez un hôte proxy

Dans l'exemple d'application, cette valeur est initialisée juste avant l'appel à`mqtt_connection_builder.websockets_with_default_aws_signing`.

`ca_filepath`

Chemin d'accès au fichier de l'autorité de certification racine Obligatoire uniquement si le serveur MQTT utilise un certificat qui ne se trouve pas déjà dans votre magasin d'approbation.

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`on_connection_interrupted`, `on_connection_resumed`

Fonctions de rappel à appeler lorsque la connexion du périphérique est interrompue et reprise

`client_id`

L'ID qui identifie ce périphérique de manière unique dans le champ Région AWS .

Dans l'exemple d'application, cette valeur est transmise à partir de la ligne de commande.

`clean_session`

Que ce soit pour démarrer une nouvelle session persistante ou, si une session est présente, se reconnecter à une session existante

`keep_alive_secs`

La valeur « keep alive », en secondes, à envoyer dans la`CONNECT`de la demande. Un ping sera automatiquement envoyé à cet intervalle. Le serveur suppose que la connexion est perdue s'il ne reçoit pas de ping après 1,5 fois cette valeur.

HTTPS

Qu'en est-il du HTTPS ? AWS IoT Core prend en charge les périphériques qui publient des requêtes HTTPS. Du point de vue de la programmation, les périphériques envoient des requêtes HTTPS à AWS IoT Core comme le ferait toute autre demande. Pour obtenir un exemple de programme Python qui envoie un message HTTP à partir d'un appareil, consultez.Exemple de code HTTPS (p. 86)Utilisation de Python`requests`bibliothèque. Cet exemple envoie un message à AWS IoT Core utilisant HTTPS de telle sorte que AWS IoT Core l'interprète comme un message MQTT.

Bien que AWS IoT Core prend en charge les requêtes HTTPS provenant de périphériques, veillez à consulter les informations surChoisir un protocole pour la communication de votre appareil (p. 81)afin que vous puissiez prendre une décision éclairée sur le protocole à utiliser pour les communications de votre appareil.

Sessions permanentes

Dans l'exemple d'application, définissez le paramètre`clean_session`Paramètre à`False`indique que la connexion doit être persistante. En pratique, cela signifie que la connexion ouverte par cet appel se reconnecte à une session persistante existante, si elle existe. Sinon, il crée et se connecte à une nouvelle session persistante.

Avec une session persistante, les messages envoyés au périphérique sont stockés par le courtier de messages alors que le périphérique n'est pas connecté. Lorsqu'un périphérique se reconnecte à une session persistante, le courtier de messages envoie à l'appareil tous les messages stockés auxquels il s'est abonné.

Sans session persistante, l'appareil ne recevra pas de messages envoyés tant que l'appareil n'est pas connecté. L'option à utiliser dépend de votre application et si les messages qui se produisent alors qu'un périphérique n'est pas connecté doivent être communiqués. Pour plus d'informations, consultez Utilisation des sessions permanentes MQTT (p. 83).

Qualité de service

Lorsque l'appareil publie des messages et s'abonne à ceux-ci, la qualité de service (QoS) préférée peut être définie.AWS IoTprend en charge les niveaux QoS 0 et 1 pour les opérations de publication et d'abonnement. Pour plus d'informations sur les niveaux de QoS dansAWS IoT, voirOptions de qualité de service (QoS) MQTT (p. 83).

La .AWSLe runtime CRT pour Python définit ces constantes pour les niveaux QoS qu'il prend en charge :

Niveaux de service Python

Niveau QoS MQTT	Valeur symbolique Python utilisée par SDK	Description
QoS niveau 0	`mqtt.QoS.AT_MOST_ONCE`	Une seule tentative d'envoi du message sera faite, qu'il soit reçu ou non. Le message peut ne pas être envoyé du tout, par exemple, si le périphérique n'est pas connecté ou s'il y a une erreur réseau.
QoS niveau 1	`mqtt.QoS.AT_LEAST_ONCE`	Le message est envoyé à plusieurs reprises jusqu'à ce qu'un`PUBACK`un accusé de réception est reçu.

Dans l'exemple d'application, les demandes de publication et d'abonnement sont effectuées avec un niveau QoS de 1 (`mqtt.QoS.AT_LEAST_ONCE`).

- QoS lors de la publication

  Lorsqu'un périphérique publie un message avec QoS niveau 1, il envoie le message à plusieurs reprises jusqu'à ce qu'il reçoive un`PUBACK`Réponse du courtier de messages. Si le périphérique n'est pas connecté, le message est mis en file d'attente pour être envoyé après sa reconnexion.
- QoS sur abonnement

  Lorsqu'un périphérique s'abonne à un message de niveau QoS 1, le courtier de messages enregistre les messages auxquels le périphérique est abonné jusqu'à ce qu'ils puissent être envoyés à l'appareil. Le courtier de messages renvoie les messages jusqu'à ce qu'il reçoive un`PUBACK`réponse de l'appareil.

Publication des messages

Après avoir établi avec succès une connexion à AWS IoT Core , les appareils peuvent publier des messages. La `pubsub.py`fait cela en appelant la méthode`publish`de la méthode`mqtt_connection`Objet

```
mqtt_connection.publish(
    topic=args.topic,
    payload=message,
    qos=mqtt.QoS.AT_LEAST_ONCE
)
```

`topic`

Nom de rubrique du message qui identifie le message

Dans l'exemple d'application, cela est transmis à partir de la ligne de commande.

`payload`

La charge utile du message formatée en tant que chaîne (par exemple, un document JSON)

Dans l'exemple d'application, cela est transmis à partir de la ligne de commande.

Un document JSON est un format de charge utile commun et reconnu par d'autresAWS IoT ; toutefois, le format de données de la charge utile du message peut être tout ce que les éditeurs et les abonnés conviennent. AutreAWS IoT, cependant, ne reconnaissent que JSON et CBOR, dans certains cas, pour la plupart des opérations.

`qos`

Niveau QoS pour ce message

Abonnement aux messages

Pour recevoir des messages depuisAWS IoTet d'autres services et appareils, les appareils s'abonnent à ces messages par leur nom de rubrique. Les appareils peuvent s'abonner à des messages individuels en spécifiant unNom de rubrique (p. 89)et à un groupe de messages en spécifiant unFiltre de rubrique (p. 90), qui peut inclure des caractères génériques. La `pubsub.py`utilise le code affiché ici pour s'abonner aux messages et enregistrer les fonctions de rappel pour traiter le message après sa réception.

```
subscribe_future, packet_id = mqtt_connection.subscribe(
    topic=args.topic,
    qos=mqtt.QoS.AT_LEAST_ONCE,
    callback=on_message_received
)
subscribe_result = subscribe_future.result()
```

`topic`

rubrique à laquelle s'abonner. Il peut s'agir d'un nom de rubrique ou d'un filtre de rubrique.

Dans l'exemple d'application, cela est transmis à partir de la ligne de commande.

`qos`

Indique si le courtier de messages doit stocker ces messages lorsque le périphérique est déconnecté.

Une valeur de`mqtt.QoS.AT_LEAST_ONCE`(QoS niveau 1), nécessite la spécification d'une session persistante (`clean_session=False`) lorsque la connexion est créée.

`callback`

Fonction à appeler pour traiter le message abonné.

La `.mqtt_connection.subscribe`fonction retourne un futur et un ID de paquet. Si la demande d'abonnement a été lancée avec succès, l'ID de paquet renvoyé est supérieur à 0. Pour vous assurer que l'abonnement a été reçu et enregistré par le courtier de messages, vous devez attendre le retour du résultat de l'opération asynchrone, comme indiqué dans l'exemple de code.

La fonction de rappel

Le rappel dans le`pubsub.py`traite les messages souscrits au fur et à mesure que le périphérique les reçoit.

```
def on_message_received(topic, payload, **kwargs):
    print("Received message from topic '{}': {}".format(topic, payload))
    global received_count
    received_count += 1
    if received_count == args.count:
        received_all_event.set()
```

`topic`

Le sujet du message

Il s'agit du nom de rubrique spécifique du message reçu, même si vous vous êtes abonné à un filtre de rubrique.

`payload`

Charge utile du message

Le format de ce format est spécifique à l'application.

`kwargs`

Arguments supplémentaires possibles comme décrit dans`mqtt.Connection.subscribe`.

Dans`pubsub.py`Exemple,`on_message_received`affiche uniquement la rubrique et sa charge utile. Il compte également les messages reçus pour mettre fin au programme après que la limite est atteinte.

Votre application évaluera la rubrique et la charge utile pour déterminer les actions à effectuer.

Déconnexion et reconnexion de l'appareil

La `.pubsub.py`inclut des fonctions de rappel qui sont appelées lorsque le périphérique est déconnecté et lorsque la connexion est rétablie. Les actions que votre appareil prend sur ces événements sont spécifiques à l'application.

Lorsqu'un appareil se connecte pour la première fois, il doit s'abonner aux rubriques à recevoir. Si la session d'un appareil est présente lorsqu'il se reconnecte, ses abonnements sont restaurés et tous les messages stockés provenant de ces abonnements sont envoyés à l'appareil après sa reconnexion.

Si la session d'un appareil n'existe plus lorsqu'il se reconnecte, il doit se réabonner à ses abonnements. Les sessions persistantes ont une durée de vie limitée et peuvent expirer lorsque le périphérique est déconnecté trop longtemps.

Connect votre appareil et communiquez avec AWS IoT Core

Cette section présente quelques exercices pour vous aider à explorer différents aspects de la connexion de votre appareil à AWS IoT Core . Pour ces exercices, vous utiliserez l'outilClient de test MQTTdans leAWS IoTpour voir ce que votre appareil publie et publier des messages sur votre appareil. Ces exercices utilisent le`pubsub.py`Exemple à partir duAWS IoTSDK de périphérique version 2 pour Pythonet misez sur votre expérience avecMise en route avec AWS IoT Core (p. 18)Didacticiels de

Dans cette section, vous effectuerez les opérations suivantes :

- S'abonner aux filtres de rubrique génériques (p. 121)
- Traiter les abonnements aux filtres de rubrique (p. 123)
- Publier des messages à partir de votre appareil (p. 124)

Pour ces exercices, vous commencerez à partir du`pubsub.py`Exemple de programme.

> Note
>
> Ces exercices supposent que vous avez terminé leMise en route avec AWS IoT Core (p. 18)et utilisez la fenêtre du terminal de votre appareil à partir de ce didacticiel.

S'abonner aux filtres de rubrique génériques

Dans cet exercice, vous allez modifier la ligne de commande utilisée pour appeler`pubsub.py`pour vous abonner à un filtre de rubrique générique et traiter les messages reçus en fonction de la rubrique du message.

Procédure d'exercice

Pour cet exercice, imaginez que votre appareil contient un contrôle de la température et un contrôle de la lumière. Il utilise ces noms de rubrique pour identifier les messages les concernant.

1. Avant de démarrer l'exercice, essayez d'exécuter cette commande à partir de la boîte de dialogueMise en route avec AWS IoT Core (p. 18)sur votre appareil pour vous assurer que tout est prêt pour l'exercice.

   ```
   cd ~/aws-iot-device-sdk-python-v2/samples
   python3 pubsub.py --topic topic_1 --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/
   certs/device.pem.crt --key ~/certs/private.pem.key –endpoint your-iot-endpoint
   ```

   Vous devriez voir la même sortie que celle que vous avez vue dans leDidacticiel de démarrage (p. 57).
2. Pour cet exercice, modifiez ces paramètres de ligne de commande.

Action	Paramètre de ligne de commande	Effet
ajouter	--message ""	Configuration`pubsub.py`Pour écouter seulement
ajouter	--count 2	Terminer le programme après avoir reçu deux messages

Action	Paramètre de ligne de commande	Effet
CHANGE	--topic `device/+/details`	Définissez le filtre de rubrique auquel vous souhaitez vous abonner

Cette ligne de commande permet d'apporter ces modifications à la ligne de commande initiale. Entrez cette commande dans la fenêtre du terminal de votre appareil.

```
python3 pubsub.py --message "" --count 2 --topic device/+/details --root-ca ~/certs/
Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key –
endpoint your-iot-endpoint
```

Le programme doit afficher quelque chose comme suit :

```
Connecting to a3qexamplesffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-24d7cdcc-cc01-458c-8488-2d05849691e1'...
Connected!
Subscribing to topic 'device/+/details'...
Subscribed with QoS.AT_LEAST_ONCE
Waiting for all messages to be received...
```

Si vous voyez quelque chose comme ceci sur votre terminal, votre appareil est prêt et écoute les messages où les noms de rubrique commencent par`topic-1/`et se terminent par`/detail`. Alors, nous allons tester ça.

3. Voici quelques messages que votre appareil peut recevoir.

Nom de la rubrique	Charge utile du message
`device/temp/details`	`{ "desiredTemp": 20, "currentTemp": 15 }`
`device/light/details`	`{ "desiredLight": 100, "currentLight": 50 }`

4. utilisation du client de test MQTT dansAWS IoT, envoyez les messages décrits à l'étape précédente à votre appareil.

   a. Ouverture d'Client de test MQTTdans leAWS IoTconsole

   b. DansS'abonner à une rubrique, dans laChamp de rubriques abonnement, saisissez le filtre de rubriques :**`device/+/details`**, puis choisissezS'abonner à la rubrique.

   c. DansSubscriptionsdu client de test MQTT, choisissezpériphérique/+/détails.

   d. Pour chacune des rubriques du tableau précédent, procédez comme suit dans le client de test MQTT :

      1. DansPublier, entrez la valeur de la propriétéNom de la rubriqueColonne de la table.
      2. Dans le champ de charge utile du message sous le nom de rubrique, entrez la valeur du champ de champCharge utile du messageColonne de la table.
      3. Regardez la fenêtre du terminal où`pubsub.py`est en cours d'exécution et, dans le client de test MQTT, choisissezPublier dans la rubrique.

      Vous devriez voir que le message a été reçu par`pubsub.py`dans la fenêtre du terminal.

Résultat de l'exercice

Avec cela,`pubsub.py`, s'est abonné aux messages à l'aide d'un filtre de rubrique générique, les a reçus et les a affichés dans la fenêtre du terminal. Notez comment vous vous êtes abonné à un seul filtre de rubrique, et la fonction de rappel a été appelée pour traiter les messages ayant deux rubriques distinctes.

Traiter les abonnements aux filtres de rubrique

En se basant sur l'exercice précédent, modifiez le`pubsub.py`pour évaluer les rubriques des messages et traiter les messages abonnés en fonction de la rubrique.

Procédure d'exercice

Pour évaluer la rubrique du message

1. Copiez `pubsub.py` dans `pubsub2.py`.
2. Ouvrir`pubsub2.py`Dans votre éditeur de texte favori ou votre IDE.
3. Dans`pubsub2.py`, trouvez la`on_message_received`.
4. Dans`on_message_received`, insérez le code suivant après la ligne qui commence par`print("Received message`et avant la ligne qui commence par`global received_count`.

```
    topic_parsed = False
    if "/" in topic:
        parsed_topic = topic.split("/")
        if len(parsed_topic) == 3:
            # this topic has the correct format
            if (parsed_topic[0] == 'device') and (parsed_topic[2] == 'details'):
                # this is a topic we care about, so check the 2nd element
                if (parsed_topic[1] == 'temp'):
                    print("Received temperature request: {}".format(payload))
                    topic_parsed = True
                if (parsed_topic[1] == 'light'):
                    print("Received light request: {}".format(payload))
                    topic_parsed = True
    if not topic_parsed:
        print("Unrecognized message topic.")
```

5. Enregistrez vos modifications et exécutez le programme modifié à l'aide de cette ligne de commande.

```
python3 pubsub2.py --message "" --count 2 --topic device/+/details --root-ca ~/certs/
Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key —
endpoint your-iot-endpoint
```

6. DansAWS IoT, ouvrez la consoleClient de test MQTT.
7. DansS'abonner à une rubrique, dans laChamp de rubriques abonnement, saisissez le filtre de rubriques :**`device/+/details`**, puisS'abonner à la rubrique.
8. DansSubscriptionsdu client de test MQTT, choisissezpériphérique/+/détails.
9. Pour chacune des rubriques de ce tableau, procédez comme suit dans le client de test MQTT :

Nom de la rubrique	Charge utile du message
`device/temp/details`	`{ "desiredTemp": 20, "currentTemp": 15 }`
`device/light/details`	`{ "desiredLight": 100, "currentLight": 50 }`

1. DansPublier, entrez la valeur de la propriétéNom de la rubriqueColonne de la table.
2. Dans le champ de charge utile du message sous le nom de rubrique, entrez la valeur du champ de champCharge utile du messageColonne de la table.
3. Regardez la fenêtre du terminal où`pubsub.py`est en cours d'exécution et, dans le client de test MQTT, choisissezPublier dans la rubrique.

Vous devriez voir que le message a été reçu par`pubsub.py`dans la fenêtre du terminal.

Le résultat devrait être similaire à celui-ci dans la fenêtre de votre terminal.

```
Connecting to a3qexamplesffp-ats.iot.us-west-2.amazonaws.com with client ID 'test-
af794be0-7542-45a0-b0af-0b0ea7474517'...
Connected!
Subscribing to topic 'device/+/details'...
Subscribed with QoS.AT_LEAST_ONCE
Waiting for all messages to be received...
Received message from topic 'device/light/details': b'{ "desiredLight": 100,
 "currentLight": 50 }'
Received light request: b'{ "desiredLight": 100, "currentLight": 50 }'
Received message from topic 'device/temp/details': b'{ "desiredTemp": 20, "currentTemp":
 15 }'
Received temperature request: b'{ "desiredTemp": 20, "currentTemp": 15 }'
2 message(s) received.
Disconnecting...
Disconnected!
```

Résultat de l'exercice

Dans cet exercice, vous avez ajouté du code afin que l'application exemple reconnaisse et traite plusieurs messages dans la fonction de rappel. Avec cela, votre appareil pourrait recevoir des messages et agir sur eux.

Une autre façon pour votre appareil de recevoir et de traiter plusieurs messages serait de s'abonner à différents messages séparément et d'affecter chaque abonnement à sa propre fonction de rappel.

Publier des messages à partir de votre appareil

Vous pouvez utiliser l'exemple d'application pubsub.py pour publier des messages à partir de votre appareil. Bien qu'il publie les messages tels quels, les messages ne peuvent pas être lus en tant que documents JSON. Cet exercice modifie l'exemple d'application pour pouvoir publier des documents JSON dans la charge utile du message qui peut être lue par AWS IoT Core .

Procédure d'exercice

Dans cet exercice, le message suivant sera envoyé avec le`device/data`.

```
{
    "timestamp": 1601048303,
    "sensorId": 28,
    "sensorData": [
        {
        "sensorName": "Wind speed",
        "sensorValue": 34.2211224
        }
    ]
```

```
}
```

Pour préparer votre client de test MQTT à surveiller les messages de cet exercice

1. DansS'abonner à une rubrique, dans laChamp de rubriques abonnement, saisissez le filtre de rubriques :**`device/data`**, puisS'abonner à la rubrique.
2. DansSubscriptionsdu client de test MQTT, choisissezpériphérique/données.
3. Laissez la fenêtre du client de test MQTT ouverte pour attendre les messages de votre appareil.

Pour envoyer des documents JSON avec l'exemple d'application pubsub.py

1. Sur votre appareil, copiez`pubsub.py`sur`pubsub3.py`.
2. Modifier`pubsub3.py`pour modifier la façon dont il met en forme les messages qu'il publie.
   a. Ouvrez `pubsub3.py` dans un éditeur de texte.
   b. Recherchez cette ligne de code :

      ```
      message = "{} [{}]".format(args.message, publish_count)
      ```

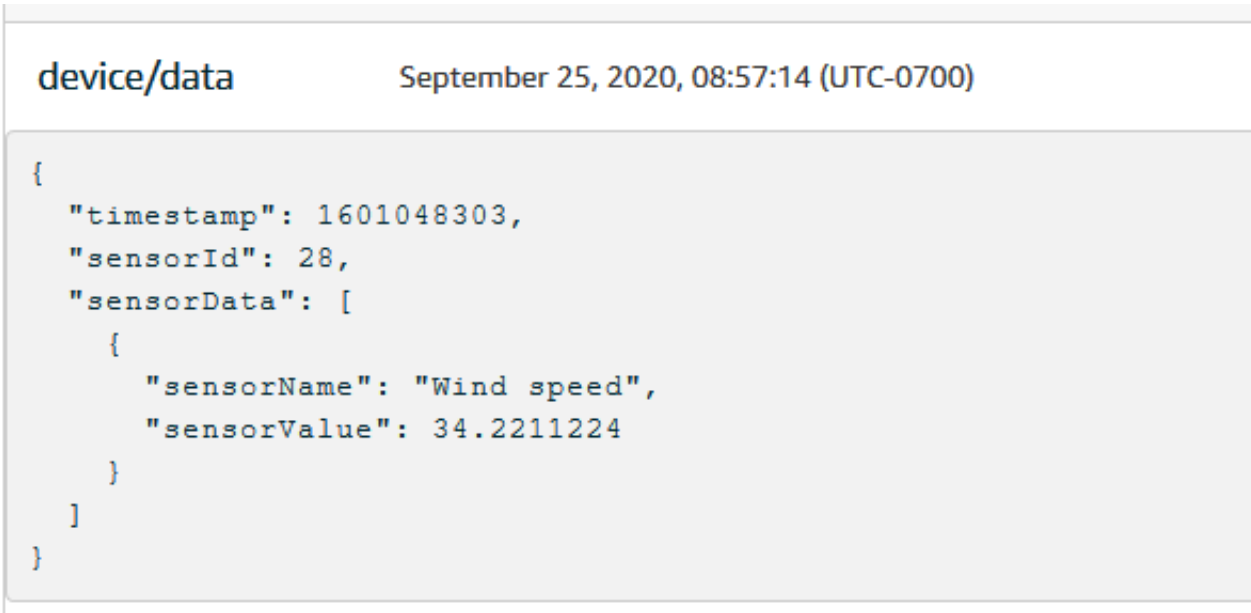
   c. Changez-le en :

      ```
      message = "{}".format(args.message)
      ```

   d. Enregistrez vos modifications.
3. Sur votre appareil, exécutez cette commande pour envoyer le message deux fois.

```
python3 pubsub3.py  --root-ca ~/certs/Amazon-root-CA-1.pem  --cert ~/certs/
device.pem.crt  --key ~/certs/private.pem.key  --topic device/data  --count 2 --
message '{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'  --endpoint your-iot-endpoint
```

4. Dans le client de test MQTT, vérifiez qu'il a interprété et formaté le document JSON dans la charge utile du message, comme ceci :



Par défaut,`pubsub3.py`s'abonne également aux messages qu'il envoie. Vous devriez voir qu'il a reçu les messages dans la sortie de l'application. La fenêtre du terminal devrait se présenter comme suit.

```
Connecting to a3qj468xinsffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-5cff18ae-1e92-4c38-a9d4-7b9771afc52f'...
Connected!
Subscribing to topic 'device/data'...
Subscribed with QoS.AT_LEAST_ONCE
Sending 2 message(s)
Publishing message to topic 'device/data':
 {"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}
Received message from topic 'device/data':
 b'{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'
Publishing message to topic 'device/data':
 {"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}
Received message from topic 'device/data':
 b'{"timestamp":1601048303,"sensorId":28,"sensorData":[{"sensorName":"Wind
 speed","sensorValue":34.2211224}]}'
2 message(s) received.
Disconnecting...
Disconnected!
```

Résultat de l'exercice

Avec cela, votre appareil peut générer des messages à envoyer à AWS IoT Core pour tester la connectivité de base et fournir des messages de périphérique pour AWS IoT Core pour traiter. Par exemple, vous pouvez utiliser cette application pour envoyer des données de test à partir de votre appareil pour testerAWS IoTActions de règle.

Vérifiez les résultats

Les exemples de ce tutoriel vous ont donné une expérience pratique sur les bases de la façon dont les appareils peuvent communiquer avec AWS IoT Core — une partie fondamentale de votreAWS IoTSolution. Lorsque vos appareils sont en mesure de communiquer avec AWS IoT Core , ils peuvent transmettre des messages àAWSservices et autres appareils sur lesquels ils peuvent agir. De même,AWSet d'autres appareils peuvent traiter des informations qui donnent lieu à des messages renvoyés à vos appareils.

Lorsque vous êtes prêt à explorer AWS IoT Core plus loin, essayez ces tutoriels :

- the section called “Envoi d'une notification Amazon SNS” (p. 139)
- the section called “Stocker les données de périphérique dans une table DynamoDB” (p. 146)
- the section called “Formater une notification à l'aide d'unAWS Lambdafonction” (p. 152)

Utilisation de l' AWS IoT Device SDK for Embedded C

Cette section décrit comment exécuter le AWS IoT Device SDK for Embedded C .

Installation de la AWS IoT Device SDK for Embedded C

Le AWS IoT Device SDK for Embedded C est généralement destiné aux appareils à ressources limitées qui nécessitent un moteur d'exécution optimisé en langage C. Vous pouvez utiliser le kit SDK sur n'importe quel système d'exploitation et l'héberger sur n'importe quel type de processeur (par exemple, microcontrôleurs et MPU). Si vous disposez de plus de ressources de mémoire et de traitement disponibles, nous vous invitons à utiliser l'un des kits SDK pour appareils et mobiles AWS IoT plus sophistiqués (par exemple, C++, Java, JavaScript et Python).

En général, le AWS IoT Device SDK for Embedded C est destiné aux systèmes qui utilisent des microcontrôleurs ou des MPU bas de gamme qui exécutent des systèmes d'exploitation embarqués. Pour l'exemple de programmation de cette section, nous supposons que votre appareil utilise Linux.

Example

1. Télécharger le AWS IoT Device SDK for Embedded C à votre appareil depuisGitHub.

   ```
   git clone https://github.com/aws/aws-iot-device-sdk-embedded-c.git --recurse-submodules
   ```

   Cette opération crée un répertoire nommé `aws-iot-device-sdk-embedded-c` dans le répertoire actuel.
2. Accédez à ce répertoire et retirez la dernière version. Veuillez consulterGithub.com/AWS/AWS-IOT-Device-SDK-Embedded-C/Tagspour la dernière balise de mise à jour.

   ```
   cd aws-iot-device-sdk-embedded-C
   git checkout latest-release-tag
   ```
3. Installez OpenSSL version 1.1.0 ou ultérieure. Les bibliothèques de développement OpenSSL sont généralement appelées « libssl-dev » ou « openssl-devel » lorsqu'elles sont installées via un gestionnaire de paquets.

   ```
   sudo apt-get install libssl-dev
   ```

Configuration de l'exemple d'application

Le AWS IoT Device SDK for Embedded C inclut des exemples d'applications que vous pouvez essayer. Pour plus de simplicité, ce didacticiel utilise le`mqtt_demo_mutual_auth`, qui illustre comment se connecter à l'objet AWS IoT Core Courtier de messages et abonnez-vous et publiez sur les rubriques MQTT.

1. Copiez le certificat et la clé privée que vous avez créés dansMise en route avec AWS IoT Core (p. 18)dans le`build/bin/certificates`répertoire.

   Note

   Les certificats d'autorité de certification racine et d'appareil sont susceptibles d'expirer ou d'être révoqués. Si ces certificats expirent ou sont révoqués, vous devez copier un nouveau certificat d'autorité de certification ou une nouvelle clé privée et un nouveau certificat d'appareil sur votre appareil.
2. Vous devez configurer l'exemple avec votre point de terminaison AWS IoT Core personnel, votre clé privée, votre certificat et votre certificat d'autorité de certification racine. Accédez au répertoire `aws-iot-device-sdk-embedded-c/demos/mqtt/mqtt_demo_mutual_auth`.

   Si vous avez installé l'AWS CLI, vous pouvez utiliser la commande aws iot describe-endpoint --endpoint-type iot:Data-ATS pour trouver l'URL de votre point de terminaison personnel. Si vous n'avez pas installé l'AWS CLI, ouvrez votre console AWS IoT. Dans le panneau de navigation, choisissez Manage (Gérer), puis Things (Objets). Choisissez l'objet IoT pour votre appareil, puis choisissezInteragir. Votre point de terminaison s'affiche dans la section HTTPS de la page de détails de l'objet.
3. Ouverture d'`demo_config.h`et mettez à jour les valeurs pour :

   AWS_IOT_ENDPOINT

   Votre point de terminaison personnel.

CLIENT_CERT_PATH

Votre chemin d'accès au fichier de certificat, par exemple`certificates/device.pem.crt`".

CLIENT_PRIVATE_KEY_PATH

Le nom de fichier de clé privée, par exemple`certificates/private.pem.key`.

Exemples :

```
// Get from demo_config.h
// ================================================
#define AWS_IOT_ENDPOINT                "my-endpoint-ats.iot.us-east-1.amazonaws.com"
#define AWS_MQTT_PORT                   8883
#define CLIENT_IDENTIFIER               "testclient"
#define ROOT_CA_CERT_PATH               "certificates/AmazonRootCA1.crt"
#define CLIENT_CERT_PATH                "certificates/my-device-cert.pem.crt"
#define CLIENT_PRIVATE_KEY_PATH         "certificates/my-device-private-key.pem.key"
// ================================================
```

4. Vérifiez si CMake est installé sur votre appareil à l'aide de cette commande.

```
cmake --version
```

Si vous voyez les informations de version du compilateur, vous pouvez passer à la section suivante.

Si vous obtenez une erreur ou que vous ne voyez aucune information, vous devrez installer le package cmake à l'aide de cette commande.

```
sudo apt-get install cmake
```

Exécutez lecmake --versionCommande à nouveau et vérifiez que CMake a été installé et que vous êtes prêt à continuer.

5. Vérifiez si les outils de développement sont installés sur votre appareil à l'aide de cette commande.

```
gcc --version
```

Si vous voyez les informations de version du compilateur, vous pouvez passer à la section suivante.

Si vous obtenez une erreur ou que vous ne voyez aucune information de compilateur, vous devrez installer le package `build-essential` à l'aide de cette commande.

```
sudo apt-get install build-essential
```

Exécutez à nouveau la commande gcc --version et vérifiez que les outils de génération ont été installés et que vous êtes prêt à continuer.

Créer et exécuter l'exemple d'application

Pour exécuter les exemples d'applications du AWS IoT Device SDK for Embedded C

1. Accédez à`aws-iot-device-sdk-embedded-c`et créez un répertoire de construction.

```
mkdir build && cd build
```

2. Entrez la commande CMake suivante pour générer les Makefiles nécessaires à la construction.

```
cmake ..
```

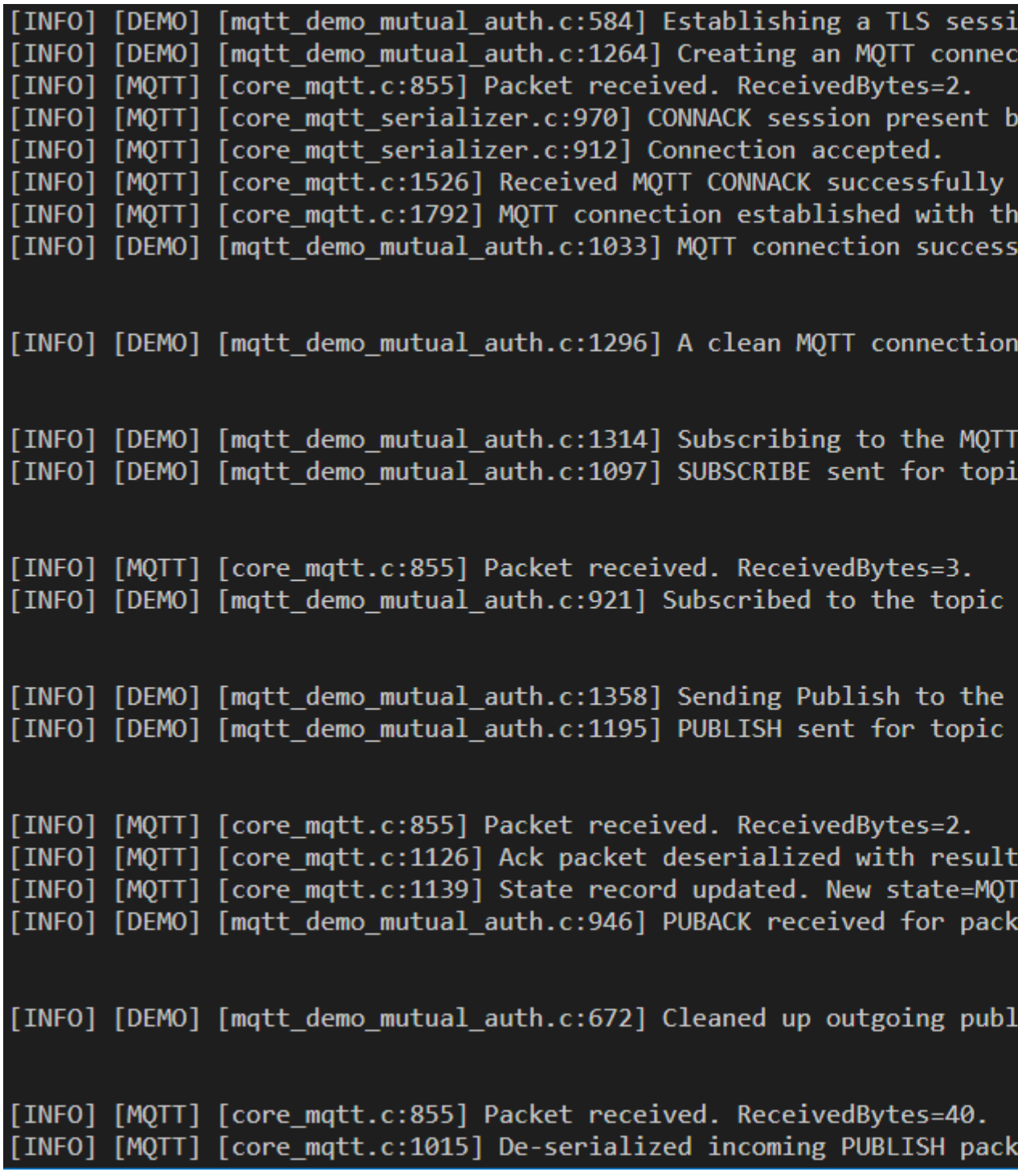
3. Entrez la commande suivante pour générer le fichier d'application exécutable.

```
make
```

4. Exécutez l'application `mqtt_demo_mutual_auth` avec cette commande.

```
cd bin
./mqtt_demo_mutual_auth
```

Vous devez voir des résultats similaires à ce qui suit :



Votre appareil est maintenant connecté àAWS IoTUtilisation du AWS IoT Device SDK for Embedded C .

Vous pouvez également utiliser l'AWS IoTPour afficher les messages MQTT que l'exemple d'application publie. Pour de plus amples informations sur l'utilisation du client MQTT dans la console AWS IoT, veuillez consulter the section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” (p. 62) .

Didacticiels concernant les règles AWS IoT

Ces didacticiels vous montrent comment créer et testerAWS IoTà l'aide de certaines des actions de règle les plus courantes.

AWS IoTenvoient des données de vos appareils à d'autresAWSServices . Ils écoutent des messages MQTT spécifiques, mettent en forme les données dans les charges utiles du message et envoient le résultat à d'autresAWSServices .

Nous vous recommandons de les essayer dans l'ordre où ils sont affichés ici, même si votre objectif est de créer une règle qui utilise une fonction Lambda ou quelque chose d'encore plus complexe. Les tutoriels sont présentés dans l'ordre du plus basique au plus complexe. Ils présentent de nouveaux concepts de manière incrémentielle pour vous aider à apprendre les concepts que vous pouvez utiliser pour créer des actions de règle qui n'ont pas de didacticiel spécifique.

> Note
>
> AWS IoTvous permettent d'envoyer les données de vos appareils IoT à d'autresAWSServices . Pour ce faire, cependant, vous aurez besoin d'une connaissance pratique des autres services sur lesquels vous souhaitez envoyer les données. Bien que ces didacticiels fournissent les informations nécessaires pour effectuer les tâches, vous pouvez trouver utile d'en savoir plus sur les services auxquels vous souhaitez envoyer des données avant de les utiliser dans votre solution. Une explication détaillée de l'autreAWSest en dehors de la portée de ces didacticiels.

Présentation du didacticiel

Le scénario de ces tutoriels est celui d'un capteur de météo qui publie périodiquement ses données. Il y a beaucoup de tels dispositifs de capteurs dans ce système imaginaire. Toutefois, les didacticiels de cette section se concentrent sur un seul périphérique tout en montrant comment vous pouvez accueillir plusieurs capteurs.

Les didacticiels de cette section vous montrent comment utiliserAWS IoTpour effectuer les tâches suivantes avec ce système imaginaire de capteurs météorologiques.

- Republication d'un message MQTT (p. 132)

  Ce didacticiel montre comment republier un message MQTT reçu des capteurs météorologiques sous la forme d'un message contenant uniquement l'ID du capteur et la valeur de température. Il utilise uniquement AWS IoT Core Vous trouverez une requête SQL simple et comment utiliser le client MQTT pour tester votre règle.
- Envoi d'une notification Amazon SNS (p. 139)

  Ce didacticiel montre comment envoyer un message SNS lorsqu'une valeur d'un capteur météorologique dépasse une valeur spécifique. Il s'appuie sur les concepts présentés dans le tutoriel précédent et ajoute comment travailler avec un autreAWSservice,Amazon Simple Notification Service(Amazon SNS).

  Si vous débutez avec Amazon SNS, consultez saMise en routeexercices avant de commencer ce tutoriel.
- Stocker les données de périphérique dans une table DynamoDB (p. 146)

  Ce didacticiel montre comment stocker les données des capteurs météorologiques dans une table de base de données. Il utilise l'instruction de requête de règle et les modèles de substitution pour formater les données de message pour le service de destination,Amazon DynamoDB.

  Si vous débutez avec DynamoDB, consultez saMise en routeexercices avant de commencer ce tutoriel.
- Formater une notification à l'aide d'unAWS Lambdafonction (p. 152)

  Ce didacticiel montre comment appeler une fonction Lambda pour reformater les données de l'appareil, puis l'envoyer sous forme de message texte. Il ajoute un script Python etAWSFonctions SDK dans

unAWS Lambdapour formater avec les données de charge utile du message provenant des capteurs météorologiques et envoyer un message texte.

Si vous débutez avec Lambda, consultez saMise en routeexercices avant de commencer ce tutoriel.

AWS IoTPrésentation de la règle

Tous ces tutoriels créentAWS IoTRègles

Pour unAWS IoTpour envoyer les données d'un périphérique à un autreAWS, il utilise :

- Instruction de requête de règle qui se compose de :
  - Clause SQL SELECT qui sélectionne et met en forme les données de la charge utile du message
  - Filtre de rubrique (objet FROM dans l'instruction de requête de règle) qui identifie les messages à utiliser
  - Instruction conditionnelle facultative (clause SQL WHERE) qui spécifie les conditions spécifiques sur lesquelles agir
- Au moins une action de règle

Les appareils publient des messages dans des rubriques MQTT. Le filtre de rubrique de l'instruction SQL SELECT identifie les rubriques MQTT auxquelles appliquer la règle. Les champs spécifiés dans l'instruction SQL SELECT mettent en forme les données de la charge utile du message MQTT entrant à utiliser par les actions de la règle. Pour obtenir la liste complète des actions de règle, consultez Actions de règle AWS IoT (p. 401).

Tutoriels dans cette section

Republication d'un message MQTT

Ce didacticiel explique comment créer unAWS IoTqui publie un message MQTT lorsqu'un message MQTT spécifié est reçu. La charge utile des messages entrants peut être modifiée par la règle avant sa publication, ce qui permet de créer des messages adaptés à des applications spécifiques sans qu'il soit nécessaire de modifier votre appareil ou son firmware. Vous pouvez également utiliser l'aspect filtrage d'une règle pour publier des messages uniquement lorsqu'une condition spécifique est remplie.

Les messages republiés par une règle agissent comme des messages envoyés par n'importe quel autreAWS IoTpériphérique ou client. Les appareils peuvent s'abonner aux messages republiés tout comme ils peuvent s'abonner à n'importe quelle autre rubrique de message MQTT.

Ce que vous apprendrez dans ce tutoriel :

- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester unAWS IoTRègle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel :

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Configurer votre Compte AWS (p. 19)

  Vous aurez besoin de votre Compte AWS andAWS IoTPour suivre ce didacticiel.
- VérificationAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62)

  Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner à une rubrique. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans cette procédure.

Examinez les rubriques MQTT etAWS IoTRègles

Avant de parler deAWS IoT, il aide à comprendre le protocole MQTT. Dans les solutions IoT, le protocole MQTT offre certains avantages par rapport aux autres protocoles de communication réseau, tels que HTTP, ce qui en fait un choix populaire pour les appareils IoT. Cette section passe en revue les principaux aspects de MQTT tels qu'ils s'appliquent à ce tutoriel. Pour plus d'informations sur la façon dont MQTT se compare à HTTP, consultezChoisir un protocole pour la communication de votre appareil (p. 81).

Protocole MQTT

Le protocole MQTT utilise un modèle de communication de publication et d'abonnement avec son hôte. Pour envoyer des données, les appareils publient des messages identifiés par des rubriques dans leAWS IoTCourtier de messages. Pour recevoir des messages du courtier de messages, les appareils s'abonnent aux rubriques qu'ils recevront en envoyant des filtres de rubriques dans les demandes d'abonnement au courtier de messages. La .AWS IoTLe moteur de règles reçoit les messages MQTT du courtier de messages.

Règles AWS IoT

AWS IoTLes règles sont constituées d'une instruction de requête de règle et d'une ou plusieurs actions de règle. Lorsque la fonctionAWS IoTreçoit un message MQTT, ces éléments agissent sur le message comme suit.

- Instruction de requête de règles

  L'instruction requête de la règle décrit les rubriques MQTT à utiliser, interprète les données de la charge utile du message et met en forme les données comme décrit par une instruction SQL similaire aux instructions utilisées par les bases de données SQL courantes. Le résultat de l'instruction de requête est les données envoyées aux actions de la règle.
- Action de la règle

  Chaque action de règle d'une règle agit sur les données qui résultent de l'instruction de requête de la règle.AWS IoTprend en chargeNombre d'actions (p. 401). Dans ce tutoriel, cependant, vous allez vous concentrer sur leRepublish (p. 433), qui publie le résultat de l'instruction de requête sous la forme d'un message MQTT avec une rubrique spécifique.

Création d'unAWS IoTrègle pour republier un message MQTT

La .AWS IoTque vous allez créer dans ce didacticiel s'abonne à la`device/`*`device_id`*`/data`Rubriques MQTT*`device_id`*ID du périphérique qui a envoyé le message. Ces rubriques sont décrites par unFiltre de rubriques (p. 90)comme`device/+/data`, où la+est un caractère générique qui correspond à n'importe quelle chaîne entre les deux caractères barre oblique.

Lorsque la règle reçoit un message à partir d'une rubrique correspondante, elle supprime le`device_id`and`temperature`et les republie sous la forme d'un nouveau message MQTT avec la propriété`device/data/temp`rubrique.

Par exemple, la charge utile d'un message MQTT avec`device/22/data`Rubrique se présente comme suit :

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

La règle prend le paramètre`temperature`à partir de la charge utile du message et de la valeur`device_id`de la rubrique et les republie sous la forme d'un message MQTT avec l'option`device/data/temp`et une charge utile de message semblable à ceci :

```
{
  "device_id": "22",
  "temperature": 28
}
```

Avec cette règle, les appareils qui n'ont besoin que de l'ID de l'appareil et des données de température s'abonnent au`device/data/temp`pour recevoir uniquement ces informations.

Pour créer une règle qui republie un message MQTT

1. Ouvrirl'opérateurRègleshub duAWS IoTconsole.
2. DansRègles, choisissezCréeret commencez à créer votre nouvelle règle.
3. Dans la partie supérieure deCréer une règle :
   a. DansNom, entrez le nom de la règle. Pour ce didacticiel, nommez-le**`republish_temp`**.

      N'oubliez pas qu'un nom de règle doit être unique au sein de votre compte et de votre région, et qu'il ne peut comporter aucun espace. Nous avons utilisé un caractère de soulignement dans ce nom pour séparer les deux mots du nom de la règle.
   b. DansDescription, décrivez la règle.

      Une description significative permet de se souvenir plus facilement de ce que fait cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, alors soyez aussi détaillée que possible.
4. DansInstruction de requête de règlesdeCréer une règle :
   a. DansUtilisation de SQL, sélectionnez**`2016-03-23`**.
   b. DansInstruction de requête de règlesDans la zone d'édition, entrez l'instruction :

      ```
      SELECT topic(2) as device_id, temperature FROM 'device/+/data'
      ```

      Cette déclaration :

      - Écoute les messages MQTT avec une rubrique qui correspond à la`device/+/data`Filtre de rubriques.

- Sélectionne le deuxième élément de la chaîne de rubrique et l'assigne à la propriété`device_id`field.
- Sélectionne la valeur`temperature`à partir de la charge utile du message et l'affecte à la`temperature`field.

5. DansDéfinissez une ou plusieurs actions :

   a. Pour ouvrir la liste des actions de règle pour cette règle, choisissezAjouter une action.
   b. DansSélectionner une action, choisissezRepubliez un message dans unAWS IoTtopic.
   c. Au bas de la liste d'actions, sélectionnezConfigurer une actionPour ouvrir la page de configuration de l'action sélectionnée.

6. DansConfigurer une action :

   a. DansRubrique, saisissez**`device/data/temp`**. Il s'agit de la rubrique MQTT du message que cette règle publiera.
   b. DansQualité de service, choisissez0 - Le message est remis zéro ou plusieurs fois.
   c. DansChoisir ou créer un rôle à attribuerAWS IoTAccès à cette action :

      i. Choisissez Create Role (Créer le rôle). La .Créer un rôles'ouvre.
      ii. Entrez un nom qui décrit le nouveau rôle. Dans ce didacticiel, utilisez**`republish_role`**.

      Lorsque vous créez un rôle, les stratégies appropriées pour exécuter l'action de règle sont créées et attachées au nouveau rôle. Si vous modifiez la rubrique de cette action de règle ou si vous utilisez ce rôle dans une autre action de règle, vous devez mettre à jour la stratégie de ce rôle pour autoriser la nouvelle rubrique ou action. Pour mettre à jour un rôle existant, choisissezMettre à jour leDans cette section.
      iii. ChoisissezCréation d'un rôlePour créer le rôle et fermer la boîte de dialogue.

   d. ChoisissezAjouter une actionpour ajouter l'action à la règle et revenir à laCréer une règle.

7. La .Republiez un message dans unAWS IoTtopicest maintenant répertoriée dansDéfinissez une ou plusieurs actions.

   Dans la vignette de la nouvelle action, sousRepubliez un message dans unAWS IoTtopic, vous pouvez voir la rubrique dans laquelle votre action de republication sera publiée.

   Il s'agit de la seule action de règle que vous allez ajouter à cette règle.

8. DansCréer une règle, faites défiler l'écran jusqu'en bas et choisissezCréer une règlepour créer la règle et terminer cette étape.

Testez votre nouvelle règle

Pour tester votre nouvelle règle, vous utiliserez le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouverture d'Client MQTTAWS IoTconsoleDans une nouvelle fenêtre. Cela vous permettra de modifier la règle sans perdre la configuration de votre client MQTT. Le client MQTT ne conserve aucun abonnements ou journaux de messages si vous le laissez accéder à une autre page de la console.

Pour utiliser le client MQTT pour tester votre règle

1. DansClient MQTTAWS IoTconsole, abonnez-vous aux rubriques d'entrée, dans ce cas,`device/+/data`.

   a. Dans le client MQTT, sousSubscriptions, choisissezS'abonner à une rubrique.
   b. Dansrubrique abonnement, entrez la rubrique du filtre de rubrique d'entrée,**`device/+/data`**.
   c. Conservez les paramètres par défaut des autres champs.

d. Choisissez Subscribe to topic (S'abonner à la rubrique).

DansSubscriptionscolumn, sousPublier dans une rubrique,**`device/+/data`**apparaît.

2. Abonnez-vous à la rubrique que votre règle publiera :`device/data/temp`.

a. UnderSubscriptions, choisissezS'abonner à une rubriqueà nouveau, et dansrubrique abonnement, entrez la rubrique du message republié,**`device/data/temp`**.

b. Conservez les paramètres par défaut des autres champs.

c. Choisissez Subscribe to topic (S'abonner à la rubrique).

DansSubscriptionscolumn, souspériphérique/+/données,**`device/data/temp`**apparaît.

3. Publier un message dans la rubrique d'entrée avec un ID de périphérique spécifique,**`device/22/data`**. Vous ne pouvez pas publier dans les rubriques MQTT qui contiennent des caractères génériques.

a. Dans le client MQTT, sousSubscriptions, choisissezPublier dans la rubrique.

b. DansPublierDans le champ, entrez le nom de la rubrique en entrée,**`device/22/data`**.

c. Copiez les exemples de données affichés ici et, dans la zone d'édition située sous le nom de la rubrique, collez les exemples de données.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

d. Pour envoyer votre message MQTT, choisissezPublier dans la rubrique.

4. Passez en revue les messages qui ont été envoyés.

a. Dans le client MQTT, sousSubscriptions, il y a un point vert à côté des deux rubriques auxquelles vous vous êtes abonné plus tôt.

Les points verts indiquent qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.

b. UnderSubscriptions, choisissezpériphérique/+/donnéespour vérifier que la charge utile du message correspond à ce que vous venez de publier et ressemble à ceci :

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

c. UnderSubscriptions, choisissezpériphérique/données/temppour vérifier que votre charge utile de message réédité ressemble à ceci :

```
{
  "device_id": "22",
  "temperature": 28
```

```
}
```

Notez que la`device_id`est une chaîne entre guillemets et la valeur`temperature`est numérique. C'est parce que le`topic()`a extrait la chaîne du nom de rubrique du message d'entrée alors que la fonction`temperature`utilise la valeur numérique de la charge utile du message d'entrée.

Si vous souhaitez que le`device_id`valeur une valeur numérique, remplacez`topic(2)`Dans l'instruction de requête de règle avec :

```
cast(topic(2) AS DECIMAL)
```

Notez que le coulage de la`topic(2)`à une valeur numérique ne fonctionnera que si cette partie de la rubrique contient uniquement des caractères numériques.

5. Si vous constatez que le message correct a été publié dans le`périphérique/données/temp`, alors votre règle a fonctionné. Pour en savoir plus sur l'action Republier la règle, reportez-vous à la section suivante.

   Si vous ne voyez pas que le message correct a été publié dans`périphérique/+/données`ou`périphérique/données/temp`, consultez les conseils de dépannage.

Dépannage de votre règle de republication des messages

Voici quelques éléments à vérifier si vous ne voyez pas les résultats que vous attendez.

- Vous avez une bannière d'erreur

  Si une erreur s'est produite lors de la publication du message d'entrée, corrigez d'abord cette erreur. Les étapes suivantes peuvent vous aider à corriger cette erreur.

- Le message d'entrée ne s'affiche pas dans le client MQTT

  Chaque fois que vous publiez votre message d'entrée dans le`device/22/data`, ce message doit apparaître dans le client MQTT si vous vous êtes abonné au`device/+/data`Filtre de rubrique comme décrit dans la procédure.

  ### Choses à vérifier

  - Vérifiez le filtre de rubrique auquel vous vous êtes abonné

    Si vous vous êtes abonné à la rubrique du message d'entrée comme décrit dans la procédure, vous devriez voir une copie du message d'entrée chaque fois que vous le publiez.

    Si le message ne s'affiche pas, vérifiez le nom de la rubrique à laquelle vous vous êtes abonné et comparez-le à la rubrique à laquelle vous avez publié. Les noms de rubrique sont sensibles à la casse et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique à laquelle vous avez publié la charge utile du message.

  - Vérifier la fonction de publication des messages

    Dans le client MQTT, sous**Subscriptions**, choisissez`périphérique/+/données`, vérifiez la rubrique du message de publication, puis choisissez**Publier dans la rubrique**. Vous devriez voir la charge utile du message dans la zone d'édition située en dessous de la rubrique s'afficher dans la liste des messages.

- Le message suivant ne s'affiche pas dans le client MQTT

  Pour que votre règle fonctionne, elle doit avoir la stratégie correcte qui l'autorise à recevoir et à republier un message et elle doit recevoir le message.

Choses à vérifier

- Vérifiez la Région AWS de votre client MQTT et la règle que vous avez créée

  La console dans laquelle vous exécutez le client MQTT doit être dans le mêmeAWSRégion comme règle que vous avez créée.
- Vérifiez la rubrique du message d'entrée dans l'instruction de requête de règle

  Pour que la règle fonctionne, elle doit recevoir un message avec le nom de rubrique correspondant au filtre de rubrique de la clause FROM de l'instruction de requête de règle.

  Vérifiez l'orthographe du filtre de rubrique dans l'instruction de requête de règle avec celle de la rubrique dans le client MQTT. Les noms de rubrique sont sensibles à la casse et la rubrique du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.
- Vérifiez le contenu de la charge utile du message d'entrée

  Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message qui est déclarée dans l'instruction SELECT.

  Vérifiez l'orthographe du`temperature`dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ sont sensibles à la casse et l'`temperature`de l'instruction de requête de règle doit être identique à l'objet`temperature`dans la charge utile du message.

  Assurez-vous que le document JSON de la charge utile du message est correctement formaté. Si le JSON a des erreurs, telles qu'une virgule manquante, la règle ne pourra pas la lire.
- Vérifier la rubrique du message republié dans l'action de la règle

  La rubrique à laquelle l'action Republier la règle publie le nouveau message doit correspondre à la rubrique à laquelle vous vous êtes abonné dans le client MQTT.

  Ouvrez la règle que vous avez créée dans la console et vérifiez la rubrique dans laquelle l'action de règle republiera le message.
- Vérifiez le rôle utilisé par la règle

  L'action de règle doit être autorisée à recevoir la rubrique d'origine et à publier la nouvelle rubrique.

  Les stratégies qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez la rubrique utilisée pour republier les données du message, vous devez mettre à jour le rôle de l'action de règle pour mettre à jour sa stratégie afin qu'elle corresponde à la rubrique actuelle.

  Si vous soupçonnez que c'est le problème, modifiez l'action Republier la règle et créez un nouveau rôle. Les nouveaux rôles créés par l'action de règle reçoivent les autorisations nécessaires pour effectuer ces actions.

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel

- Vous avez utilisé une requête SQL simple et quelques fonctions dans une instruction de requête de règle pour produire un nouveau message MQTT.
- Vous avez créé une règle qui a republié ce nouveau message.
- Vous avez utilisé le client MQTT pour tester votreAWS IoTRègle.

Étapes suivantes

Après avoir republié quelques messages avec cette règle, essayez de l'expérimenter pour voir comment la modification de certains aspects du didacticiel affecte le message republié. Voici quelques idées pour commencer.

- Remplacez par*device_id*dans la rubrique du message d'entrée et observez l'effet dans la charge utile du message republié.
- Modifiez les champs sélectionnés dans l'instruction de requête de règle et observez l'effet dans la charge utile du message republié.
- Essayez le prochain tutoriel de cette série et apprenez àEnvoi d'une notification Amazon SNS (p. 139).

L'action Republier la règle utilisée dans ce didacticiel peut également vous aider à déboguer les instructions de requête de règle. Par exemple, vous pouvez ajouter cette action à une règle pour voir comment son instruction de requête de règle met en forme les données utilisées par ses actions de règle.

Envoi d'une notification Amazon SNS

Ce didacticiel explique comment créer unAWS IoTqui envoie les données de message MQTT à une rubrique Amazon SNS afin qu'elles puissent être envoyées sous forme de message texte SMS.

Dans ce didacticiel, vous allez créer une règle qui envoie les données de message à partir d'un capteur météo à tous les abonnés d'une rubrique Amazon SNS, lorsque la température dépasse la valeur définie dans la règle. La règle détecte lorsque la température signalée dépasse la valeur définie par la règle et crée une nouvelle charge utile de message qui inclut uniquement l'ID de périphérique, la température signalée et la limite de température qui a été dépassée. La règle envoie la nouvelle charge utile de message sous la forme d'un document JSON à une rubrique SNS, qui informe tous les abonnés de la rubrique SNS.

Ce que vous apprendrez dans ce tutoriel :

- Comment créer et tester une notification Amazon SNS
- Comment appeler une notification Amazon SNS à partir d'unAWS IoTRègle
- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester unAWS IoTRègle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel :

- Créez une rubrique Amazon SNS qui envoie un message texte SMS (p. 140)
- Création d'unAWS IoTRègle pour envoyer le message texte (p. 141)
- Tester laAWS IoTRègle et notification Amazon SNS (p. 142)
- Passez en revue les résultats et les prochaines étapes (p. 146)

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Configurer votre Compte AWS (p. 19)

  Vous aurez besoin de votre Compte AWS andAWS IoTPour suivre ce didacticiel.
- VérificationAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62)

  Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner à une rubrique. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans cette procédure.
- Consultez laAmazon Simple Notification Service

Si vous n'avez jamais utilisé Amazon SNS, consultezConfiguration de l'accès à Amazon SNS. Si vous avez déjà terminé d'autresAWS IoTDidacticiels, votre Compte AWS doit déjà être configuré correctement.

Créez une rubrique Amazon SNS qui envoie un message texte SMS

Pour créer une rubrique Amazon SNS qui envoie un SMS

1. Créez une rubrique Amazon SNS.
   a. Connectez-vous à laConsole Amazon SNS.
   b. Dans le panneau de navigation de gauche, choisissez Topics (Rubriques).
   c. Dans la pageRubriquesPage, choisissezCréation d'une rubrique.
   d. DansDétails, choisissez leStandardType Par défaut, la console crée une rubrique FIFO.
   e. DansNom, entrez le nom de la rubrique SNS. Dans le cadre de ce didacticiel, entrez **`high_temp_notice`**.
   f. Faites défiler la page jusqu'en bas et choisissezCréation d'une rubrique.

      La console ouvre laDétails.
2. Créez un abonnement Amazon SNS.

   Note

   Le numéro de téléphone que vous utilisez dans cet abonnement peut entraîner des frais de messagerie texte provenant des messages que vous allez envoyer dans ce didacticiel.

   a. Danshigh_temp_noticePage des détails du sujet, choisissezCréer un abonnement.
   b. DansCréer un abonnement, dans laDétailsdans laProtocoleListe, choisissezSMS.
   c. DansPoint de terminaison, entrez le numéro d'un téléphone pouvant recevoir des messages texte. Assurez-vous de l'entrer de telle sorte qu'il commence par un+, inclut l'indicatif du pays et de la région, et n'inclut aucun autre caractère de ponctuation.
   d. Choisissez Create subscription (Créer un abonnement).
3. Testez la notification Amazon SNS.
   a. DansConsole Amazon SNSDans le volet de navigation de gauche, sélectionnezRubriques.
   b. Pour ouvrir la page de détails de la rubrique, dansRubriquesDans la liste des rubriques, sélectionnezhigh_temp_notice.
   c. Pour ouvrirPublier un message à un sujetPage, dans lahigh_temp_noticePage de détails, choisissezPublier le message.
   d. DansPublier un message à un sujet, dans laCorps du messageSection, dansCorps du message à envoyer au point de terminaison, entrez un message court.
   e. Faites défiler jusqu'au bas de la page et choisissezPublier le message.
   f. Sur le téléphone avec le numéro que vous avez utilisé précédemment lors de la création de l'abonnement, confirmez que le message a été reçu.

   Si vous n'avez pas reçu le message de test, vérifiez deux fois le numéro de téléphone et les paramètres de votre téléphone.

   Assurez-vous que vous pouvez publier des messages de test à partir duConsole Amazon SNSAvant de continuer le didacticiel.

Création d'unAWS IoTRègle pour envoyer le message texte

La .AWS IoTque vous allez créer dans ce didacticiel s'abonne à la`device/`*`device_id`*`/data`Rubriques MQTT*`device_id`*ID du périphérique qui a envoyé le message. Ces rubriques sont décrites dans un filtre de rubrique comme`device/+/data`, où la+est un caractère générique qui correspond à n'importe quelle chaîne entre les deux caractères barre oblique. Cette règle teste également la valeur de la propriété`temperature`dans la charge utile du message.

Lorsque la règle reçoit un message à partir d'une rubrique correspondante, elle prend la propriété*`device_id`*Dans le nom de la rubrique, le`temperature`de la charge utile du message, ajoute une valeur constante pour la limite de test, et envoie ces valeurs sous la forme d'un document JSON à une rubrique de notification Amazon SNS.

Par exemple, un message MQTT provenant du dispositif de capteur météo numéro 32 utilise le`device/32/data`et dispose d'une charge utile de message semblable à ceci :

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

L'instruction de requête de règle de la règle prend la propriété`temperature`de la charge utile du message, la valeur*`device_id`*à partir du nom de la rubrique, et ajoute la constante`max_temperature`pour envoyer une charge utile de message semblable à ceci à la rubrique Amazon SNS :

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30
}
```

Pour créer unAWS IoTpour détecter une valeur de température de dépassement de limite et créer les données à envoyer à la rubrique Amazon SNS :

1. Ouvrirl'opérateurRègleshub duAWS IoTconsole.
2. S'il s'agit de votre première règle, choisissezCréer, ouCréer une règle.
3. DansCréer une règle :

   a. Pour Name (Nom), entrez **`temp_limit_notify`**.

      N'oubliez pas de rappeler qu'un nom de règle doit être unique dans votre Compte AWS et Region, et il ne peut pas avoir d'espaces. Nous avons utilisé un caractère de soulignement dans ce nom pour séparer les mots dans le nom de la règle.

   b. DansDescription, décrivez la règle.

      Une description significative permet de se souvenir plus facilement de ce que fait cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, alors soyez aussi détaillée que possible.
4. DansInstruction de requête de règlesdeCréer une règle :

   a. DansUtilisation de SQL, sélectionnez2016-03-23.

b. DansInstruction de requête de règlesDans la zone d'édition, entrez l'instruction :

```
SELECT topic(2) as device_id,
    temperature as reported_temperature,
    30 as max_temperature
  FROM 'device/+/data'
  WHERE temperature > 30
```

Cette déclaration :

- Écoute les messages MQTT avec une rubrique qui correspond à la`device/+/data`et qui ont un`temperature`valeur supérieure à 30.
- Sélectionne le deuxième élément de la chaîne de rubrique et l'assigne à la propriété`device_id`field.
- Sélectionne la valeur`temperature`à partir de la charge utile du message et l'affecte à la`reported_temperature`field.
- Crée une valeur constante`30`pour représenter la valeur limite et l'assigne à la propriété`max_temperature`field.

5. Pour ouvrir la liste des actions de règle pour cette règle, dansDéfinissez une ou plusieurs actions, choisissezAjouter une action.
6. DansSélectionner une action, choisissezEnvoyer un message en tant que notification push SNS.
7. Pour ouvrir la page de configuration de l'action sélectionnée, au bas de la liste des actions, sélectionnezConfigurer une action.
8. DansConfigurer une action :

   a. DansCible SNS, choisissezTâche de sélection, trouvez votre rubrique SNS nomméehigh_temp_notice, puis choisissezTâche de sélection.
   b. DansFormat des messages, choisissezRAW.
   c. DansChoisir ou créer un rôle à attribuerAWS IoTAccès à cette action, choisissezCréation d'un rôle.
   d. DansCréer un rôle, dansNom, entrez un nom unique pour le nouveau rôle. Dans le cadre de ce didacticiel, utilisez **`sns_rule_role`**.
   e. Sélectionnez Create role (Créer un rôle).

   Si vous répétez ce didacticiel ou si vous réutilisez un rôle existant, choisissezMettre à jour leavant de continuer. Ceci met à jour le document de stratégie du rôle pour qu'il fonctionne avec la cible SNS.
9. ChoisissezAjouter une actionet revenez àCréer une règle.

   Dans la vignette de la nouvelle action, sousEnvoyer un message en tant que notification push SNS, vous pouvez voir la rubrique SNS que votre règle appellera.

   Il s'agit de la seule action de règle que vous allez ajouter à cette règle.
10. Pour créer la règle et terminer cette étape, dansCréer une règle, faites défiler l'écran jusqu'en bas et choisissezCréer une règle.

Tester laAWS IoTRègle et notification Amazon SNS

Pour tester votre nouvelle règle, vous utiliserez le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouverture d'Client MQTTAWS IoTconsoleDans une nouvelle fenêtre. Cela vous permettra de modifier la règle sans perdre la configuration de votre client MQTT. Si vous quittez le client MQTT pour accéder à une autre page de la console, il ne conservera aucun abonnements ou journaux de messages.

Pour utiliser le client MQTT pour tester votre règle

1. DansClient MQTTAWS IoTconsole, abonnez-vous aux rubriques d'entrée, dans ce cas,`device/+/data`.

   a. Dans le client MQTT, sousSubscriptions, choisissezS'abonner à une rubrique.

   b. Dansrubrique abonnement, entrez la rubrique du filtre de rubrique d'entrée,**`device/+/data`**.

   c. Conservez les paramètres par défaut des autres champs.

   d. Choisissez Subscribe to topic (S'abonner à la rubrique).

      DansSubscriptionscolumn, sousPublier dans une rubrique,**`device/+/data`**apparaît.

2. Publier un message dans la rubrique d'entrée avec un ID de périphérique spécifique,**`device/32/data`**. Vous ne pouvez pas publier dans les rubriques MQTT qui contiennent des caractères génériques.

   a. Dans le client MQTT, sousSubscriptions, choisissezPublier dans la rubrique.

   b. DansPublierDans le champ, entrez le nom de la rubrique en entrée,**`device/32/data`**.

   c. Copiez les exemples de données affichés ici et, dans la zone d'édition située sous le nom de la rubrique, collez les exemples de données.

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. ChoisissezPublier dans la rubriquepour publier votre message MQTT.

3. Confirmez que le message texte a été envoyé.

   a. Dans le client MQTT, sousSubscriptions, il y a un point vert à côté du sujet auquel vous vous êtes abonné plus tôt.

      Le point vert indique qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.

   b. UnderSubscriptions, choisissezpériphérique/+/donnéespour vérifier que la charge utile du message correspond à ce que vous venez de publier et ressemble à ceci :

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   c. Vérifiez le téléphone que vous avez utilisé pour vous abonner à la rubrique SNS et confirmez que le contenu de la charge utile du message ressemble à ceci :

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

Notez que la`device_id`est une chaîne entre guillemets et la valeur`temperature`est numérique. C'est parce que le`topic()`a extrait la chaîne du nom de rubrique du message d'entrée alors que la fonction`temperature`utilise la valeur numérique de la charge utile du message d'entrée.

Si vous souhaitez que le`device_id`valeur une valeur numérique, remplacez`topic(2)`Dans l'instruction de requête de règle avec :

```
cast(topic(2) AS DECIMAL)
```

Notez que le coulage de la`topic(2)`à une valeur numérique,`DECIMAL`ne fonctionnera que si cette partie de la rubrique ne contient que des caractères numériques.

4. Essayez d'envoyer un message MQTT dans lequel la température ne dépasse pas la limite.

   a. Dans le client MQTT, sousSubscriptions, choisissezPublier dans la rubrique.

   b. DansPublierDans le champ, entrez le nom de la rubrique en entrée,**`device/33/data`**.

   c. Copiez les exemples de données affichés ici et, dans la zone d'édition située sous le nom de la rubrique, collez les exemples de données.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

   d. Pour envoyer votre message MQTT, choisissezPublier dans la rubrique.

Vous devriez voir le message que vous avez envoyé dans la liste**`device/+/data`**, cependant, étant donné que la valeur de température est inférieure à la température maximale dans l'instruction de requête de règle, vous ne devriez pas recevoir de message texte.

Si vous ne voyez pas le comportement correct, consultez les conseils de dépannage.

Dépannage de votre règle de message SNS

Voici quelques éléments à vérifier, au cas où vous ne verrez pas les résultats que vous attendez.

- Vous avez une bannière d'erreur

  Si une erreur s'est produite lors de la publication du message d'entrée, corrigez d'abord cette erreur. Les étapes suivantes peuvent vous aider à corriger cette erreur.

- Le message d'entrée ne s'affiche pas dans le client MQTT

  Chaque fois que vous publiez votre message d'entrée dans le`device/22/data`, ce message doit apparaître dans le client MQTT, si vous vous êtes abonné au`device/+/data`Filtre de rubrique comme décrit dans la procédure.

  ### Choses à vérifier

  - Vérifiez le filtre de rubrique auquel vous vous êtes abonné

    Si vous vous êtes abonné à la rubrique du message d'entrée comme décrit dans la procédure, vous devriez voir une copie du message d'entrée chaque fois que vous le publiez.

Si le message ne s'affiche pas, vérifiez le nom de la rubrique à laquelle vous vous êtes abonné et comparez-le à la rubrique à laquelle vous avez publié. Les noms de rubrique sont sensibles à la casse et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique à laquelle vous avez publié la charge utile du message.

- Vérifier la fonction de publication des messages

  Dans le client MQTT, sousSubscriptions, choisissezpériphérique/+/données, vérifiez la rubrique du message de publication, puis choisissezPublier dans la rubrique. Vous devriez voir la charge utile du message dans la zone d'édition située en dessous de la rubrique s'afficher dans la liste des messages.

- Vous ne recevez pas de message SMS

  Pour que votre règle fonctionne, elle doit avoir la stratégie correcte qui l'autorise à recevoir un message et à envoyer une notification SNS, et elle doit recevoir le message.

  Choses à vérifier

  - Vérifiez la Région AWS de votre client MQTT et la règle que vous avez créée

    La console dans laquelle vous exécutez le client MQTT doit être dans le mêmeAWSRégion comme règle que vous avez créée.

  - Vérifier que la valeur de température dans la charge utile du message dépasse le seuil de test

    Si la valeur de température est inférieure ou égale à 30, telle que définie dans l'instruction de requête de règle, la règle n'effectuera aucune de ses actions.

  - Vérifiez la rubrique du message d'entrée dans l'instruction de requête de règle

    Pour que la règle fonctionne, elle doit recevoir un message avec le nom de rubrique correspondant au filtre de rubrique de la clause FROM de l'instruction de requête de règle.

    Vérifiez l'orthographe du filtre de rubrique dans l'instruction de requête de règle avec celle de la rubrique dans le client MQTT. Les noms de rubrique sont sensibles à la casse et la rubrique du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.

  - Vérifiez le contenu de la charge utile du message d'entrée

    Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message qui est déclarée dans l'instruction SELECT.

    Vérifiez l'orthographe du`temperature`dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ sont sensibles à la casse et l'`temperature`de l'instruction de requête de règle doit être identique à l'objet`temperature`dans la charge utile du message.

    Assurez-vous que le document JSON de la charge utile du message est correctement formaté. Si le JSON a des erreurs, telles qu'une virgule manquante, la règle ne pourra pas la lire.

  - Vérifier la rubrique du message republié dans l'action de la règle

    La rubrique à laquelle l'action Republier la règle publie le nouveau message doit correspondre à la rubrique à laquelle vous vous êtes abonné dans le client MQTT.

    Ouvrez la règle que vous avez créée dans la console et vérifiez la rubrique dans laquelle l'action de règle republiera le message.

  - Vérifiez le rôle utilisé par la règle

    L'action de règle doit être autorisée à recevoir la rubrique d'origine et à publier la nouvelle rubrique.

    Les stratégies qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez la rubrique utilisée pour republier les données

du message, vous devez mettre à jour le rôle de l'action de règle pour mettre à jour sa stratégie afin qu'elle corresponde à la rubrique actuelle.

Si vous soupçonnez que c'est le problème, modifiez l'action Republier la règle et créez un nouveau rôle. Les nouveaux rôles créés par l'action de règle reçoivent les autorisations nécessaires pour effectuer ces actions.

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel :

- Vous avez créé et testé une rubrique de notification Amazon SNS et un abonnement.
- Vous avez utilisé une requête SQL simple et des fonctions dans une instruction de requête de règle pour créer un nouveau message pour votre notification.
- Vous avez créé unAWS IoTpour envoyer une notification Amazon SNS qui utilisait votre charge utile de message personnalisée.
- Vous avez utilisé le client MQTT pour tester votreAWS IoTRègle.

Étapes suivantes

Après avoir envoyé quelques messages texte avec cette règle, essayez de l'expérimenter pour voir comment la modification de certains aspects du didacticiel affecte le message et quand il est envoyé. Voici quelques idées pour commencer.

- Remplacez par*`device_id`*dans la rubrique du message d'entrée et observez l'effet dans le contenu du message texte.
- Modifiez les champs sélectionnés dans l'instruction de requête de règle et observez l'effet dans le contenu du message texte.
- Modifiez le test dans l'instruction de requête de règle pour tester une température minimale au lieu d'une température maximale. N'oubliez pas de modifier le nom de`max_temperature` !
- Ajoutez une action de règle de republication pour envoyer un message MQTT lorsqu'une notification SNS est envoyée.
- Essayez le prochain tutoriel de cette série et apprenez àStocker les données de périphérique dans une table DynamoDB (p. 146).

Stocker les données de périphérique dans une table DynamoDB

Ce didacticiel explique comment créer unAWS IoTqui envoie les données de message à une table DynamoDB.

Dans ce didacticiel, vous allez créer une règle qui envoie les données de message à partir d'un capteur de météo imaginaire vers une table DynamoDB. La règle met en forme les données de nombreux capteurs météorologiques afin qu'elles puissent être ajoutées à une seule table de base de données.

Ce que vous apprendrez dans ce didacticiel

- Création d'une table DynamoDB
- Comment envoyer des données de message à une table DynamoDB à partir d'unAWS IoTRègle
- Comment utiliser des modèles de substitution dans unAWS IoTRègle
- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester unAWS IoTRègle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel :

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Configurer votre Compte AWS (p. 19)

  Vous aurez besoin de votre Compte AWS andAWS IoTPour suivre ce didacticiel.
- VérificationAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62)

  Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner à une rubrique. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans cette procédure.
- Consultez laAmazon DynamoDBPrésentation de

  Si vous n'avez pas utilisé DynamoDB auparavant, consultezMise en route avec DynamoDBpour se familiariser avec les concepts de base et les opérations de DynamoDB.

Créer la table DynamoDB pour ce didacticiel

Dans ce didacticiel, vous allez créer une table DynamoDB avec les attributs suivants pour enregistrer les données des périphériques de capteurs météorologiques imaginaires :

- `sample_time`est une clé primaire et décrit l'heure à laquelle l'échantillon a été enregistré.
- `device_id`est une clé de tri et décrit le périphérique qui a fourni l'exemple
- `device_data`est les données reçues du périphérique et formatées par l'instruction de requête de règle

Pour créer la table DynamoDB pour ce didacticiel

1. Ouverture d'Console DynamoDB, puisCréer une table.
2. DansCréer une table DynamoDB :
   a. DansNom de la table, entrez le nom de la table :**`wx_data`**.
   b. DansClé primaire, dansClé de partition, saisissez**`sample_time`**, et dans la liste des options en regard du champ, choisissez**`Number`**.
   c. CheckAjouter une clé de tri.
   d. Dans le champ qui s'affiche sousAjouter une clé de tri, saisissez**`device_id`**, et dans la liste des options en regard du champ, choisissez**`Number`**.
   e. Au bas de la page, choisissezCréer.

Vous allez définir`device_data`ultérieurement, lorsque vous configurez l'action de règle DynamoDB.

Création d'unAWS IoTrègle pour envoyer des données à la table DynamoDB

Dans cette étape, vous allez utiliser l'instruction de requête de règle pour formater les données des périphériques de capteurs météorologiques imaginaires à écrire dans la table de base de données.

Un exemple de charge utile de message reçu d'un capteur météorologique ressemble à ceci :

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

Pour l'entrée de base de données, vous utiliserez l'instruction de requête de règle pour aplatir la structure de la charge utile du message de manière à ce qu'elle ressemble à ceci :

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind_velocity": 22,
  "wind_bearing": 255
}
```

Dans cette règle, vous utiliserez également quelquesModèles de substitution (p. 541). Les modèles de substitution sont des expressions qui vous permettent d'insérer des valeurs dynamiques à partir de fonctions et de données de message.

Pour créer leAWS IoTrègle pour envoyer des données à la table DynamoDB

1. Ouvrirl'opérateurRègleshub de laAWS IoTconsole.
2. Pour commencer à créer votre nouvelle règle dansRègles, choisissezCréer.
3. Dans la partie supérieure deCréer une règle :

   a. DansNom, entrez le nom de la règle,**`wx_data_ddb`**.

      N'oubliez pas de rappeler qu'un nom de règle doit être unique dans votre Compte AWS et Region, et il ne peut pas avoir d'espaces. Nous avons utilisé un caractère de soulignement dans ce nom pour séparer les deux mots du nom de la règle.

   b. DansDescription, décrivez la règle.

      Une description significative permet de se souvenir plus facilement de ce que fait cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, alors soyez aussi détaillée que possible.
4. DansInstruction de requête de règlesdeCréer une règle :

   a. DansUtilisation de SQL, sélectionnez**`2016-03-23`**.

   b. DansInstruction de requête de règlesDans la zone d'édition, entrez l'instruction :

      ```
      SELECT temperature, humidity, barometer,
        wind.velocity as wind_velocity,
        wind.bearing as wind_bearing,
      FROM 'device/+/data'
      ```

      Cette déclaration :

      - Écoute les messages MQTT avec une rubrique qui correspond à la`device/+/data`Filtre de rubriques.
      - Formate les éléments de la`wind`en tant qu'attributs individuels.

- Transmet la`temperature`,`humidity`, et`barometer`Instructions inchangées.

5. DansDéfinissez une ou plusieurs actions :

   a. Pour ouvrir la liste des actions de règle pour cette règle, choisissezAction d'ajout.

   b. DansSélectionner une action, choisissezInsérer un message dans une table DynamoDB.

   c. Pour ouvrir la page de configuration de l'action sélectionnée, au bas de la liste des actions, sélectionnezConfigurer une action.

6. DansConfigurer une action :

   a. DansNom de la table, choisissez le nom de la table DynamoDB que vous avez créée à une étape précédente :**`wx_data`**.

      La .Clé de partition,Type de clé de partition,Clé de tri, etType de clé de trisont remplis avec les valeurs de votre table DynamoDB.

   b. DansValeur de la clé de partitionsaisissez.**`${timestamp()}`**.

      Il s'agit du premierModèles de substitution (p. 541)que vous utiliserez dans cette règle. Au lieu d'utiliser une valeur de la charge utile du message, il utilisera la valeur renvoyée par la méthodetimestamp (p. 533).

   c. Dans Valeur de la clé de tri, entrez**`${cast(topic(2) AS DECIMAL)}`**.

      Il s'agit de la deuxième desModèles de substitution (p. 541)que vous utiliserez dans cette règle. Il insère la valeur du deuxième élément danstopic (p. 533), qui est l'ID du périphérique, aprèsCAST (p. 501)à une valeur DECIMAL pour correspondre au format numérique de la clé.

   d. Dans Write message data to this column (Écrire des données de message dans cette colonne), entrez **`device_data`**.

      Cette opération crée le`device_data`Dans la table DynamoDB.

   e. Laissez le champ Operation (Opération) vide.

   f. DansChoisir ou créer un rôle à attribuerAWS IoTAccès à cette action, choisissezCréation d'un rôle.

   g. DansCréer un rôle, saisissez**`wx_ddb_role`**, puis choisissezCréation d'un rôle.

   h. Au bas deConfigurer une action, choisissezAction d'ajout.

   i. Pour créer la règle, en bas de l'ongletCréer une règle, choisissezCréer une règle.

Tester laAWS IoTRègle et table DynamoDB

Pour tester la nouvelle règle, vous allez utiliser le client MQTT pour publier et vous abonner aux messages MQTT utilisés dans ce test.

Ouverture d'Client MQTTAWS IoTconsoleDans une nouvelle fenêtre. Cela vous permettra de modifier la règle sans perdre la configuration de votre client MQTT. Le client MQTT ne conserve aucun abonnements ou journaux de messages si vous le laissez accéder à une autre page de la console. Vous voudrez également ouvrir une fenêtre de console distincte à la fenêtreHub de tables DynamoDB dans leAWS IoTconsolepour afficher les nouvelles entrées envoyées par votre règle.

Pour utiliser le client MQTT pour tester votre règle

1. DansClient MQTTAWS IoTconsole, abonnez-vous à la rubrique d'entrée,`device/+/data`.

   a. Dans le client MQTT, choisissezS'abonner à une rubrique.

   b. PourFiltre de rubriques, entrez la rubrique du filtre de rubrique d'entrée,**`device/+/data`**.

   c. Choisissez Subscribe.

2. Maintenant, publiez un message dans la rubrique d'entrée avec un ID de périphérique spécifique,**`device/22/data`**. Vous ne pouvez pas publier dans les rubriques MQTT qui contiennent des caractères génériques.

   a. Dans le client MQTT, choisissezPublier dans une rubrique.

   b. PourNom de la rubrique, entrez le nom de la rubrique en entrée,**`device/22/data`**.

   c. PourCharge utile des messages, entrez les exemples de données suivants.

   ```
   {
     "temperature": 28,
     "humidity": 80,
     "barometer": 1013,
     "wind": {
       "velocity": 22,
       "bearing": 255
     }
   }
   ```

   d. Pour publier le message MQTT, choisissezPublier.

   e. Maintenant, dans le client MQTT, choisissezS'abonner à une rubrique. DansS'abonner, choisissez la colonne**`device/+/data`**Abonnement. Confirmez que les exemples de données de l'étape précédente s'affichent là.

3. Vérifiez la ligne de la table DynamoDB créée par votre règle.

   a. DansHub de tables DynamoDB dans leAWS IoTconsole, choisissezwx_data, puisElémentsOnglet.

   Si vous êtes déjà sur leElémentsVous devrez peut-être actualiser l'affichage en sélectionnant l'icône d'actualisation dans le coin supérieur droit de l'en-tête du tableau.

   b. Notez que lasample_timedans la table sont des liens et ouvrent un. Si vous venez d'envoyer votre premier message, ce sera le seul dans la liste.

   Ce lien affiche toutes les données de cette ligne du tableau.

   c. Développez ladevice_datapour afficher les données résultant de l'instruction de requête de règle.

   d. Explorez les différentes représentations des données disponibles dans cet affichage. Vous pouvez également modifier les données dans cet affichage.

   e. Une fois que vous avez terminé la révision de cette ligne de données, pour enregistrer les modifications que vous avez apportées, sélectionnezEnregistrer, ou pour quitter sans enregistrer de modifications, choisissezAnnuler.

Si vous ne voyez pas le comportement correct, consultez les conseils de dépannage.

Résolution des problèmes de votre règle DynamoDB

Voici quelques éléments à vérifier si vous ne voyez pas les résultats que vous attendez.

- Vous avez une bannière d'erreur

  Si une erreur s'est produite lors de la publication du message d'entrée, corrigez d'abord cette erreur. Les étapes suivantes peuvent vous aider à corriger cette erreur.

- Le message d'entrée ne s'affiche pas dans le client MQTT

  Chaque fois que vous publiez votre message d'entrée dans le`device/22/data`, ce message doit apparaître dans le client MQTT si vous vous êtes abonné au`device/+/data`Filtre de rubrique comme décrit dans la procédure.

Choses à vérifier

- Vérifiez le filtre de rubrique auquel vous vous êtes abonné

  Si vous vous êtes abonné à la rubrique du message d'entrée comme décrit dans la procédure, vous devriez voir une copie du message d'entrée chaque fois que vous le publiez.

  Si le message ne s'affiche pas, vérifiez le nom de la rubrique à laquelle vous vous êtes abonné et comparez-le à la rubrique à laquelle vous avez publié. Les noms de rubrique sont sensibles à la casse et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique à laquelle vous avez publié la charge utile du message.
- Vérifier la fonction de publication des messages

  Dans le client MQTT, sousSubscriptions, choisissezpériphérique/+/données, vérifiez la rubrique du message de publication, puis choisissezPublier dans la rubrique. Vous devriez voir la charge utile du message dans la zone d'édition située en dessous de la rubrique s'afficher dans la liste des messages.

- Vous ne voyez pas vos données dans la table DynamoDB

  La première chose à faire est d'actualiser manuellement l'affichage en sélectionnant l'icône d'actualisation dans le coin supérieur droit de l'en-tête du tableau. Si cela n'affiche pas les données que vous recherchez, vérifiez ce qui suit.

  Choses à vérifier

  - Vérifiez la Région AWS de votre client MQTT et la règle que vous avez créée

    La console dans laquelle vous exécutez le client MQTT doit être dans le mêmeAWSRégion comme règle que vous avez créée.
  - Vérifiez la rubrique du message d'entrée dans l'instruction de requête de règle

    Pour que la règle fonctionne, elle doit recevoir un message avec le nom de rubrique correspondant au filtre de rubrique de la clause FROM de l'instruction de requête de règle.

    Vérifiez l'orthographe du filtre de rubrique dans l'instruction de requête de règle avec celle de la rubrique dans le client MQTT. Les noms de rubrique sont sensibles à la casse et la rubrique du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.
  - Vérifiez le contenu de la charge utile du message d'entrée

    Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message qui est déclarée dans l'instruction SELECT.

    Vérifiez l'orthographe du`temperature`dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ sont sensibles à la casse et l'`temperature`de l'instruction de requête de règle doit être identique à l'objet`temperature`dans la charge utile du message.

    Assurez-vous que le document JSON de la charge utile du message est correctement formaté. Si le JSON a des erreurs, telles qu'une virgule manquante, la règle ne pourra pas la lire.
  - Vérifiez les noms de clé et de champ utilisés dans l'action de règle

    Les noms de champs utilisés dans la règle de rubrique doivent correspondre à ceux trouvés dans la charge utile du message JSON du message publié.

    Ouvrez la règle que vous avez créée dans la console et vérifiez les noms des champs dans la configuration de l'action de règle avec ceux utilisés dans le client MQTT.
  - Vérifiez le rôle utilisé par la règle

L'action de règle doit être autorisée à recevoir la rubrique d'origine et à publier la nouvelle rubrique.

Les stratégies qui autorisent la règle à recevoir des données de message et à mettre à jour la table DynamoDB sont spécifiques aux rubriques utilisées. Si vous modifiez la rubrique ou le nom de table DynamoDB utilisé par la règle, vous devez mettre à jour le rôle de l'action de règle pour mettre à jour sa stratégie en conséquence.

Si vous soupçonnez que c'est le problème, modifiez l'action de la règle et créez un nouveau rôle. Les nouveaux rôles créés par l'action de règle reçoivent les autorisations nécessaires pour effectuer ces actions.

Passez en revue les résultats et les prochaines étapes

Après avoir envoyé quelques messages à la table DynamoDB avec cette règle, essayez de l'expérimenter pour voir comment la modification de certains aspects du didacticiel affecte les données écrites dans la table. Voici quelques idées pour commencer.

- Remplacez par*device_id*dans la rubrique du message d'entrée et observez l'effet sur les données. Vous pouvez l'utiliser pour simuler la réception de données provenant de plusieurs capteurs météorologiques.
- Modifiez les champs sélectionnés dans l'instruction de requête de règle et observez l'effet sur les données. Vous pouvez utiliser cette option pour filtrer les données stockées dans la table.
- Ajoutez une action de règle de republication pour envoyer un message MQTT pour chaque ligne ajoutée à la table. Vous pouvez l'utiliser pour le débogage.

Après avoir terminé ce didacticiel, consultez le suivant !

Formater une notification à l'aide d'unAWS Lambdafonction

Ce didacticiel explique comment envoyer les données de message MQTT à une instanceAWS Lambdaaction pour le formatage et l'envoi à un autreAWSservice. Dans ce didacticiel, leAWS Lambdautilise laAWSKit de développement logiciel (SDK) pour envoyer le message formaté à la rubrique Amazon SNS que vous avez créée dans le didacticiel sur la façon dethe section called “Envoi d'une notification Amazon SNS” (p. 139).

Dans le tutoriel sur la façon dethe section called “Envoi d'une notification Amazon SNS” (p. 139), le document JSON résultant de l'instruction de requête de la règle a été envoyé en tant que corps du message texte. Le résultat a été un message texte qui ressemblait à cet exemple :

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

Dans ce didacticiel, vous allez utiliser unAWS Lambdaaction de règle pour appeler unAWS Lambdaqui met en forme les données de l'instruction de requête de règle dans un format plus convivial, tel que cet exemple :

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

La .AWS Lambdaque vous allez créer dans ce didacticiel met en forme la chaîne de message à l'aide des données de l'instruction de requête de règle et appelle la fonctionPublication SNSde laAWSSDK pour créer la notification.

Ce que vous apprendrez dans ce didacticiel

- Comment créer et tester unAWS Lambdafonction
- Comment utiliser laAWSkit SDK dans unAWS Lambdafonction pour publier une notification Amazon SNS
- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester unAWS IoTRègle

Ce didacticiel vous prendra environ 45 minutes.

Dans ce didacticiel :

- Création d'unAWS LambdaFonction qui envoie un message texte (p. 153)
- Création d'unAWS IoTRègle avec uneAWS LambdaAction de règle (p. 156)
- Tester laAWS IoTRègle etAWS LambdaAction de règle (p. 157)
- Passez en revue les résultats et les prochaines étapes (p. 160)

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Configurer votre Compte AWS (p. 19)

  Vous aurez besoin de votre Compte AWS andAWS IoTPour suivre ce didacticiel.
- VérificationAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62)

  Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner à une rubrique. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans cette procédure.
- A terminé les didacticiels des autres règles de cette section

  Ce didacticiel nécessite la rubrique de notification SNS que vous avez créée dans le didacticiel sur la façon dethe section called “Envoi d'une notification Amazon SNS” (p. 139). Il suppose également que vous avez terminé les autres didacticiels relatifs aux règles de cette section.
- Consultez laAWS LambdaPrésentation de

  Si vous n'avez pas utiliséAWS Lambdaavant, révisionAWS LambdaandMise en route avec LambdaPour apprendre ses termes et concepts.

Création d'unAWS LambdaFonction qui envoie un message texte

La .AWS Lambdadans ce didacticiel reçoit le résultat de l'instruction de requête de règle, insère les éléments dans une chaîne de texte et envoie la chaîne résultante à Amazon SNS comme message dans une notification.

Contrairement au tutoriel sur la façon dethe section called “Envoi d'une notification Amazon SNS” (p. 139), qui a utilisé unAWS IoTpour envoyer la notification, ce didacticiel envoie la notification à partir de la fonction Lambda à l'aide d'une fonction deAWSKIT SDK. Cependant, la rubrique de notification Amazon SNS utilisée dans ce didacticiel est la même que celle que vous avez utilisée dans le didacticiel sur la façon dethe section called “Envoi d'une notification Amazon SNS” (p. 139).

Pour créer unAWS LambdaFonction qui envoie un message texte

1. Création d'un nouveauAWS Lambda.

   a. Dans la console AWS Lambda, choisissez Créer une fonction.
   b. DansCréation d'une fonction, sélectionnezUtiliser un plan.

      Recherchez et choisissez**`hello-world-python`**, puis choisissezConfiguration.

c. DansInformations de base :

i. DansNom de la fonction, entrez le nom de cette fonction,**`format-high-temp-notification`**.

ii. DansExecution role (Rôle d'exécution), choisissezCréez un rôle à partir deAWSModèles de stratégie.

iii. Dans Role name, entrez le nom du nouveau rôle,**`format-high-temp-notification-role`**.

iv. DansModèles de stratégie -facultatif, recherchez et sélectionnezPolitique de publication Amazon SNS.

v. Sélectionnez Create function (Créer une fonction).

2. Modifiez le code de Blueprint pour formater et envoyer une notification Amazon SNS.

a. Après avoir créé votre fonction, vous devriez voir l'Format-high temp-notificationPage de détails. Si vous ne le faites pas, ouvrez-le à partir de laLambdaFonctions.

b. DansFormat-high temp-notificationSur la page des détails, choisissez laConfigurationet faites défiler jusqu'à l'ongletCode de fonctionPanneau.

c. DansCode de fonction, dans la fenêtreEnvironment (Environnement), choisissez le fichier Python,`lambda_function.py`.

d. DansCode de fonction, supprimez tout le code du programme d'origine du Blueprint et remplacez-le par ce code.

```
import boto3
#
#   expects event parameter to contain:
#   {
#       "device_id": "32",
#       "reported_temperature": 38,
#       "max_temperature": 30,
#       "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
#   }
#
#   sends a plain text string to be used in a text message
#
#      "Device {0} reports a temperature of {1}, which exceeds the limit of {2}."
#
#   where:
#       {0} is the device_id value
#       {1} is the reported_temperature value
#       {2} is the max_temperature value
#
def lambda_handler(event, context):

    # Create an SNS client to send notification
    sns = boto3.client('sns')

    # Format text message from data
    message_text = "Device {0} reports a temperature of {1}, which exceeds the
 limit of {2}.".format(
            str(event['device_id']),
            str(event['reported_temperature']),
            str(event['max_temperature'])
        )

    # Publish the formatted message
    response = sns.publish(
            TopicArn = event['notify_topic_arn'],
            Message = message_text
        )
```

```
    return response
```

e. Choisissez Deploy (Déployer).

3. Dans une nouvelle fenêtre, recherchez le Amazon Resource Name (ARN) de votre rubrique Amazon SNS à partir du didacticiel sur la façon d'the section called “Envoi d'une notification Amazon SNS” (p. 139).

   a. Dans une nouvelle fenêtre, ouvrez la fenêtrePage Rubriques de la console Amazon SNS.

   b. DansRubriques, recherchez la pagehigh_temp_noticedans la liste des rubriques Amazon SNS.

   c. Recherchez la valeurARNde lahigh_temp_noticeRubrique de notification à utiliser à l'étape suivante.

4. Créez un cas de test pour votre fonction Lambda.

   a. DansLambdaFonctionsde la console, sur la pageFormat-high temp-notificationPage de détails, choisissezSélectionnez un événement de testEn haut à droite de la page (même si elle semble désactivée), puis choisissezConfigurer les événements de test.

   b. DansConfigurer l'événement de test, choisissezCréez un événement de test.

   c. DansNom de l'événement, saisissez**`SampleRuleOutput`**.

   d. Dans l'éditeur JSON ci-dessousNom de l'événement, collez cet exemple de document JSON. Voici un exemple de ce que votreAWS IoTLa règle enverra à la fonction Lambda.

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30,
  "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
}
```

   e. Reportez-vous à la fenêtre qui a leARNde lahigh_temp_noticeNotification et copiez la valeur de l'ARN.

   f. Remplacez par`notify_topic_arn`dans l'éditeur JSON avec l'ARN de votre rubrique de notification.

      Gardez cette fenêtre ouverte afin que vous puissiez utiliser à nouveau cette valeur ARN lorsque vous créez leAWS IoTRègle.

   g. Sélectionnez Create (Créer).

5. Testez la fonction avec des données d'échantillon.

   a. DansFormat-high temp-notification, dans le coin supérieur droit de la page, confirmez queSampleRuleOutputapparaît à côté de la fenêtreTest. Si ce n'est pas le cas, choisissez-le dans la liste des événements de test disponibles.

   b. Pour envoyer l'exemple de message de sortie de règle à votre fonction, sélectionnezTest.

Si la fonction et la notification ont tous deux fonctionné, vous obtiendrez un message texte sur le téléphone qui s'est abonné à la notification.

Si vous n'avez pas reçu de message texte sur le téléphone, vérifiez le résultat de l'opération. DansCode de fonction, dans le panneauRésultat de l'exécution, passez en revue la réponse pour trouver les erreurs qui se sont produites. Ne passez pas à l'étape suivante jusqu'à ce que votre fonction puisse envoyer la notification à votre téléphone.

Création d'unAWS IoTRègle avec uneAWS LambdaAction de règle

Dans cette étape, vous allez utiliser l'instruction de requête de règle pour formater les données du capteur de météo imaginaire à envoyer à une fonction Lambda, qui formera et enverra un message texte.

Un exemple de charge utile de message reçu des appareils météorologiques ressemble à ceci :

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

Dans cette règle, vous allez utiliser l'instruction de requête de règle pour créer une charge utile de message pour la fonction Lambda qui ressemble à ceci :

```
{
  "device_id": "32",
  "reported_temperature": 38,
  "max_temperature": 30,
  "notify_topic_arn": "arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice"
}
```

Il contient toutes les informations dont la fonction Lambda a besoin pour formater et envoyer le message texte correct.

Pour créer leAWS IoTRègle pour appeler une fonction Lambda

1. Ouverture d'Règleshub de laAWS IoTconsole.
2. Pour commencer à créer votre nouvelle règle dansRègles, choisissezCréer.
3. Dans la partie supérieure deCréer une règle :

   a. DansNom, entrez le nom de la règle,**`wx_friendly_text`**.

      N'oubliez pas de rappeler qu'un nom de règle doit être unique dans votre Compte AWS et Region, et il ne peut pas avoir d'espaces. Nous avons utilisé un caractère de soulignement dans ce nom pour séparer les deux mots du nom de la règle.

   b. DansDescription, décrivez la règle.

      Une description significative permet de se souvenir plus facilement de ce que fait cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, alors soyez aussi détaillée que possible.
4. DansInstruction de requête de règlesdeCréer une règle :

   a. DansUtilisation de SQL, sélectionnez**`2016-03-23`**.

   b. DansInstruction de requête de règlesDans la zone d'édition, entrez l'instruction :

```
SELECT
  cast(topic(2) AS DECIMAL) as device_id,
  temperature as reported_temperature,
  30 as max_temperature,
  'arn:aws:sns:us-east-1:57EXAMPLE833:high_temp_notice' as notify_topic_arn
```

```
FROM 'device/+/data' WHERE temperature > 30
```

Cette déclaration :

- Écoute les messages MQTT avec une rubrique qui correspond à la`device/+/data`et qui ont un`temperature`valeur supérieure à 30.
- Sélectionne le deuxième élément de la chaîne de rubrique, le convertit en nombre décimal, puis l'assigne à la propriété`device_id`field.
- Sélectionne la valeur du`temperature`à partir de la charge utile du message et l'affecte à la`reported_temperature`field.
- Crée une valeur constante,`30`, pour représenter la valeur limite et l'affecter à la propriété`max_temperature`field.
- Crée une valeur constante pour les`notify_topic_arn`field.

c. Reportez-vous à la fenêtre qui a leARNde lahigh_temp_noticeNotification et copiez la valeur de l'ARN.

d. Remplacez la valeur ARN (*`arn:aws:SNS : EXAMPLE833:HIGH_Temp_Notice`*) dans l'éditeur d'instructions de requête de règle avec l'ARN de votre rubrique de notification.

5. DansDéfinissez une ou plusieurs actions :

a. Pour ouvrir la liste des actions de règle pour cette règle, choisissezAction d'ajout.

b. DansSélectionner une action, choisissezEnvoyer un message. fonction Lambda.

c. Pour ouvrir la page de configuration de l'action sélectionnée, au bas de la liste des actions, sélectionnezConfigurer une action.

6. DansConfigurer une action :

a. DansNom de la fonction, choisissezTâche de sélection.

b. ChoisissezFormat-high temp-notification.

c. Au bas deConfigurer une action, choisissezAction d'ajout.

d. Pour créer la règle, en bas de l'ongletCréer une règle, choisissezCréer une règle.

Tester laAWS IoTRègle etAWS LambdaAction de règle

Pour tester votre nouvelle règle, vous utiliserez le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouverture d'Client MQTTAWS IoTconsoleDans une nouvelle fenêtre. Vous pouvez maintenant modifier la règle sans perdre la configuration de votre client MQTT. Si vous quittez le client MQTT pour accéder à une autre page de la console, vous perdrez vos abonnements ou vos journaux de messages.

Pour utiliser le client MQTT pour tester votre règle

1. DansClient MQTTAWS IoTconsole, abonnez-vous aux rubriques d'entrée, dans ce cas,`device/+/data`.

a. Dans le client MQTT, sousSubscriptions, choisissezS'abonner à une rubrique.

b. Dansrubrique abonnement, entrez la rubrique du filtre de rubrique d'entrée,**`device/+/data`**.

c. Conservez les paramètres par défaut des autres champs.

d. Choisissez Subscribe to topic (S'abonner à la rubrique).

DansSubscriptionscolumn, sousPublier dans une rubrique,**`device/+/data`**apparaît.

2. Publier un message dans la rubrique d'entrée avec un ID de périphérique spécifique,**`device/32/data`**. Vous ne pouvez pas publier dans les rubriques MQTT qui contiennent des caractères génériques.

a. Dans le client MQTT, sousSubscriptions, choisissezPublier dans la rubrique.

b. DansPublierDans le champ, entrez le nom de la rubrique en entrée,**`device/32/data`**.

c. Copiez les exemples de données affichés ici et, dans la zone d'édition située sous le nom de la rubrique, collez les exemples de données.

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

d. Pour publier votre message MQTT, choisissezPublier dans la rubrique.

3. Confirmez que le message texte a été envoyé.

a. Dans le client MQTT, sousSubscriptions, il y a un point vert à côté du sujet auquel vous vous êtes abonné plus tôt.

Le point vert indique qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.

b. UnderSubscriptions, choisissezpériphérique/+/donnéespour vérifier que la charge utile du message correspond à ce que vous venez de publier et ressemble à ceci :

```
{
  "temperature": 38,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

c. Vérifiez le téléphone que vous avez utilisé pour vous abonner à la rubrique SNS et confirmez que le contenu de la charge utile du message ressemble à ceci :

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

Si vous modifiez l'élément ID de rubrique dans la rubrique de message, n'oubliez pas que la diffusion de l'élément`topic(2)`à une valeur numérique ne fonctionnera que si cet élément de la rubrique du message ne contient que des caractères numériques.

4. Essayez d'envoyer un message MQTT dans lequel la température ne dépasse pas la limite.

a. Dans le client MQTT, sousSubscriptions, choisissezPublier dans la rubrique.

b. DansPublierDans le champ, entrez le nom de la rubrique en entrée,**`device/33/data`**.

c. Copiez les exemples de données affichés ici et, dans la zone d'édition située sous le nom de la rubrique, collez les exemples de données.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
```

```
    "bearing": 255
  }
}
```

d. Pour envoyer votre message MQTT, choisissezPublier dans la rubrique.

Le message que vous avez envoyé dans l'**`device/+/data`** ; cependant, étant donné que la valeur de température est inférieure à la température maximale dans l'instruction de requête de règle, vous ne devriez pas recevoir de message texte.

Si vous ne voyez pas le comportement correct, consultez les conseils de dépannage.

Dépannage deAWS LambdaRègle et notification

Voici quelques éléments à vérifier, au cas où vous ne verrez pas les résultats que vous attendez.

- Vous avez une bannière d'erreur

  Si une erreur s'est produite lors de la publication du message d'entrée, corrigez d'abord cette erreur. Les étapes suivantes peuvent vous aider à corriger cette erreur.
- Le message d'entrée ne s'affiche pas dans le client MQTT

  Chaque fois que vous publiez votre message d'entrée dans le`device/32/data`, ce message doit apparaître dans le client MQTT, si vous vous êtes abonné au`device/+/data`Filtre de rubrique comme décrit dans la procédure.

  ### Choses à vérifier

  - Vérifiez le filtre de rubrique auquel vous vous êtes abonné

    Si vous vous êtes abonné à la rubrique du message d'entrée comme décrit dans la procédure, vous devriez voir une copie du message d'entrée chaque fois que vous le publiez.

    Si le message ne s'affiche pas, vérifiez le nom de la rubrique à laquelle vous vous êtes abonné et comparez-le à la rubrique à laquelle vous avez publié. Les noms de rubrique sont sensibles à la casse et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique à laquelle vous avez publié la charge utile du message.
  - Vérifier la fonction de publication des messages

    Dans le client MQTT, sousSubscriptions, choisissezpériphérique/+/données, vérifiez la rubrique du message de publication, puis choisissezPublier dans la rubrique. Vous devriez voir la charge utile du message dans la zone d'édition située en dessous de la rubrique s'afficher dans la liste des messages.
- Vous ne recevez pas de message SMS

  Pour que votre règle fonctionne, elle doit avoir la stratégie correcte qui l'autorise à recevoir un message et à envoyer une notification SNS, et elle doit recevoir le message.

  ### Choses à vérifier

  - Vérifiez la Région AWS de votre client MQTT et la règle que vous avez créée

    La console dans laquelle vous exécutez le client MQTT doit être dans le mêmeAWSRégion comme règle que vous avez créée.
  - Vérifier que la valeur de température dans la charge utile du message dépasse le seuil de test

    Si la valeur de température est inférieure ou égale à 30, telle que définie dans l'instruction de requête de règle, la règle n'effectuera aucune de ses actions.

- Vérifiez la rubrique du message d'entrée dans l'instruction de requête de règle

  Pour que la règle fonctionne, elle doit recevoir un message avec le nom de rubrique correspondant au filtre de rubrique de la clause FROM de l'instruction de requête de règle.

  Vérifiez l'orthographe du filtre de rubrique dans l'instruction de requête de règle avec celle de la rubrique dans le client MQTT. Les noms de rubrique sont sensibles à la casse et la rubrique du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.
- Vérifiez le contenu de la charge utile du message d'entrée

  Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message qui est déclarée dans l'instruction SELECT.

  Vérifiez l'orthographe de la`temperature`dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ sont sensibles à la casse et l'`temperature`de l'instruction de requête de règle doit être identique à l'objet`temperature`dans la charge utile du message.

  Assurez-vous que le document JSON de la charge utile du message est correctement formaté. Si le JSON a des erreurs, telles qu'une virgule manquante, la règle ne pourra pas la lire.
- Vérifiez la notification Amazon SNS

  DansCréez une rubrique Amazon SNS qui envoie un message texte SMS (p. 140), reportez-vous à l'étape 3 qui décrit comment tester la notification Amazon SNS et tester la notification pour vous assurer que la notification fonctionne.
- Vérifiez la fonction Lambda

  DansCréation d'unAWS LambdaFonction qui envoie un message texte (p. 153), reportez-vous à l'étape 5 qui décrit comment tester la fonction Lambda à l'aide des données de test et tester la fonction Lambda.
- Vérifiez le rôle utilisé par la règle

  L'action de règle doit être autorisée à recevoir la rubrique d'origine et à publier la nouvelle rubrique.

  Les stratégies qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez la rubrique utilisée pour republier les données du message, vous devez mettre à jour le rôle de l'action de règle pour mettre à jour sa stratégie afin qu'elle corresponde à la rubrique actuelle.

  Si vous soupçonnez que c'est le problème, modifiez l'action Republier la règle et créez un nouveau rôle. Les nouveaux rôles créés par l'action de règle reçoivent les autorisations nécessaires pour effectuer ces actions.

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel :

- Vous avez créé unAWS IoTpour appeler une fonction Lambda qui a envoyé une notification Amazon SNS utilisant votre charge utile de message personnalisée.
- Vous avez utilisé une requête SQL simple et des fonctions dans une instruction de requête de règle pour créer une nouvelle charge utile de message pour votre fonction Lambda.
- Vous avez utilisé le client MQTT pour tester votreAWS IoTRègle.

Étapes suivantes

Après avoir envoyé quelques messages texte avec cette règle, essayez de l'expérimenter pour voir comment la modification de certains aspects du didacticiel affecte le message et quand il est envoyé. Voici quelques idées pour commencer.

- Remplacez par*device_id*dans la rubrique du message d'entrée et observez l'effet dans le contenu du message texte.
- Modifiez les champs sélectionnés dans l'instruction de requête de règle, mettez à jour la fonction Lambda pour les utiliser dans un nouveau message et observez l'effet dans le contenu du message texte.
- Modifiez le test dans l'instruction de requête de règle pour tester une température minimale au lieu d'une température maximale. Mettez à jour la fonction Lambda pour formater un nouveau message et n'oubliez pas de changer le nom de`max_temperature`.
- Pour en savoir plus sur la recherche d'erreurs qui peuvent se produire lors du développement et de l'utilisationAWS IoTrègles, voirSurveillance de AWS IoT (p. 352).

AWS IoTDidacticiels Device

Ces didacticiels vous montrent comment utiliser l'AWS IoTService Device Shadow pour stocker et mettre à jour les informations d'état d'un périphérique. Le document Shadow, qui est un document JSON, affiche la modification de l'état de l'appareil en fonction des messages publiés par un appareil, une application locale ou un service. Dans ce didacticiel, le document Ombre montre le changement de couleur d'une ampoule. Ces didacticielsmontrent également comment l'ombre stocke ces informations même lorsque l'appareil est déconnecté d'Internet, et transmet les dernières informations d'état à l'appareil lorsqu'il revient en ligne et demande ces informations.

Nous vous recommandons d'essayer ces tutoriels dans l'ordre où ils sont affichés ici, en commençant par leAWS IoTque vous devez créer et la configuration matérielle nécessaire, ce qui vous aide également à apprendre les concepts de manière incrémentielle. Ces tutoriels montrent comment configurer et connecter un périphérique Raspberry Pi à utiliser avecAWS IoT. Si vous ne disposez pas du matériel requis, vous pouvez suivre ces didacticiels en les adaptant à un appareil de votre choix ou enCréation d'un appareil virtuel avec Amazon EC2 (p. 39).

Présentation du scénario du didacticiel

Le scénario de ces didacticiels est une application ou un service local qui change la couleur d'une ampoule et qui publie ses données dans des rubriques d'ombre réservées. Ces didacticiels sont similaires à la fonctionnalité Device Shadow décrite dans leDidacticiel de démarrage (p. 21)etSur un appareil Raspberry Pi. Les didacticiels de cette section se concentrent sur une ombre classique unique tout en montrant comment vous pouvez accueillir des ombres nommées ou plusieurs périphériques.

Les didacticiels suivants de cette section vous apprendront à utiliser leAWS IoTService Device Shadow.

- CréerAWS IoTet connectez Raspberry Pi pour exécuter l'application shadow (p. 163)

  Ce tutoriel montre comment configurer un périphérique Raspberry Pi pour la connexion avecAWS IoT. Vous allez également créer unAWS IoTet une ressource de chose, téléchargez les certificats, puis attachez la stratégie à cette ressource de chose. Ce didacticiel vous prendra environ 30 minutes.
- Installez le kit SDK des appareils et exécutez l'**`shadow.py`**exemple d'application pour Device Shadows (p. 168)

  Ce didacticiel montre comment installer les outils requis, les logiciels et leAWS IoTSDK de périphérique pour Python, puis exécutez l'exemple d'application d'ombre. Ce didacticiel s'appuie sur les concepts présentés dansConnect un Raspberry Pi ou un autre appareil (p. 53)et prend 20 minutes.
- Interagissez avec Device Shadow en utilisant**`shadow.py`**exemple d'application et de client de test MQTT (p. 174)

Ce didacticiel montre comment vous utilisez le`shadow.py`Exemple d'application etAWS IoTconsolepour observer l'interaction entreAWS IoTDevice Shadows et les changements d'état de l'ampoule. Le didacticiel montre également comment envoyer des messages MQTT aux rubriques réservées de Device Shadow. Ce didacticiel vous prendra 45 minutes.

AWS IoTPrésentation Device Shadow

Un Device Shadow est une représentation virtuelle persistante d'un périphérique géré par unRessource d'objet (p. 203)que vous avez créées dans laAWS IoTRegistre. Le document Shadow est un document JSON ou un document de notation JavaScript utilisé pour stocker et extraire les informations sur l'état en cours d'un appareil. Vous pouvez utiliser le shadow pour obtenir et définir l'état d'un appareil sur les rubriques MQTT ou HTTP REST API, que l'appareil soit connecté ou non à Internet.

Un document Shadow contient un`state`qui décrit ces aspects de l'état de l'appareil.

- `desired` : les applications spécifient les états souhaités des propriétés de l'appareil en mettant à jour le`desired`Objet
- `reported` : les appareils rapportent leur état actuel dans l'`reported`Objet
- `delta` :AWS IoTrapporte les différences entre l'état souhaité et l'état rapporté dans le`delta`Objet

Voici un exemple de document d'état Shadow.

```
{
  "state": {
    "desired": {
      "color": "green"
      },
    "reported": {
      "color": "blue"
      },
    "delta": {
      "color": "green"
      }
   }
}
```

Pour mettre à jour le document Shadow d'un périphérique, vous pouvez utiliser l'outilRubriques MQTT réservées (p. 103), leAPI REST Device Shadow (p. 573)qui prennent en charge l'`GET`,`UPDATE`, et`DELETE`avec HTTP, et la méthodeAWS IoTCLI.

Dans l'exemple précédent, supposons que vous souhaitiez modifier l'`desired`color`yellow`. Pour ce faire, envoyez une demande auUpdateThingShadow (p. 574)ou publiez un message dans l'interface utilisateur.Mise à jour (p. 579)la rubrique ;`$aws/things/THING_NAME/shadow/update`.

```
{
  "state": {
    "desired": {
      "color": yellow
    }
  }
}
```

Les mises à jour concernent uniquement les champs spécifiés dans la demande. Après avoir mis à jour correctement l'Device Shadow,AWS IoTPublie laNEW `desired`à l'état`delta`la rubrique ;`$aws/things/THING_NAME/shadow/delta`. Le document Shadow dans ce cas ressemble à ceci :

```
{
```

```
  "state": {
    "desired": {
      "color": yellow
    },
    "reported": {
      "color": green
    },
    "delta": {
      "color": yellow
      }
  }
}
```

Le nouvel état est ensuite rapporté auAWS IoTShadow des appareils utilisant la`Update`topic`$aws/things/THING_NAME/shadow/update`Avec le message JSON suivant :

```
{
  "state": {
    "reported": {
      "color": yellow
    }
  }
}
```

Si vous souhaitez obtenir les informations sur l'état actuel, envoyez une demande à l'GetThingShadow (p. 573)ou publiez un message MQTT sur leFaites (p. 578)la rubrique ;`$aws/things/THING_NAME/shadow/get`.

Pour en savoir plus sur l'utilisation du service Device Shadow, consultez.Service AWS IoT Device Shadow (p. 547).

Pour en savoir plus sur l'utilisation des shadows de périphériques sur les appareils, dans les applications et dans les services, consultez.Utilisation des shadows sur les appareils (p. 551)andUtilisation des shadows dans les applications et les services (p. 554).

Pour plus d'informations sur l'interaction avecAWS IoTshadows, voirInteraction avec les shadows (p. 566).

Pour en savoir plus sur les rubriques réservées MQTT et les API REST HTTP, consultez.Rubriques MQTT de Device Shadow (p. 577)andAPI REST Device Shadow (p. 573).

CréerAWS IoTet connectez Raspberry Pi pour exécuter l'application shadow

Ce didacticiel montre comment configurer et configurer un périphérique Raspberry Pi et créer leAWS IoTdont un périphérique a besoin pour se connecter et échanger des messages MQTT.

> Note
>
> Si vous avez l'intention dethe section called “Créez un appareil virtuel avec Amazon EC2” (p. 39), vous pouvez ignorer cette page et continuer àthe section called “Configurer votre appareil” (p. 39). Vous allez créer ces ressources lorsque vous créez votre objet virtuel. Si vous souhaitez utiliser un autre appareil à la place du Raspberry Pi, vous pouvez essayer de suivre ces tutoriels en les adaptant à un appareil de votre choix.

Dans ce didacticiel, vous allez apprendre à faire :

- Configurez un périphérique Raspberry Pi et configurez-le pour une utilisation avecAWS IoT.
- Création d'unAWS IoT, qui autorise votre appareil à interagir avecAWS IoTServices .

- Créer une ressource d'objet dansAWS IoTles certificats de périphérique X.509, puis joignez le document de stratégie.

  La chose est la représentation virtuelle de votre appareil dans leAWS IoTRegistre. Le certificat authentifie votre appareil surAWS IoTCore, et le document de stratégie autorise votre appareil à interagir avecAWS IoT.

Comment exécuter ce didacticiel ?

Pour exécuter le`shadow.py`exemple d'application pour Device Shadows, vous aurez besoin d'un périphérique Raspberry Pi qui se connecte àAWS IoT. Nous vous recommandons de suivre ce tutoriel dans l'ordre qu'il est présenté ici, en commençant par configurer le Raspberry Pi et ses accessoires, puis en créant une stratégie et en attachant la stratégie à une ressource de chose que vous créez. Vous pouvez ensuite suivre ce didacticiel en utilisant l'interface utilisateur graphique (GUI) prise en charge par Raspberry Pi pour ouvrir leAWS IoTsur le navigateur Web de l'appareil, ce qui facilite également le téléchargement des certificats directement sur votre Raspberry Pi pour la connexion àAWS IoT.

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Un Compte AWS . Si vous n'en avez pas, suivez les étapes décrites dansConfigurer votre Compte AWS (p. 19)Avant de continuer. Vous aurez besoin de votre Compte AWS andAWS IoTPour terminer ce didacticiel, veuillez consulter.
- Le Raspberry Pi et ses accessoires nécessaires. Vous aurez besoin de :
  - ARaspberry Pi 3 Modèle Bou un modèle plus récent. Ce tutoriel peut fonctionner sur des versions antérieures du Raspberry Pi, mais nous ne l'avons pas testé.
  - Système d'exploitation Raspberry Pi (32 bits)ou version ultérieure. Nous vous recommandons d'utiliser la dernière version du système d'exploitation Raspberry Pi. Les versions antérieures du système d'exploitation peuvent fonctionner, mais nous ne l'avons pas testé.
  - Une connexion Ethernet ou Wi-Fi.
  - Clavier, souris, moniteur, câbles et blocs d'alimentation.

Ce didacticiel vous prendra environ 30 minutes.

Configuration et configuration d'un périphérique Raspberry Pi

Dans cette section, nous allons configurer un périphérique Raspberry Pi pour une utilisation avecAWS IoT.

> **Important**
>
> L'adaptation de ces instructions à d'autres périphériques et systèmes d'exploitation peut s'avérer difficile. Vous devez bien comprendre votre appareil pour pouvoir interpréter ces instructions et les appliquer à votre appareil. Si vous rencontrez des difficultés, vous pouvez essayer l'une des autres options de l'appareil comme alternative, par exempleCréez un appareil virtuel avec Amazon EC2 (p. 39)ouUtilisez votre PC Windows ou Linux ou Mac en tant queAWS IoTAppareil (p. 47).

Vous devez configurer votre Raspberry Pi de sorte qu'il puisse démarrer le système d'exploitation (OS), se connecter à Internet et vous permettre d'interagir avec lui à l'aide d'une interface de ligne de commande. Vous pouvez également utiliser l'interface utilisateur graphique (GUI) d'Raspberry Pi pour ouvrir leAWS IoTet exécutez le reste de ce didacticiel.

Pour configurer le Raspberry Pi :

1. Insérez la carte SD dans l'emplacement microSD de l'appareil Raspberry Pi. Certaines cartes SD sont préchargées avec un gestionnaire d'installation qui vous invite avec un menu pour installer le système d'exploitation après le démarrage de la carte. Vous pouvez également utiliser l'imageur Raspberry Pi pour installer le système d'exploitation sur votre carte.

2. Connect un téléviseur ou un moniteur HDMI au câble HDMI qui se connecte au port HDMI du Raspberry Pi.
3. Connect le clavier et la souris aux ports USB du Raspberry Pi, puis branchez l'adaptateur secteur pour démarrer la carte.

Après le démarrage du Raspberry Pi, si la carte SD est préchargée avec le gestionnaire d'installation, un menu apparaît pour installer le système d'exploitation. Si vous rencontrez des problèmes dans l'installation du système d'exploitation, vous pouvez essayer les étapes suivantes. Pour en savoir plus sur la configuration du Raspberry Pi, consultez.Configuration de votre Raspberry Pi.

Si vous rencontrez des problèmes dans la configuration du Raspberry Pi :

- Vérifiez si vous avez inséré la carte SD avant de démarrer la carte. Si vous branchez la carte SD après avoir démarré la carte, le menu d'installation risque de ne pas apparaître.
- Assurez-vous que le téléviseur ou le moniteur est allumé et que l'entrée correcte est sélectionnée.
- Assurez-vous d'utiliser le logiciel compatible Raspberry Pi.

Si le problème persiste ou si vous avez besoin d'aide supplémentaire, consultez.Obtenir de l'aide pour votre Raspberry Pi.

Après avoir installé le système d'exploitation, utilisez le navigateur Web de Raspberry Pi pour ouvrir leAWS IoTPour poursuivre le reste des étapes de ce didacticiel.

Création d'unAWS IoTStratégie pour l'shadow d'objet

Les certificats X.509 authentifient votre appareil avec AWS IoT Core .AWS IoTsont attachées au certificat qui permet au périphérique d'effectuerAWS IoT, telles que l'abonnement ou la publication aux rubriques réservées MQTT utilisées par le service Device Shadow. Votre appareil présente son certificat lorsqu'il se connecte et envoie des messages à AWS IoT Core .

Au cours de cette procédure, vous créerez une stratégie qui permettra à votre appareil d'exécuter leAWS IoTnécessaires pour exécuter l'exemple de programme. Nous vous recommandons de créer une stratégie qui accorde uniquement les autorisations requises pour exécuter la seule tâche. Vous créez l'AWS IoTd'abord, puis attachez-la au certificat de périphérique que vous allez créer ultérieurement.

Pour créer une stratégie AWS IoT

1. Dans le menu de gauche, choisissezSecure, puisStratégies. Si votre compte possède des stratégies existantes, choisissezCréer, dans le cas contraire, sur leVous n'avez pas encore de policePage, choisissezCréer une stratégie.
2. Sur la page Create a policy (Créer une stratégie) :

   a. Entrez un nom pour la stratégie dans la zoneNom(par exemple,**`My_Device_Shadow_policy`**). N'utilisez pas d'informations personnelles identifiables dans vos noms de stratégie.

   b. Dans le document de stratégie, vous décrivez les actions de connexion, d'abonnement, de réception et de publication qui donnent à l'appareil l'autorisation de publier et de s'abonner aux rubriques réservées MQTT.

      Copiez l'exemple de stratégie suivant et collez-le dans votre document de stratégie. Remplacez`thingname`avec le nom de la chose que vous allez créer (par exemple,`My_light_bulb`),`region`avec leAWS IoTRégion dans laquelle vous utilisez les services, et`account`avec vos recettes Compte AWS Nombre Pour plus d'informations surAWS IoTStratégies, consultezStratégies AWS IoT Core (p. 270).

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

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get/
accepted",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/get/
rejected",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
accepted",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
rejected",
        "arn:aws:iot:region:account:topic/$aws/things/thingname/shadow/update/
delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/get/
accepted",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/get/
rejected",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/accepted",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/rejected",
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingname/shadow/
update/delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "iot:Connect",
      "Resource": "arn:aws:iot:region:account:client/test-*"
    }
  ]
}
```

Créer une ressource de chose et attacher la stratégie à la chose

Périphériques connectés àAWS IoTpeut être représenté parRessources d'objetdans leAWS IoTRegistre. ARessource d'objetreprésente un appareil spécifique ou une entité logique, telle que l'ampoule de ce didacticiel.

Pour apprendre à créer un objet dansAWS IoT, suivez les étapes décrites dansCréer un objet d'objet (p. 37). Voici quelques éléments clés à noter lorsque vous suivez les étapes de ce didacticiel :

1. ChoisissezCréer un objet unique, et dans leNom, entrez un nom pour l'objet qui est identique à l'objet`thingname`(par exemple,`My_light_bulb`) que vous avez spécifié lors de la création de la stratégie précédemment.

   Vous ne pouvez pas modifier un nom d'objet une fois qu'il a été créé. Si vous lui avez donné un autre nom que`thingname`, créez une nouvelle chose avec le nom`thingname`et supprimez l'ancienne chose.

   > **Note**
   >
   > N'utilisez pas d'informations personnelles identifiables dans votre nom d'objet. Le nom de chose peut apparaître dans les communications et les rapports non chiffrés.

2. Nous vous recommandons de télécharger chacun des fichiers de certificat sur leCertificat créé !dans un emplacement où vous pouvez facilement les trouver. Vous devrez installer ces fichiers pour exécuter l'exemple d'application.

   Nous vous recommandons de télécharger les fichiers dans un`certs`dans votre`home`sur le Raspberry Pi et nommez chacun d'eux avec un nom plus simple comme suggéré dans le tableau suivant.

   Noms des fichiers de certificat

Fichier	Chemin d'accès du fichier
Certificat racine de l'autorité de certification	`~/certs/Amazon-root-CA-1.pem`
Certificat de l'appareil	`~/certs/device.pem.crt`
Clé privée	`~/certs/private.pem.key`

3. Après avoir activé le certificat pour activer les connexions àAWS IoT, choisissezAttacher une stratégieet assurez-vous que vous attachez la stratégie que vous avez créée (par exemple,**`My_Device_Shadow_policy`**) à la chose.

   Une fois que vous avez créé un objet, vous pouvez voir votre ressource chose affichée dans la liste des objets dans laAWS IoTconsole

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous apprenez à faire :

- Configurez et configurez le périphérique Raspberry Pi.
- Création d'unAWS IoTqui autorise votre appareil à interagir avecAWS IoTServices .
- Créez une ressource objet et un certificat de périphérique X.509 associé, et joignez-y le document de stratégie.

Étapes suivantes

Vous pouvez désormais installer l'objetAWS IoTKit SDK des appareils pour Python, exécutez l'objet`shadow.py`et utilisez Device Shadows pour contrôler l'état. Pour plus d'informations sur l'exécution de ce didacticiel, consultez.Installez le kit SDK des appareils et exécutez l'`shadow.py`exemple d'application pour Device Shadows (p. 168).

Tutoriels dans cette section

- Installez le kit SDK des appareils et exécutez l'shadow.pyexemple d'application pour Device Shadows (p. 168)
- Interagissez avec Device Shadow en utilisantshadow.pyexemple d'application et de client de test MQTT (p. 174)

Installez le kit SDK des appareils et exécutez l'`shadow.py`exemple d'application pour Device Shadows

Cette section explique comment installer le logiciel requis et leAWS IoTKit SDK des appareils pour Python et exécutez l'`shadow.py`pour modifier le document Shadow et contrôler l'état de l'ombre.

Dans ce didacticiel, vous allez apprendre à faire :

- Utilisez le logiciel installé etAWS IoTSDK de périphérique pour Python pour exécuter l'exemple d'application.
- Découvrez comment la saisie d'une valeur à l'aide de l'exemple d'application publie la valeur désirée dans l'ongletAWS IoTconsole
- Vérifiez les`shadow.py`et comment il utilise le protocole MQTT pour mettre à jour l'état de l'ombre.

Avant d'exécuter ce didacticiel :

Vous devez avoir configuré votre adresse. Compte AWS , configuré votre périphérique Raspberry Pi et créé unAWS IoTqui donne à l'appareil les autorisations de publier et de s'abonner aux rubriques réservées MQTT du service Device Shadow. Pour plus d'informations, consultez CréerAWS IoTet connectez Raspberry Pi pour exécuter l'application shadow (p. 163).

Vous devez également avoir installé Git, Python et leAWS IoTKit SDK des périphériques pour Python. Ce didacticiel s'appuie sur les concepts présentés dans le didacticielConnect un Raspberry Pi ou un autre appareil (p. 53). Si vous n'avez pas essayé ce didacticiel, nous vous recommandons de suivre les étapes décrites dans ce didacticiel pour installer les fichiers de certificat et le SDK de périphérique, puis de revenir à ce didacticiel pour exécuter le`shadow.py`Exemple d'application

Dans ce didacticiel, vous allez :

- Exécutez l'exemple d'application shadow.py (p. 168)
- Passez en revue l'exemple d'application shadow.py Device SDK (p. 171)
- Résolution des problèmes liés à l'outilshadow.pyExemple d'application (p. 172)
- Passez en revue les résultats et les prochaines étapes (p. 174)

Ce didacticiel vous prendra environ 20 minutes.

Exécutez l'exemple d'application shadow.py

Avant d'exécuter le`shadow.py`, vous aurez besoin des informations suivantes en plus des noms et de l'emplacement des fichiers de certificat que vous avez installés.

Valeurs des paramètres d'application

Paramètre	Où chercher la valeur
votre-iot-nom-chose	Nom du kitAWS IoTCe que vous avez créé précédemment dansthe section called “Créer une ressource de chose et attacher la stratégie à la chose” (p. 166). Pour trouver cette valeur, dans la zoneAWS IoTconsole, choisissezGérer, puisObjets.
votre-iot-point de terminaison	La *votre-iot-point de terminaison*a un format de :*endpoint_id*-`ats.iot.`*region*`.amazonaws.com`, par

Paramètre	Où chercher la valeur
	exemple,`a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com`. Pour trouver cette valeur : 1. DansAWS IoTconsole, choisissezGérer, puisObjets. 2. Choisissez l'objet IoT que vous avez créé pour votre appareil,my_Light_bulbe, que vous avez utilisé précédemment, puis choisissezInteragir. Sur la page de détails de l'objet, votre point de terminaison s'affiche dans leHTTPSSection.

Installez et exécutez l'exemple d'application

1. Accédez au répertoire exemple d'application.

```
cd ~/aws-iot-device-sdk-python-v2/samples
```

2. Dans la fenêtre de la ligne de commande, remplacez*votre-iot-point de terminaison*and*votre-iot-nom-chose*comme indiqué et exécutez cette commande.

```
python3 shadow.py --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt
 --key ~/certs/private.pem.key --endpoint your-iot-endpoint --thing-name your-iot-
thing-name
```

3. Observez que l'exemple d'application :

   1. Se connecte àAWSService IoT pour votre compte.
   2. S'abonne à`Delta`Événements et`Update`and`Get`Réponses.
   3. Vous invite à entrer une valeur souhaitée dans le terminal.
   4. Affiche les résultats similaires à ce qui suit :

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to Delta events...
Subscribing to Update responses...
Subscribing to Get responses...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
  Shadow contains reported value 'off'.
Enter desired value:
```

> **Note**
>
> Si vous rencontrez des problèmes dans l'exécution du`shadow.py`Exemple d'application, révisionthe section called “Résolution des problèmes liés à l'outil`shadow.py`Exemple d'application” (p. 172). Pour obtenir des informations supplémentaires susceptibles de vous aider à corriger le problème, ajoutez le`--verbosity Debug`à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'elle fait.

Entrez des valeurs et observez les mises à jour dans le document Shad

Vous pouvez entrer des valeurs dans le terminal pour spécifier la propriété`desired`, qui met également à jour la valeur`reported`Valeur . Disons que vous entrez la couleur`yellow`Dans le terminal. La .`reported`est également mise à jour à la couleur`yellow`. Voici les messages affichés dans le terminal :

```
Enter desired value:
yellow
Changed local shadow value to 'yellow'.
Updating reported shadow value to 'yellow'...
Update request published.
Finished updating reported shadow value to 'yellow'.
```

Lorsque vous publiez cette demande de mise à jour,AWS IoTcrée une ombre classique par défaut pour la ressource chose. Vous pouvez observer la demande de mise à jour que vous avez publiée dans le`reported`and`desired`valeurs dans laAWS IoTen regardant le document Shadow pour la ressource chose que vous avez créée (par exemple,`My_light_bulb`). Pour afficher la mise à jour dans le document Shadow :

1. DansAWS IoT, choisissezGérer, puis.Objets.
2. Dans la liste des objets affichés, sélectionnez l'objet que vous avez créé, choisissezShadows, puisOmbre classique.

Le document Shadow doit ressembler à l'objet suivant, montrant le`reported`and`desired`valeurs définies sur la couleur`yellow`. Vous voyez ces valeurs dans laÉtat de l'ombredu document.

```
{
  "desired": {
    "welcome": "aws-iot",
    "color": "yellow"
  },
  "reported": {
    "welcome": "aws-iot",
    "color": "yellow"
  }
}
```

Vous voyez également unMétadonnéesqui contient les informations d'horodatage et le numéro de version de la demande.

Vous pouvez utiliser la version du document d'état pour vous assurer que vous mettez à jour la version la plus récente d'un document shadow d'appareil. Si vous envoyez une autre demande de mise à jour, le numéro de version est incrémenté de 1. Lorsque vous fournissez une version dans une demande de mise à jour, le service rejette la demande avec un code de réponse de conflit HTTP 409 si la version actuelle du document d'état ne correspond pas à la version fournie.

```
{
  "metadata": {
    "desired": {
      "welcome": {
        "timestamp": 1620156892
      },
      "color": {
        "timestamp": 1620156893
      }
    },
    "reported": {
      "welcome": {
        "timestamp": 1620156892
      },
```

```
      "color": {
        "timestamp": 1620156893
      }
    }
  },
  "version": 10
}
```

Pour en savoir plus sur le document Shadow et observer les modifications apportées aux informations d'état, passez au didacticiel suivantInteragissez avec Device Shadow en utilisant`shadow.py`exemple d'application et de client de test MQTT (p. 174)Comme cela est décrit dans laPassez en revue les résultats et les prochaines étapes (p. 174)section de ce didacticiel. Le cas échéant, vous pouvez également apprendre sur l'`shadow.py`et comment il utilise le protocole MQTT dans la section suivante.

Passez en revue l'exemple d'application shadow.py Device SDK

Cette section passe en revue les`shadow.py`Exemple d'application depuis l'AWS IoTKit SDK des périphériques version 2 pour Pythonutilisé dans ce didacticiel. Ici, nous allons examiner comment il se connecte à AWS IoT Core en utilisant le protocole MQTT et MQTT sur WSS. La .AWSExécution courante (AWS-CRT)fournit la prise en charge du protocole de communication de bas niveau et est incluse avec leAWS IoTKit SDK des périphériques v2 pour Python.

Bien que ce tutoriel utilise MQTT et MQTT sur WSS,AWS IoTprend en charge les périphériques qui publient des requêtes HTTPS. Pour obtenir un exemple de programme Python qui envoie un message HTTP à partir d'un appareil, consultez leExemple de code HTTPS (p. 86)Utilisation de Python`requests`bibliothèque.

Pour plus d'informations sur la façon dont vous pouvez prendre une décision éclairée sur le protocole à utiliser pour les communications de votre appareil, consultez leChoisir un protocole pour la communication de votre appareil (p. 81).

MQTT

La .`shadow.py`Exemples d'appel`mtls_from_path`(illustré ici) dans la`mqtt_connection_builder`Pour établir une connexion avec AWS IoT Core En utilisant le protocole MQTT.`mtls_from_path`utilise les certificats X.509 et TLS v1.2 pour authentifier le périphérique. La .AWS-La bibliothèque CRT gère les détails de niveau inférieur de cette connexion.

```
mqtt_connection = mqtt_connection_builder.mtls_from_path(
    endpoint=args.endpoint,
    cert_filepath=args.cert,
    pri_key_filepath=args.key,
    ca_filepath=args.root_ca,
    client_bootstrap=client_bootstrap,
    on_connection_interrupted=on_connection_interrupted,
    on_connection_resumed=on_connection_resumed,
    client_id=args.client_id,
    clean_session=False,
    keep_alive_secs=6
)
```

- `endpoint`est votreAWS IoTque vous avez transmis à partir de la ligne de commande et`client_id`est l'ID qui identifie de manière unique ce périphérique dans la boîte de dialogue Région AWS .
- `cert_filepath`,`pri_key_filepath`, et`ca_filepath`sont les chemins d'accès aux fichiers de certificat et de clé privée du périphérique, ainsi qu'au fichier d'autorité de certification racine.
- `client_bootstrap`est l'objet d'exécution commun qui gère les activités de communication de socket, et est instancié avant l'appel à`mqtt_connection_builder.mtls_from_path`.
- `on_connection_interrupted`and`on_connection_resumed`sont des fonctions de rappel à appeler lorsque la connexion du périphérique est interrompue et reprise.

- `clean_session`est de savoir s'il faut démarrer une nouvelle session persistante, ou si une session est présente, se reconnecter à une session existante.`keep_alive_secs`est la valeur keep alive, en secondes, à envoyer dans le`CONNECT`de la demande. Un ping sera automatiquement envoyé à cet intervalle. Le serveur suppose que la connexion est perdue s'il ne reçoit pas de ping après 1,5 fois cette valeur.

La `.shadow.py`exemple appelle également`websockets_with_default_aws_signing`dans le`mqtt_connection_builder`Pour établir une connexion avec AWS IoT Core en utilisant le protocole MQTT sur WSS. MQTT sur WSS utilise également les mêmes paramètres que MQTT et prend ces paramètres supplémentaires :

- `region`est leAWSsignature Région utilisée par l'authentification Signature V4, et`credentials_provider`est leAWSinformations d'identification fournies à utiliser pour l'authentification. La région est transmise à partir de la ligne de commande, et l'`credentials_provider`est instancié juste avant l'appel à`mqtt_connection_builder.websockets_with_default_aws_signing`.
- `websocket_proxy_options`est les options de proxy HTTP, si vous utilisez un hôte proxy. Dans`shadow.py`, cette valeur est instanciée juste avant l'appel à`mqtt_connection_builder.websockets_with_default_aws_signing`.

Abonnez-vous aux sujets et événements Shadow

La `.shadow.py`échantillon attemps pour établir une connexion et attend d'être entièrement connecté. S'il n'est pas connecté, les commandes sont mises en file d'attente. Une fois connecté, l'exemple s'abonne aux événements delta et met à jour et reçoit des messages, et publie des messages avec un niveau de qualité de service (QoS) de 1 (`mqtt.QoS.AT_LEAST_ONCE`).

Lorsqu'un périphérique s'abonne à un message de niveau QoS 1, le courtier de messages enregistre les messages auxquels le périphérique est abonné jusqu'à ce qu'ils puissent être envoyés à l'appareil. Le courtier de messages renvoie les messages jusqu'à ce qu'il reçoive un`PUBACK`réponse de l'appareil.

Pour plus d'informations sur le protocole MQTT, consultez.Vérifiez le protocole MQTT (p. 114)andMQTT (p. 82).

Pour plus d'informations sur la façon dont MQTT, MQTT sur WSS, les sessions persistantes et les niveaux QoS utilisés dans ce didacticiel, consultezPassez en revue l'exemple d'application pubsub.py Device SDK (p. 115).

Résolution des problèmes liés à l'outil`shadow.py`Exemple d'application

Lorsque vous exécutez le`shadow.py`, vous devriez voir certains messages affichés dans le terminal et une invite pour entrer un`desired`Valeur . Si le programme génère une erreur, alors pour déboguer l'erreur, vous pouvez commencer par vérifier si vous avez exécuté la commande correcte pour votre système.

Dans certains cas, le message d'erreur peut indiquer des problèmes de connexion et ressembler à :`Host name was invalid for dns resolution`ou`Connection was closed unexpectedly`. Dans de tels cas, voici quelques éléments que vous pouvez vérifier :

- Vérifiez l'adresse du point de terminaison dans la commande

  Vérifiez les`endpoint`dans la commande que vous avez saisie pour exécuter l'exemple d'application, (par exemple,`a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com`) et vérifiez cette valeur dans la boîte de dialogueAWS IoTconsole.

  Pour vérifier si vous avez utilisé la valeur correcte :

  1. DansAWS IoTconsole, choisissezGérer, puis.Objets.

2. Choisissez l'objet que vous avez créé pour votre exemple d'application (par exemple,my_Light_bulbe), puis.Interagir.

Sur la page de détails de l'objet, votre point de terminaison s'affiche dans leHTTPSSection. Vous devez également voir un message qui dit :`This thing already appears to be connected.`

- Vérifier l'activation du certificat

  Les certificats authentifient votre appareil avec AWS IoT Core .

  Pour vérifier si votre certificat est actif :

  1. DansAWS IoTconsole, choisissezGérer, puis.Objets.
  2. Choisissez l'objet que vous avez créé pour votre exemple d'application (par exemple,my_Light_bulbe), puis.Sécurité.
  3. Sélectionnez le certificat, puis, dans la page de détails du certificat, choisissez Sélectionner le certificat, puis, dans la page de détails du certificat, choisissezActions.

  Si dans la liste déroulanteActivaten'est pas disponible et vous ne pouvez choisirDeactivate, votre certificat est actif. Dans le cas contraire, choisissezActivateet exécutez à nouveau l'exemple de programme.

  Si le programme ne s'exécute toujours pas, vérifiez les noms des fichiers de certificat dans la`certs`folder.

- Vérifiez la stratégie attachée à la ressource de chose

  Alors que les certificats authentifient votre appareil,AWS IoTpermettent au périphérique d'effectuerAWS IoT, telles que l'abonnement ou la publication à des rubriques réservées MQTT.

  Pour vérifier si la stratégie correcte est attachée :

  1. Recherchez le certificat comme décrit précédemment, puis choisissezStratégies.
  2. Choisissez la stratégie affichée et vérifiez si elle décrit la`connect`,`subscribe`,`receive`, et`publish`qui donnent à l'appareil l'autorisation de publier et de s'abonner aux rubriques réservées MQTT.

     Pour obtenir un exemple de stratégie, consultez.Création d'unAWS IoTStratégie pour l'shadow d'objet (p. 165).

  Si vous voyez des messages d'erreur indiquant des problèmes de connexion àAWS IoT, cela peut être dû aux autorisations que vous utilisez pour la stratégie. Si tel est le cas, nous vous recommandons de commencer par une stratégie qui fournit un accès complet àAWS IoTRessources, puis exécutez à nouveau l'exemple de programme. Vous pouvez modifier la stratégie actuelle ou choisir la stratégie actuelle, choisissezDetach, puis créez une autre stratégie qui fournit un accès complet et l'attachez-la à votre ressource de truc. Vous pouvez ensuite limiter la stratégie aux actions et aux stratégies dont vous avez besoin pour exécuter le programme.

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

- Vérifiez l'installation de votre kit SDK de périphérique

Si le programme ne s'exécute toujours pas, vous pouvez réinstaller le SDK de périphérique pour vous assurer que l'installation de votre SDK est complète et correcte.

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous apprenez à faire :

- Installez le logiciel, les outils et leAWS IoTKit SDK des périphériques pour Python.
- Comprendre comment l'exemple d'application,`shadow.py`, utilise le protocole MQTT pour récupérer et mettre à jour l'état actuel de l'ombre.
- Exécutez l'exemple d'application pour Device Shadows et observez la mise à jour du document Shadow dans leAWS IoTconsole Vous avez également appris à résoudre les problèmes et à corriger les erreurs lors de l'exécution du programme.

Étapes suivantes

Vous pouvez désormais exécuter l'`shadow.py`et utilisez Device Shadows pour contrôler l'état. Vous pouvez observer les mises à jour du document Shadow dans la fenêtreAWS IoTConsole et observez les événements delta auxquels l'exemple d'application répond. À l'aide du client de test MQTT, vous pouvez vous abonner aux rubriques d'ombre réservées et observer les messages reçus par les rubriques lors de l'exécution de l'exemple de programme. Pour plus d'informations sur l'exécution de ce didacticiel, consultez.Interagissez avec Device Shadow en utilisant`shadow.py`exemple d'application et de client de test MQTT (p. 174).

Interagissez avec Device Shadow en utilisant`shadow.py`exemple d'application et de client de test MQTT

Pour interagir avec l'objet`shadow.py`, entrez une valeur dans le terminal pour l'application`desired`Valeur . Par exemple, vous pouvez spécifier des couleurs qui ressemblent aux feux de circulation etAWS IoTrépond à la demande et met à jour les valeurs signalées.

Dans ce didacticiel, vous allez apprendre à faire :

- Utilisation de l'`shadow.py`pour spécifier les états souhaités et mettre à jour l'état actuel de l'ombre.
- Modifiez le document Shadow pour observer les événements delta et la façon dont le`shadow.py`exemple d'application répond à cela.
- Utilisez le client de test MQTT pour vous abonner aux rubriques d'ombre et observer les mises à jour lorsque vous exécutez l'exemple de programme.

Avant d'exécuter ce didacticiel, vous devez disposer des éléments suivants :

Configurez votre Compte AWS , configuré votre périphérique Raspberry Pi et créé unAWS IoTchose et politique. Vous devez également avoir installé le logiciel requis, le SDK de périphérique, les fichiers de certificat et exécuter l'exemple de programme dans le terminal. Pour plus d'informations, consultez les didacticiels précédents.CréerAWS IoTet connectez Raspberry Pi pour exécuter l'application shadow (p. 163)and??? (p. 168). Vous devez suivre ces didacticiels si vous ne l'avez pas déjà fait.

Dans ce didacticiel, vous allez :

- Mettre à jour les valeurs souhaitées et signalées àshadow.pyExemple d'application (p. 175)
- Consultez les messages provenant de la pageshadow.pyExemple d'application dans le client de test MQTT (p. 176)

- Résoudre les erreurs liées aux interactions Device Shadow (p. 180)
- Passez en revue les résultats et les prochaines étapes (p. 181)

Ce didacticiel vous prendra environ 45 minutes.

Mettre à jour les valeurs souhaitées et signalées à`shadow.py`Exemple d'application

Dans le didacticiel précédent??? (p. 168), vous avez appris à observer un message publié dans le document Shadow dans leAWS IoTlorsque vous entrez une valeur souhaitée comme décrit dans la sectionInstallez le kit SDK des appareils et exécutez l'`shadow.py`exemple d'application pour Device Shadows (p. 168).

Dans l'exemple précédent, nous définissons la couleur désirée sur`yellow`. Après avoir entré chaque valeur, le terminal vous invite à entrer un autre`desired`Valeur . Si vous entrez à nouveau la même valeur (`yellow`), l'application le reconnaît et vous invite à entrer un nouveau`desired`Valeur .

```
Enter desired value:
yellow
Local value is already 'yellow'.
Enter desired value:
```

Maintenant, dites que vous entrez la couleur`green`.AWS IoTrépond à la demande et met à jour le`reported`Valeur`green`. C'est ainsi que la mise à jour se produit lorsque le`desired`est différent du type d'état`reported`état, provoquant un delta.

Procédures**`shadow.py`**exemple d'application simule les interactions Device Shadow :

1. Saisissez une`desired`Valeur (disons`yellow`) dans le terminal pour publier l'état souhaité.
2. En tant que`desired`est différent du type d'état`reported`état (dire la couleur`green`), un delta se produit et l'application qui est abonnée au delta reçoit ce message.
3. L'application répond au message et met à jour son état à la`desired`Valeur,`yellow`.
4. L'application publie ensuite un message de mise à jour avec la nouvelle valeur signalée de l'état de l'appareil,`yellow`.

La suite affiche les messages affichés dans le terminal qui montre comment la demande de mise à jour est publiée.

```
Enter desired value:
green
Changed local shadow value to 'green'.
Updating reported shadow value to 'green'...
Update request published.
Finished updating reported shadow value to 'green'.
```

DansAWS IoT, le document Shadow reflète la valeur mise à jour sur`green`pour les`reported`and`desired`, et le numéro de version est incrémenté par 1. Par exemple, si le numéro de version précédente était affiché sous la forme 10, le numéro de version actuel s'affichera sous la forme 11.

> **Note**
>
> La suppression d'une ombre ne réinitialise pas le numéro de version à 0. Vous verrez que la version instantanée est incrémentée de 1 lorsque vous publiez une demande de mise à jour ou créez une autre ombre portant le même nom.

Modifier le document Shadow pour observer les événements delta

La .`shadow.py`est également abonné à`delta`, et répond lorsqu'il y a une modification de la propriété`desired`Valeur . Par exemple, vous pouvez modifier l'objet`desired`à la couleur`red`. Pour ce faire, consultezAWS IoT, modifiez le document Shadow en cliquant surModifier, puis définissez le`desired`Valeur`red`dans le JSON, tout en conservant le`reported`Valeur`green`. Avant d'enregistrer les modifications, gardez le terminal sur le Raspberry Pi ouvert car les messages s'affichent dans le terminal lorsque la modification se produit.

```
{
  "desired": {
    "welcome": "aws-iot",
    "color": "red"
  },
  "reported": {
    "welcome": "aws-iot",
    "color": "green"
  }
}
```

Après avoir enregistré la nouvelle valeur, la propriété`shadow.py`répond à cette modification et affiche des messages dans le terminal indiquant le delta. Vous devriez alors voir les messages suivants s'afficher sous l'invite pour entrer le`desired`Valeur .

```
Enter desired value:
Received shadow delta event.
  Delta reports that desired value is 'red'. Changing local value...
Changed local shadow value to 'red'.
Updating reported shadow value to 'red'...
Finished updating reported shadow value to 'red'.
Enter desired value:
Update request published.
Finished updating reported shadow value to 'red'.
```

Consultez les messages provenant de la page`shadow.py`Exemple d'application dans le client de test MQTT

Vous pouvez utiliser l'Client de test MQTTdans leAWS IoTconsolepour surveiller les messages MQTT qui sont transmis dans votre Compte AWS . En vous abonnant aux rubriques MQTT réservées utilisées par le service Device Shadow, vous pouvez observer les messages reçus par les rubriques lors de l'exécution de l'exemple d'application.

Si vous n'avez pas déjà utilisé le client de test MQTT, vous pouvez consulterAfficher les messages MQTT de l'appareil avec le client MQTT AWS IoT (p. 62). Cela vous apprendra à utiliser l'Client de test MQTTdans leAWS IoTconsolePour afficher les messages MQTT lorsqu'ils transitent par le courtier de messages.

1. Ouvrez le client de test MQTT

   Ouverture d'Client de test MQTT dans la boîte de dialogueAWS IoTconsoledans une nouvelle fenêtre afin que vous puissiez observer les messages reçus par les rubriques MQTT sans perdre la configuration de votre client de test MQTT. Le client de test MQTT ne conserve pas d'abonnements ou de journaux de messages si vous le laissez accéder à une autre page de la console. Pour cette section du didacticiel, vous pouvez avoir le document Shadow de votreAWS IoTet le client de test MQTT s'ouvrent dans des fenêtres séparées pour observer plus facilement l'interaction avec Device Shadows.
2. Abonnez-vous aux rubriques Shadow réservées de MQTT

   Vous pouvez utiliser le client de test MQTT pour entrer les noms des rubriques réservées MQTT de Device Shadow et vous y abonner pour recevoir des mises à jour lors de l'exécution de la commande`shadow.py`Exemple d'application Pour vous abonner aux rubriques :

a. DansClient de test MQTTdans leAWS IoTconsole, choisissezS'abonner à une rubrique.

b. DansFiltre de rubriques, entrez :$aws/things/things/***thingName***/ombre/mise à jour/#. Ici,`thingname`est le nom de la ressource de objet que vous avez créée (par exemple,`My_light_bulb`).

c. Conservez les valeurs par défaut pour les autres paramètres de configuration, puis choisissezS'abonner.

En utilisant l'#dans l'abonnement de rubrique, vous pouvez vous abonner à plusieurs rubriques MQTT en même temps et observer tous les messages échangés entre l'appareil et son Shadow dans une seule fenêtre. Pour en savoir plus sur les caractères génériques et leur utilisation, consultez.Rubriques MQTT (p. 88).

3. Run (Exécuter Lambda)**`shadow.py`**exemple de programme et observer les messages

Dans la fenêtre de ligne de commande du Raspberry Pi, si vous avez déconnecté le programme, exécutez à nouveau l'exemple d'application et regardez les messages dans leClient de test MQTTdans leAWS IoTconsole.

a. Exécutez la commande suivante pour redémarrer l'exemple de programme. Remplacez*votre-iot-nom-chose*and*votre-iot-point de terminaison*avec les noms deAWS IoTCe que vous avez créé précédemment (par exemple,`My_light_bulb`) et le point de terminaison pour interagir avec le périphérique.

```
cd ~/aws-iot-device-sdk-python-v2/samples
python3 shadow.py --root-ca ~/certs/Amazon-root-CA-1.pem --cert ~/certs/
device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint --thing-
name your-iot-thing-name
```

La .`shadow.py`s'exécute et récupère l'état actuel de l'ombre. Si vous avez supprimé l'ombre ou effacé les états actuels, le programme définit la valeur actuelle sur`off`, puis vous invite à saisir un objet`desired`Valeur .

```
Connecting to a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com with client ID
 'test-0c8ae2ff-cc87-49d2-a82a-ae7ba1d0ca5a'...
Connected!
Subscribing to Delta events...
Subscribing to Update responses...
Subscribing to Get responses...
Requesting current shadow state...
Launching thread to read user input...
Finished getting initial shadow state.
  Shadow document lacks 'color' property. Setting defaults...
Changed local shadow value to 'off'.
Updating reported shadow value to 'off'...
Update request published.
Finished updating reported shadow value to 'off'...
Enter desired value:
```

D'un autre côté, si le programme était en cours d'exécution et que vous l'avez redémarré, vous verrez la dernière valeur de couleur signalée dans le terminal. Dans le client de test MQTT, vous verrez une mise à jour des rubriques$aws/things/things/***thingName***/ombre/getand$aws/things/things/***thingName***/thadow/get/accepted.

Supposons que la dernière couleur signalée soit`green`. Voici le contenu de l'$aws/things/things/***thingName***/thadow/get/acceptedFichier JSON

```
{
```

```
  "state": {
    "desired": {
      "welcome": "aws-iot",
      "color": "green"
    },
    "reported": {
      "welcome": "aws-iot",
      "color": "green"
    }
  },
  "metadata": {
    "desired": {
      "welcome": {
        "timestamp": 1620156892
      },
      "color": {
        "timestamp": 1620161643
      }
    },
    "reported": {
      "welcome": {
        "timestamp": 1620156892
      },
      "color": {
        "timestamp": 1620161643
      }
    }
  },
  "version": 10,
  "timestamp": 1620173908
}
```


b. Saisissez unedesireddans le terminal, commeyellow. La .shadow.pyrépond et affiche les messages suivants dans le terminal qui affichent la modification dans lereportedValeuryellow.

```
Enter desired value:
yellow
Changed local shadow value to 'yellow'.
Updating reported shadow value to 'yellow'...
Update request published.
Finished updating reported shadow value to 'yellow'.
```

DansClient de test MQTTdans leAWS IoTconsole, sousSubscriptions, vous voyez que les rubriques suivantes ont reçu un message :

- $aws/things/things/*thingName*/ombre/mise à jour : montre quedesiredandupdatedles valeurs changent à la couleuryellow.
- $aws/things/things/*thingName*/thadow/update/accepted : affiche les valeurs actuelles de ladesiredandreportedet leurs métadonnées et informations de version.
- $aws/things/things/*thingName*/thadow/update/documents : affiche les valeurs précédentes et actuelles de la propriétédesiredandreportedet leurs métadonnées et informations de version.

Comme le document$aws/things/things/*thingName*/thadow/update/documentscontient également de l'information qui est contenue dans les deux autres sujets, nous pouvons l'examiner pour voir l'information de l'état. L'état précédent montre la valeur rapportée définie surgreen, ses métadonnées et ses informations de version, ainsi que l'état actuel qui affiche la valeur rapportée mise à jour àyellow.

```
{
  "previous": {
    "state": {
      "desired": {
        "welcome": "aws-iot",
        "color": "green"
      },
      "reported": {
        "welcome": "aws-iot",
        "color": "green"
      }
    },
    "metadata": {
      "desired": {
        "welcome": {
          "timestamp": 1617297888
        },
        "color": {
          "timestamp": 1617297898
        }
      },
      "reported": {
        "welcome": {
          "timestamp": 1617297888
        },
        "color": {
          "timestamp": 1617297898
        }
      }
    },
    "version": 10
  },
  "current": {
    "state": {
      "desired": {
        "welcome": "aws-iot",
        "color": "yellow"
      },
      "reported": {
        "welcome": "aws-iot",
        "color": "yellow"
      }
    },
    "metadata": {
      "desired": {
        "welcome": {
          "timestamp": 1617297888
        },
        "color": {
          "timestamp": 1617297904
        }
      },
      "reported": {
        "welcome": {
          "timestamp": 1617297888
        },
        "color": {
          "timestamp": 1617297904
        }
      }
    },
    "version": 11
  },
  "timestamp": 1617297904
```

```
}
```

c. Maintenant, si vous entrez un autre`desired`, vous voyez d'autres modifications apportées à la`reported`et les mises à jour des messages reçus par ces rubriques. Le numéro de version incrémente également de 1. Par exemple, si vous entrez la valeur`green`, l'état précédent indique la valeur`yellow`et l'état actuel rapporte la valeur`green`.

4. Modifier le document Shadow pour observer les événements delta

   Pour observer les modifications apportées à la rubrique delta, modifiez le document Shadow dans la boîte de dialogueAWS IoTconsole Par exemple, vous pouvez modifier l'objet`desired`à la couleur`red`. Pour ce faire, consultezAWS IoT, choisissezModifier, puis définissez le`desired`en rouge dans le JSON, tout en conservant la propriété`reported`Valeur définie sur`green`. Avant d'enregistrer la modification, gardez le terminal ouvert car le message delta est signalé dans le terminal.

```
{
  "desired": {
    "welcome": "aws-iot",
    "color": "red"
  },
  "reported": {
    "welcome": "aws-iot",
    "color": "green"
  }
}
```

   La `.shadow.py`répond à cette modification et affiche des messages dans le terminal indiquant le delta. Dans le client de test MQTT, l'objet`update`recevront un message indiquant les modifications apportées à la`desired`and`reported`Valeurs.

   Vous voyez également que le sujet$aws/things/things/***thingName***/thadow/update/deltaa reçu un message. Pour afficher le message, choisissez cette rubrique, qui est répertoriée sousSubscriptions.

```
{
  "version": 13,
  "timestamp": 1617318480,
  "state": {
    "color": "red"
  },
  "metadata": {
    "color": {
      "timestamp": 1617318480
    }
  }
}
```

Résoudre les erreurs liées aux interactions Device Shadow

Lorsque vous exécutez l'exemple d'application Shadow, vous pouvez rencontrer des problèmes lors de l'observation des interactions avec le service Device Shadow.

Si le programme s'exécute avec succès et vous invite à saisir un`desired`, vous devez être en mesure d'observer les interactions Device Shadow à l'aide du document Shadow et du client de test MQTT comme décrit précédemment. Cependant, si vous ne parvenez pas à voir les interactions, voici quelques éléments que vous pouvez vérifier :

- Vérifiez le nom de la chose et son ombre dans laAWS IoTconsole

  Si vous ne voyez pas les messages dans le document Shadow, passez en revue la commande et assurez-vous qu'elle correspond au nom de la chose dans leAWS IoTconsole. Vous pouvez

également vérifier si vous avez une ombre classique en choisissant votre ressource de truc, puis en choisissantShadows. Ce tutoriel se concentre principalement sur les interactions avec l'ombre classique.

Vous pouvez également confirmer que l'appareil que vous avez utilisé est connecté à Internet. DansAWS IoTconsole, choisissez l'objet que vous avez créé précédemment, puis choisissezInteragir. Sur la page de détails de l'objet, vous devriez voir un message ici indiquant :`This thing already appears to be connected.`

- Consultez les rubriques réservées de MQTT auxquelles vous vous êtes abonné

  Si les messages ne s'affichent pas dans le client de test MQTT, vérifiez si les rubriques auxquelles vous vous êtes abonné sont correctement formatées. Les rubriques MQTT Device Shadow ont un format$aws/things/things/***`thingName`***/shadow/et pourrait avoir`update`,`get`, ou`delete`En fonction des actions que vous souhaitez effectuer sur l'ombre. Ce didacticiel utilise.$aws/things/things/***`thingName`***/ombre/ #donc assurez-vous de l'avoir correctement saisi lorsque vous vous abonnez à la rubrique dans laFiltre de rubriquesdu client de test.

  Lorsque vous entrez le nom de la rubrique, assurez-vous que la propriété*`thingName`*est le même que le nom duAWS IoTCe que vous avez créé plus tôt. Vous pouvez également vous abonner à d'autres rubriques MQTT pour voir si une mise à jour a été effectuée avec succès. Par exemple, vous pouvez vous abonner à la rubrique$aws/things/things/***`thingName`***/thadow/update/acceptedpour recevoir un message chaque fois qu'une demande de mise à jour a échoué afin que vous puissiez déboguer des problèmes de connexion. Pour plus d'informations sur les rubriques réservées, consultez.the section called “Rubriques de shadow” (p. 103)andRubriques MQTT de Device Shadow (p. 577).

Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous apprenez à faire :

- Utilisation de l'`shadow.py`pour spécifier les états souhaités et mettre à jour l'état actuel de l'ombre.
- Modifiez le document Shadow pour observer les événements delta et la façon dont le`shadow.py`exemple d'application répond à cela.
- Utilisez le client de test MQTT pour vous abonner aux rubriques d'ombre et observer les mises à jour lorsque vous exécutez l'exemple de programme.

Étapes suivantes

Vous pouvez vous abonner à d'autres rubriques réservées MQTT pour observer les mises à jour de l'application shadow. Par exemple, si vous vous abonnez uniquement à la rubrique$aws/things/things/***`thingName`***/thadow/update/accepted, vous verrez uniquement les informations d'état actuellorsqu'une mise à jour est exécutée avec succès.

Vous pouvez également vous abonner à d'autres rubriques d'ombre pour déboguer des problèmes ou en savoir plus sur les interactions de l'Ombre de l'appareil et également déboguer tous les problèmes liés aux interactions de l'Ombre de l'appareil. Pour plus d'informations, consultez the section called “Rubriques de shadow” (p. 103) et Rubriques MQTT de Device Shadow (p. 577).

Vous pouvez également choisir d'étendre votre application en utilisant des ombres nommées ou en utilisant du matériel supplémentaire connecté au Raspberry Pi pour les LED et observer les changements de leur état à l'aide des messages envoyés depuis le terminal.

Pour plus d'informations sur le service Device Shadow et son utilisation dans les appareils, les applications et les services, consultezService AWS IoT Device Shadow (p. 547),Utilisation des shadows sur les appareils (p. 551), etUtilisation des shadows dans les applications et les services (p. 554).

Autres didacticiels AWS IoT

Les didacticiels de cette section vous montrent comment utiliser plusieurs services AWS IoT pour effectuer une tâche. Ces didacticiels sont davantage axés sur l'intégration des services que sur un examen approfondi des fonctions de l'AWS IoT. Les didacticiels de cette section peuvent s'appuyer les uns sur les autres pour montrer comment vous pouvez commencer modestement et faire évoluer vos solutions à mesure que les besoins de votre entreprise évoluent ou se développent.

Chaque didacticiel inclut une liste des pré-requis, y compris le matériel spécifique. Dans la mesure du possible, les didacticiels proposent des solutions alternatives si vous ne possédez pas tout le matériel requis.

Créez un mécanisme d'autorisation personnalisé pour AWS IoT Core

Ce didacticiel illustre les étapes à suivre pour créer, valider et utiliser l'authentification personnalisée à l'aide de l'outilAWS CLI. En option, à l'aide de ce didacticiel, vous pouvez utiliser Postman pour envoyer des données à AWS IoT Core à l'aide de l'API de publication HTTP.

Ce tutoriel vous montre comment créer un exemple de fonction Lambda qui implémente la logique d'autorisation et d'authentification et un autorisateur personnalisé à l'aide de l'create-authorizeravec la signature de jeton activée. L'autorisateur est ensuite validé à l'aide de latest-invoke-authorizer, et enfin vous pouvez envoyer des données à AWS IoT Core à l'aide de l'API de publication HTTP à une rubrique MQTT de test. Exemple de demande spécifiera l'autorisation à invoquer à l'aide de la méthode`x-amz-customauthorizer-name`et passez le nom de la clé de jeton et`x-amz-customauthorizer-signature`Dans les en-têtes de requête.

Ce que vous apprendrez dans ce tutoriel :

- Comment créer une fonction Lambda pour être un gestionnaire d'autorisations personnalisé
- Comment créer un mécanisme d'autorisation personnalisé à l'aide de l'outilAWS CLIavec la signature de jeton activée
- Comment tester votre autorisation personnalisée à l'aide de l'outiltest-invoke-authorizerCommande de l'
- Comment publier une rubrique MQTT à l'aide dePostmanet validez la demande avec votre autorisation personnalisée

Ce didacticiel vous prendra environ 60 minutes.

Dans ce didacticiel, vous allez :

- Créez une fonction Lambda pour votre mécanisme d'autorisation personnalisée (p. 183)
- Créez une key pair publique et privée pour votre mécanisme d'autorisation personnalisée (p. 186)
- Créer une ressource d'autorisation client et son autorisation (p. 186)
- Testez l'mécanisme d'autorisation en appelant test-invoke-authorizer (p. 189)
- Tester la publication du message MQTT à l'aide de Postman (p. 190)
- Afficher les messages dans le client de test MQTT (p. 192)
- Examiner les résultats et les prochaines étapes (p. 192)
- Nettoyage après l'exécution de ce tutoriel (p. 193)

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- Configurer votre Compte AWS (p. 19)

Vous aurez besoin de vos Compte AWS andAWS IoTPour suivre ce didacticiel.

Le compte que vous utilisez pour ce tutoriel fonctionne mieux lorsqu'il inclut au moins cesAWSStratégies gérées :

- `IAMFullAccess`
- `AWSIoTFullAccess`
- `AWSLambda_FullAccess`

  Important

  Les stratégies IAM utilisées dans ce didacticiel sont plus permissives que vous ne devriez suivre dans une implémentation de production. Dans un environnement de production, assurez-vous que vos stratégies de compte et de ressources n'accordent que les autorisations nécessaires. Lorsque vous créez des stratégies IAM pour la production, déterminez les utilisateurs d'accès et les rôles dont vous avez besoin, puis concevez les stratégies leur permettant de réaliser uniquement ces tâches.
  Pour de plus amples informations, veuillez consulterBonnes pratiques de sécurité dans IAM

- Installation duAWS CLI

  Pour de plus amples informations sur l'installation duAWS CLI, voirInstallation du kitAWSCLI. Ce didacticiel nécessiteAWS CLIversionaws-cli/2.1.3 Python/3.7.4 Darwin/18.7.0 exe/x86_64ou version ultérieure.

- Outils OpenSSL

  Les exemples de ce tutoriel utilisentLibreSSL 2.6.5. Vous pouvez également utiliserOpenSSL v1.1.1ipour ce didacticiel.

- Vérifiez lesAWS LambdaPrésentation de

  Si vous n'avez pas utiliséAWS Lambdaavant, révisionAWS LambdaandMise en route avec LambdaPour connaître ses termes et concepts.

- Examen de la façon de créer des requêtes dans Postman

  Pour de plus amples informations, veuillez consulterDemandes de construction.

- Retrait des autorisations personnalisées du tuteur précédent

  Vos Compte AWS ne peut avoir qu'un nombre limité d'autorisations personnalisées configurées à la fois. Pour plus d'informations sur la suppression d'un mécanisme d'autorisation personnalisé, consultezthe section called “Nettoyage après l'exécution de ce tutoriel” (p. 193).

Créez une fonction Lambda pour votre mécanisme d'autorisation personnalisée

Authentification personnalisée dans AWS IoT Core utilise le traitementressources d'autorisationque vous créez pour authentifier et autoriser les clients. La fonction que vous allez créer dans cette section authentifie et autorise les clients lorsqu'ils se connectent à AWS IoT Core et accès àAWS IoTAWS.

La fonction Lambda effectue les opérations suivantes :

- Si une demande provient detest-invoke-authorizer, il renvoie une stratégie IAM avec un`Deny`action.
- Si une demande provient de Passport en utilisant HTTP et le`actionToken`a une valeur de`allow`, il renvoie une stratégie IAM avec un`Allow`action. Dans le cas contraire, elle renvoie une stratégie IAM avec un`Deny`action.

Pour créer la fonction Lambda pour votre mécanisme d'autorisation personnalisée :

1. DansLambdaOuvrez la consoleFonctions.
2. Sélectionnez Créer une fonction.
3. ConfirmerCréer à partir de zéroest sélectionné.
4. Sous Basic information :
   a. Sous Nom de la fonction, entrez **`custom-auth-function`**.
   b. DansRuntime (Exécution), confirmezNode.js 14.x
5. Sélectionnez Créer une fonction.

   Lambda crée une fonction Node.js et unRôle d'exécutionqui accorde à la fonction l'autorisation de télécharger des journaux. La fonction Lambda endosse le rôle d'exécution lorsque vous appelez votre fonction, et l'utilise pour créer des informations d'identification pour la fonctionAWSSDK et pour lire les données à partir de sources d'événements.
6. Pour voir le code et la configuration de la fonction dans le champAWS Cloud9 éditeur, choisissezfonction custom-auth-Dans la fenêtre du concepteur, puis choisissezindex.jsDans le volet de navigation de l'éditeur.

   Pour les langages de script tels que Node.js, Lambda inclut une fonction de base qui renvoie une réponse de réussite. Vous pouvez utiliser laAWS Cloud9Editez votre fonction aussi longtemps que votre code source ne dépasse pas 3 Mo.
7. Remplacez leindex.jsDans l'éditeur avec le code suivant :

```
// A simple Lambda function for an authorizer. It demonstrates
// How to parse a CLI and Http password to generate a response.

exports.handler = function(event, context, callback) {

    //Http parameter to initiate allow/deny request
    const HTTP_PARAM_NAME='actionToken';
    const ALLOW_ACTION = 'Allow';
    const DENY_ACTION = 'Deny';

    //Event data passed to Lambda function
    var event_str = JSON.stringify(event);
    console.log('Complete event :'+ event_str);

    //Read protocolData from the event json passed to Lambda function
    var protocolData = event.protocolData;
    console.log('protocolData value---> ' + protocolData);

    //Get the dynamic account ID from function's ARN to be used
    // as full resource for IAM policy
    var ACCOUNT_ID = context.invokedFunctionArn.split(":")[4];
    console.log("ACCOUNT_ID---"+ACCOUNT_ID);

    //protocolData data will be undefined if testing is done via CLI.
    // This will help to test the set up.
    if (protocolData === undefined) {

        //If CLI testing, pass deny action as this is for testing purpose only.
        console.log('Using the test-invoke-authorizer cli for testing only');
        callback(null, generateAuthResponse(DENY_ACTION,ACCOUNT_ID));

    } else{

        //Http Testing from Postman
        //Get the query string from the request
        var queryString = event.protocolData.http.queryString;
```

```
        console.log('queryString values -- ' + queryString);
        /*         global URLSearchParams       */
        const params = new URLSearchParams(queryString);
        var action = params.get(HTTP_PARAM_NAME);

        if(action!=null && action.toLowerCase() === 'allow'){

            callback(null, generateAuthResponse(ALLOW_ACTION,ACCOUNT_ID));

        }else{

            callback(null, generateAuthResponse(DENY_ACTION,ACCOUNT_ID));

        }

    }

};

// Helper function to generate the authorization IAM response.
var generateAuthResponse = function(effect,ACCOUNT_ID) {

    var full_resource = "arn:aws:iot:us-west-2:"+ACCOUNT_ID+":*";

    var authResponse = {};
    authResponse.isAuthenticated = true;
    authResponse.principalId = 'principalId';

    var policyDocument = {};
    policyDocument.Version = '2012-10-17';
    policyDocument.Statement = [];
    var statement = {};
    statement.Action = 'iot:*';
    statement.Effect = effect;
    statement.Resource = full_resource;
    policyDocument.Statement[0] = statement;
    authResponse.policyDocuments = [policyDocument];
    authResponse.disconnectAfterInSeconds = 3600;
    authResponse.refreshAfterInSeconds = 600;

    console.log('custom auth policy function called from http');
    console.log('authResponse --> ' + JSON.stringify(authResponse));
    console.log(authResponse.policyDocuments[0]);

    return authResponse;
}
```

8. Choisissez Deploy (Déployer).
9. AprèsModifications déployéesapparaît au-dessus de l'éditeur :
   a. Faites défiler jusqu'à laPrésentation de la fonctionau-dessus de l'éditeur.
   b. Copiez laARN de fonctionet l'enregistrer pour l'utiliser plus loin dans ce didacticiel.
10. Testez votre fonction .
    a. Cliquez sur l'ongletTestOnglet.
    b. À l'aide des paramètres de test par défaut, choisissezInvoquer.
    c. Si le test a réussi, dans la boîte de dialogueRésultats d'exécution, ouvrezDétailsAffichage. Vous devriez voir le document de stratégie que la fonction a renvoyé.

       Si le test a échoué ou si vous ne voyez pas de document de stratégie, vérifiez le code pour trouver et corriger les erreurs.

Créez une key pair publique et privée pour votre mécanisme d'autorisation personnalisée

Votre autorisation personnalisée nécessite une clé publique et privée pour l'authentifier. Les commandes de cette section utilisent les outils openssl pour créer cette key pair.

Pour créer la key pair publiques et privées pour votre autorisation personnalisée :

1. Créez le fichier de clé privée.

```
openssl genrsa -out private-key.pem 4096
```

2. Vérifiez le fichier de clé privée que vous venez de créer.

```
openssl rsa -check -in private-key.pem -noout
```

   Si la commande n'affiche aucune erreur, le fichier de clé privée est valide.

3. Créez le fichier de clé publique.

```
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

4. Vérifiez le fichier de clé publique.

```
openssl pkey -inform PEM -pubin -in public-key.pem -noout
```

   Si la commande n'affiche aucune erreur, le fichier de clé publique est valide.

Créer une ressource d'autorisation client et son autorisation

La .AWS IoTest la ressource qui relie tous les éléments créés dans les étapes précédentes. Dans cette section, vous allez créer une ressource d'autorisation personnalisée et lui donner l'autorisation d'exécuter la fonction Lambda que vous avez créée précédemment. Vous pouvez créer une ressource d'autorisation personnalisée à l'aide de l'outilAWS IoT, la consoleAWS CLI, ou leAWSAPI.

Pour ce didacticiel, vous ne devez créer qu'un mécanisme d'autorisation personnalisé. Cette section explique comment créer à l'aide de l'outilAWS IoTet la consoleAWS CLI, de sorte que vous pouvez utiliser la méthode la plus pratique pour vous. Il n'y a pas de différence entre les ressources d'autorisation personnalisées créées par l'une ou l'autre des méthodes.

Créer une ressource d'autorisation client

Choisissez l'une des options suivantes pour créer votre ressource d'autorisation personnalisée :

- Créez un mécanisme d'autorisation personnalisé à l'aide duAWS IoTconsole (p. 186)
- Créez un mécanisme d'autorisation personnalisé à l'aide du AWS CLI (p. 187)

Pour créer un mécanisme d'autorisation personnalisée à l'aide duAWS IoTConsole :

1. Ouverture d'Page d'autorisation personnalisée duAWS IoTconsole, puis choisissezCréer.
2. DansCréation d'un mécanisme d'autorisation personnalisée :
   a. DansNommez votre mécanisme d'autorisation personnalisée, saisissez**`my-new-authorizer`**.
   b. DansFonction d'autorisationChoisissez la fonction Lambda que vous avez créée précédemment.

c. DansValidation des jetons - facultatif :

i. CheckActiver la signature de jeton.

ii. DansNom d'en-tête de jeton (facultatif), saisissez**`tokenKeyName`**.

iii. DansNom de la touche, entrez :**`FirstKey`**.

iv. DansValeur, entrez le contenu du`public-key.pem`dans le fichier. Veillez à inclure les lignes du fichier avec`-----BEGIN PUBLIC KEY-----`and`-----END PUBLIC KEY-----`et n'ajoutez ni ne supprimez aucun fil de ligne, retour chariot ou autres caractères du contenu du fichier. La chaîne que vous entrez devrait ressembler à cet exemple.

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvEBzOk4vhN+3LgslvEWt
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csalS/Rk4phD5
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshtT/F1LVCS5+v8AQ8UGGDfZmv
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sq8Lgzpey9N5ON1o1Cvldwvc
KrJJtgwW6hVqRGuShnownLpgG86M6neZ5sRMbVNZO8OzcobLngJ0Ibw9KkcUdklW
gvZ6HEJqBY2XE70iEXAMPLETPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa
/NN7l7Zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJlUzn62Q+VeNV2tdA7MfPfMC
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zLMtF+kFl2yOBmGAP0RBivRd9
JWBUCG0bqcLQPeQyjbXSOfUCAwEAAQ==
-----END PUBLIC KEY-----
```

3. CheckActiver l'mécanisme d'autorisation.
4. ChoisissezCrée un mécanisme d'autorisation.
5. Si la ressource d'autorisation personnalisée a été créée, vous verrez la liste des autorisateurs personnalisés et votre nouvel autorisateur personnalisé doit apparaître dans la liste et vous pouvez continuer à la section suivante pour le tester.

   Si vous voyez une erreur, vérifiez l'erreur et essayez de créer à nouveau votre autorisation personnalisée et vérifiez les entrées. Notez que chaque ressource d'autorisation personnalisée doit avoir un nom unique.

Pour créer un autorisateur personnalisé à l'aide de l'outilAWS CLI :

1. Remplacez vos valeurs par`authorizer-function-arn`and`token-signing-public-keys`, puis exécutez la commande suivante :

```
aws iot create-authorizer \
--authorizer-name "my-new-authorizer" \
--token-key-name "tokenKeyName" \
--status ACTIVE \
--no-signing-disabled \
--authorizer-function-arn "arn:aws:lambda:us-west-2:57EXAMPLE833:function:custom-auth-
function" \
--token-signing-public-keys FirstKey="-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAvEBzOk4vhN+3LgslvEWt
sLCqNmt5Damas3bmiTRvq2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z
QPu/vQOe5tyzz1MsKdmtFGxMqQ3qjEXAMPLEOmqyUKPP5mff58k6ePSfXAnzBH0q
lg2HioefrpU5OSAnpuRAjYKofKjbc2Vrn6N2G7hV+IfTBvCElf0csalS/Rk4phD5
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshtT/F1LVCS5+v8AQ8UGGDfZmv
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2Sq8Lgzpey9N5ON1o1Cvldwvc
KrJJtgwW6hVqRGuShnownLpgG86M6neZ5sRMbVNZO8OzcobLngJ0Ibw9KkcUdklW
gvZ6HEJqBY2XE70iEXAMPLETPHzhqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa
/NN7l7Zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJlUzn62Q+VeNV2tdA7MfPfMC
8btGYladFAnitThaz6+F0VSBJPu7pZQoLnqyEp5zLMtF+kFl2yOBmGAP0RBivRd9
```

```
JWBUCG0bqcLQPeQyjbXSOfUCAwEAAQ==
-----END PUBLIC KEY-----"
```

Où :

- La `.authorizer-function-arn`valeur est le nom de ressource Amazon (ARN) de la fonction Lambda que vous avez créée pour votre mécanisme d'autorisation personnalisée.
- La `.token-signing-public-keys`inclut le nom de la clé,**`FirstKey`**, et le contenu du`public-key.pem`dans le fichier. Veillez à inclure les lignes du fichier avec`-----BEGIN PUBLIC KEY-----`and`-----END PUBLIC KEY-----`et n'ajoutez ni ne supprimez aucun fil de ligne, retour chariot ou autre caractère du contenu du fichier.

  Remarque : soyez très prudent en entrant la clé publique car toute modification de la valeur de la clé publique la rend inutilisable.

2. Si l'autorisation personnalisée est créée, la commande renvoie le nom et l'ARN de la nouvelle ressource, comme le suivant.

```
{
    "authorizerName": "my-new-authorizer",
    "authorizerArn": "arn:aws:iot:us-west-2:57EXAMPLE833:authorizer/my-new-authorizer"
}
```

Enregistrer le`authorizerArn`Pour l'utiliser à l'étape suivante.

N'oubliez pas que chaque ressource d'autorisation personnalisée doit avoir un nom unique.

Autoriser la ressource d'autorisation personnalisée

Dans cette section, vous allez accorder l'autorisation à la ressource d'autorisation personnalisée que vous venez de créer l'autorisation d'exécuter la fonction Lambda

Accorder l'autorisation à votre fonction Lambda à l'aide duAWS CLI

1. Après avoir inséré vos valeurs, entrez la commande suivante. Remarque :`statement-id`doit être unique. Remplacez*`Id-1234`*avec une autre valeur si vous avez déjà exécuté ce tutoriel ou si vous obtenez un`ResourceConflictException`erreur.

```
aws lambda add-permission  \
--function-name "custom-auth-function" \
--principal "iot.amazonaws.com" \
--action "lambda:InvokeFunction" \
--statement-id "Id-1234" \
--source-arn authorizerArn
```

2. Si la commande réussit, elle renvoie une instruction d'autorisation, telle que cet exemple. Vous pouvez passer à la section suivante pour tester l'autorisation personnalisée.

```
{
    "Statement": "{\"Sid\":\"Id-1234\",\"Effect\":\"Allow\",\"Principal\":
{\"Service\":\"iot.amazonaws.com\"},\"Action\":\"lambda:InvokeFunction\",
\"Resource\":\"arn:aws:lambda:us-west-2:57EXAMPLE833:function:custom-auth-
function\",\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:lambda:us-
west-2:57EXAMPLE833:function:custom-auth-function\"}}}"
}
```

Si la commande ne réussit pas, elle renvoie une erreur, comme cet exemple. Vous devrez vérifier et corriger l'erreur avant de continuer.

```
An error occurred (AccessDeniedException) when calling the AddPermission operation:
 User: arn:aws:iam::57EXAMPLE833:user/EXAMPLE-1 is not authorized to perform:
 lambda:AddPer
mission on resource: arn:aws:lambda:us-west-2:57EXAMPLE833:function:custom-auth-
function
```

Testez l'mécanisme d'autorisation en appelant test-invoke-authorizer

Avec toutes les ressources définies, dans cette section, vous appellerez test-invoke-authorizer à partir de la ligne de commande pour tester la passe d'autorisation.

Notez que lorsque vous appelez l'mécanisme d'autorisation à partir de la ligne de commande,`protocolData`n'est pas défini, de sorte que l'autorisateur retournera toujours un document DENY. Ce test confirme cependant que votre autorisation personnalisée et la fonction Lambda sont configurés correctement, même s'il ne teste pas complètement la fonction Lambda.

Pour tester votre autorisateur personnalisé et sa fonction Lambda à l'aide de l'outilAWS CLI :

1. Dans le répertoire qui contient le`private-key.pem`Vous avez créé à l'étape précédente, exécutez la commande suivante.

   ```
   echo -n "tokenKeyValue" | openssl dgst -sha256 -sign private-key.pem | openssl base64 -
   A
   ```

   Cette commande crée une chaîne de signature à utiliser à l'étape suivante. La chaîne de signature doit se présenter comme suit :

   ```
   dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr+rbCvmu9Jl4KHAA9DG+V
   +MMWu09YSA86+64Y3Gt4tOykpZqn9mn
   VB1wyxp+0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNqBZzbCvsluv7i2IMjEg
   +CPY0zrWt1jr9BikgGPDxWkjaeeh
   bQHHTo357TegKs9pP30Uf4TrxypNmFswA5k7QIc01n4bIyRTm90OyZ94R4bdJsHNig1JePgnuOBvMGCEFE09jGjj
   szEHfgAUAQIWXiVGQj16BU1xKpTGSiTAwheLKUjITOEXAMPLECK3aHKYKY
   +d1vTvdthKtYHBq8MjhzJ0kggbt29V
   QJCb8RilN/P5+vcVniSXWPplyB5jkYs9UvG08REoy64AtizfUhvSul/r/F3VV8ITtQp3aXiUtcspACi6ca
   +tsDuX
   f3LzCwQQF/YSUy02u5XkWn+sto6KCkpNlkD0wU8gl3+kOzxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o
   +K
   EWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPLELotyh7h
   +flFeloZlAWQFH
   xRlXsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
   ```

   Copiez cette chaîne de signature pour l'utiliser à l'étape suivante. Veillez à ne pas inclure de caractères supplémentaires ou à en laisser de côté.
2. Dans cette commande, remplacez la`token-signature`avec la chaîne de signature de l'étape précédente et exécutez cette commande pour tester votre autorisation.

   ```
   aws iot test-invoke-authorizer \
   --authorizer-name my-new-authorizer \
   --token tokenKeyValue \
   --token-signature dBwykzlb+fo+JmSGdwoGr8dyC2qB/IyLefJJr
   +rbCvmu9Jl4KHAA9DG+V+MMWu09YSA86+64Y3Gt4tOykpZqn9mnVB1wyxp
   +0bDZh8hmqUAUH3fwi3fPjBvCa4cwNuLQNqBZzbCvsluv7i2IMjEg
   +CPY0zrWt1jr9BikgGPDxWkjaeehbQHHTo357TegKs9pP30Uf4TrxypNmFswA5k7QIc01n4bIyRTm90OyZ94R4bdJsHNig1JePg
   +d1vTvdthKtYHBq8MjhzJ0kggbt29VQJCb8RilN/P5+vcVniSXWPplyB5jkYs9UvG08REoy64AtizfUhvSul/
   ```

```
r/F3VV8ITtQp3aXiUtcspACi6ca+tsDuXf3LzCwQQF/YSUy02u5XkWn
+sto6KCkpNlkD0wU8gl3+kOzxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o
+KEWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPLELotyh7h
+flFeloZlAWQFHxRlXsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Si la commande réussit, elle renvoie les informations générées par votre fonction d'autorisation client, comme cet exemple.

```
{
    "isAuthenticated": true,
    "principalId": "principalId",
    "policyDocuments": [
        "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Action\":\"iot:*\",\"Effect\":
\"Deny\",\"Resource\":\"arn:aws:iot:us-west-2:57EXAMPLE833:*\"}]}"
    ],
    "refreshAfterInSeconds": 600,
    "disconnectAfterInSeconds": 3600
}
```

Si la commande renvoie une erreur, vérifiez l'erreur et vérifiez deux fois les commandes que vous avez utilisées dans cette section.

Tester la publication du message MQTT à l'aide de Postman

1. Pour obtenir le point de terminaison de données de votre appareil à partir de la ligne de commande, appelezdescripbe-endpointcomme indiqué ici

   ```
   aws iot describe-endpoint --output text --endpoint-type iot:Data-ATS
   ```

   Enregistrez cette adresse pour l'utiliser comme*device_data_endpoint_address*dans une étape ultérieure.
2. Ouvrez une nouvelle fenêtre Postman et créez une nouvelle requête HTTP POST.

   a. Ouvrez l'application Postman.

   b. Dans Postman, dans leFichier, choisissezNouvelle....

   c. DansNouvelle, choisissezRequête.

   d. Dans la demande de sauvegarde,

      i. DansNom de la requêtesaisissez**`Custom authorizer test request`**.

      ii. DansSélectionnez une collection ou un dossier dans lequel enregistrer :choisir ou créer une collection dans laquelle enregistrer cette demande.

      iii. ChoisissezEnregistrer dans***`nom_collection`***.
3. Créez la requête POST pour tester votre autorisation personnalisée.

   a. Dans le sélecteur de méthode de requête en regard du champ URL, choisissezPOST.

   b. Dans le champ URL, créez l'URL de votre demande à l'aide de l'URL suivante avec l'option*device_data_endpoint_address*à partir desdescripbe-endpointdans une étape précédente.

      ```
      https://device_data_endpoint_address:443/topics/test/cust-auth/topic?
      qos=0&actionToken=allow
      ```

Notez que cette URL inclut le`actionToken=allow`qui indiquera à votre fonction Lambda de retourner un document de stratégie qui permet d'accéder àAWS IoT. Une fois que vous avez entré l'URL, les paramètres de requête apparaissent également dans la zoneparamsde Postman.

c. DansAut., dans l'ongletType, choisissezPas d'Auth.

d. Dans l'onglet Headers :

i. S'il y a unHostqui est cochée, décochez celle-ci.

ii. Au bas de la liste des en-têtes, ajoutez ces nouveaux en-têtes et confirmez qu'ils sont cochés. Remplacez le**`Host`**avec votre*device_data_endpoint_address*et l'**`x-amz-customauthorizer-signature`**avec la chaîne de signature que vous avez utilisée avec latest-invoke-authorizeDans la section précédente.

Key	Valeur
`x-amz-customauthorizer-name`	**`my-new-authorizer`**
`Host`	*device_data_endpoint_address*
`tokenKeyName`	**`tokenKeyValue`**
`x-amz-customauthorizer-signature`	*dbwykzlb+fo+jmsgdwogr8dyc2qb/iylefjr+rbcvmu9jl4khaaa9dg+v+mmwu09ysa86+64y3gt4toykpzqn9mnvb1wyxp+0bdzh8hmquauh3fwi3fpjbv4cwN4cwN4cWykvvvvvvv+CPY0ZRWT1JR9BIKGGPDXWKJAEHBQHTO357TEGKS9PP3GJJSZEHFGAUAQIWXIVGQJ16Bu1xkptgsitaWhelkujitoExampleck3ahkyky+d1vtvdthktyhbq8mjhzj0kggbt29vqjcb8riln/p5+vcvnisxwpplyb5jkys9uvg08reoy64atizfuhvsulr/f3vvv8ittqp3axiutp3axiutututututuututtp3axiutu+tsduxf3lzcwqqf/ysuy02u5xkwn+sto6kckpnlkd0wu8gl3+kozxrthnq8geajd5iylx230+kewckte91i1mokdr5sj4jxivxxxxxxxxxxivxxxxxxxxh+flfelozlawqfhxrlxspqivks1ziuclazwprh/ordjplpiwfbgbiogokjidgp9gwhxiik7zwrgmwpmk9o=*

e. Dans l'onglet Corps :

i. Dans la zone d'option Format des données, choisissezRaw.

ii. Dans la liste des types de données, choisissezJavaScript.

iii. Dans le champ de texte, entrez cette charge utile de message JSON pour votre message de test :

```
{
    "data_mode": "test",
    "vibration": 200,
    "temperature": 40
}
```

4. ChoisissezEnvoyerPour envoyer la demande.

Si la demande aboutit, elle renvoie :

```
{
```

```
    "message": "OK",
    "traceId": "ff35c33f-409a-ea90-b06f-fbEXAMPLE25c"
}
```

La réponse réussie indique que votre autorisation personnalisée a autorisé la connexion àAWS IoTet que le message de test a été remis au courtier dans AWS IoT Core .

S'il renvoie une erreur, vérifiez le message d'erreur, le*device_data_endpoint_address*, la chaîne de signature et les autres valeurs d'en-tête.

Conservez cette requête dans Postman pour l'utiliser dans la section suivante.

Afficher les messages dans le client de test MQTT

Au cours de l'étape précédente, vous avez envoyé des messages de périphérique simulés àAWS IoTEn utilisant Postman. La réponse réussie indiquait que votre autorisation personnalisée autorisait la connexion àAWS IoTet que le message de test a été remis au courtier dans AWS IoT Core . Dans cette section, vous allez utiliser le client de test MQTT dans le champAWS IoTpour voir le contenu du message de ce message comme d'autres périphériques et services peuvent le faire.

Pour voir les messages de test autorisés par votre autorisation personnalisée :

1. DansAWS IoTOuvrez la consoleClient de test MQTT.
2. DansS'abonner à la rubrique, dansFiltre de rubriques, saisissez**`test/cust-auth/topic`**, qui est la rubrique du message utilisée dans l'exemple Postman de la section précédente.
3. Choisissez Subscribe.

   Gardez cette fenêtre visible pour l'étape suivante.
4. Dans Postman, dans la demande que vous avez créée pour la section précédente, choisissezEnvoyer.

   Passez en revue la réponse pour vous assurer qu'elle a réussi. Si ce n'est pas le cas, résolvez l'erreur comme décrit la section précédente.
5. DansClient de test MQTT, vous devriez voir une nouvelle entrée qui affiche la rubrique du message et, si elle est étendue, la charge utile du message provenant de la requête que vous avez envoyée à partir de Postman.

   Si vous ne voyez pas vos messages dans laClient de test MQTT, voici quelques points à vérifier :

   - Assurez-vous que votre demande Postman a été renvoyée avec succès. SiAWS IoTrejette la connexion et renvoie une erreur, le message de la requête n'est pas transmis au courtier de messages.
   - Vérifiez les éléments Compte AWS and Région AWS utilisé pour ouvrirAWS IoTsont les mêmes que vous utilisez dans l'URL Postman.
   - Assurez-vous que vous avez entré la rubrique correctement dans la fenêtreClient de test MQTT. Le filtre de rubrique est sensible à la casse. En cas de doute, vous pouvez également vous abonner au**#**, qui s'abonne à tous les messages MQTT qui transitent par le courtier de messages le Compte AWS and Région AWS utilisé pour ouvrirAWS IoTconsole

Examiner les résultats et les prochaines étapes

Dans ce tutoriel :

- Vous avez créé une fonction Lambda pour être un gestionnaire d'autorisations personnalisé
- Vous avez créé un autorisateur personnalisé avec la signature de jeton activée

- Vous avez testé votre autorisation personnalisée à l'aide dutest-invoke-authorizerCommande de l'
- Vous avez publié une rubrique MQTT en utilisantPostmanet validez la demande avec votre autorisation personnalisée
- Vous avez utilisé leClient de test MQTTpour voir les messages envoyés à partir de votre test Postman

Étapes suivantes

Après avoir envoyé des messages de Postman pour vérifier que l'autorisation personnalisée fonctionne, essayez d'expérimenter pour voir comment la modification des différents aspects de ce didacticiel affecte les résultats. Voici quelques exemples pour commencer.

- Modifiez la chaîne de signature afin qu'il ne soit plus valide pour voir comment les tentatives de connexion non autorisées sont gérées. Vous devriez obtenir une réponse d'erreur, telle que celle-ci, et le message ne doit pas apparaître dans laClient de test MQTT.

```
{
    "message": "Forbidden",
    "traceId": "15969756-a4a4-917c-b47a-5433e25b1356"
}
```

- Pour en savoir plus sur la recherche d'erreurs qui peuvent se produire lors du développement et de l'utilisationAWS IoTrègles, voirSurveillance de AWS IoT (p. 352).

Nettoyage après l'exécution de ce tutoriel

Si vous souhaitez répéter ce tutoriel, vous devrez peut-être supprimer certains de vos autorisateurs personnalisés. Vos Compte AWS ne peut avoir qu'un nombre limité d'autorisations personnalisées configurées à la fois et vous pouvez obtenir un`LimitExceededException`lorsque vous essayez d'en ajouter un nouveau sans supprimer un autorisateur personnalisé existant.

Pour supprimer un autorisateur personnalisé à l'aide de la commandeAWS IoTconsole

1. Ouverture d'Page d'autorisation personnalisée duAWS IoTconsole, et dans la liste des autorisations personnalisées, recherchez l'autorisation personnalisée à supprimer.
2. Ouvrez la page Détails de l'autorisation personnalisée et, à partir duActions, choisissezModifier.
3. Décochez la caseActiver l'mécanisme d'autorisation, puis choisissezMise à jour.

   Vous ne pouvez pas supprimer un mécanisme d'autorisation personnalisé tant qu'il est actif.
4. À partir de la page Détails de l'autorisation personnalisée, ouvrez leActions, puis choisissezSupprimer.

Pour supprimer un autorisateur personnalisé à l'aide de la commandeAWS CLI

1. Dressez la liste des autorisations personnalisées que vous avez installées et recherchez le nom de l'mécanisme d'autorisation personnalisée que vous souhaitez supprimer.

   ```
   aws iot list-authorizers
   ```
2. Définissez l'mécanisme d'autorisation personnalisée sur`inactive`en exécutant cette commande après avoir remplacé*`Custom_Auth_Name`*avec le`authorizerName`de l'mécanisme d'autorisation personnalisée à supprimer.

   ```
   aws iot update-authorizer --status INACTIVE --authorizer-name Custom_Auth_Name
   ```

3. Supprimez l'autorisation personnalisée en exécutant cette commande après avoir remplacé*Custom_Auth_Name*avec le`authorizerName`de l'mécanisme d'autorisation personnalisée à supprimer.

```
aws iot delete-authorizer --authorizer-name Custom_Auth_Name
```

Surveillance de l'humidité du sol avecAWS IoTet Raspberry Pi

Ce didacticiel vous montre comment utiliser un Raspberry Pi, un capteur d'humidité, et AWS IoT pour surveiller le niveau d'humidité du sol pour une plante d'intérieur ou un jardin. Le Raspberry Pi exécute un code qui lit le niveau d'humidité et la température à partir du capteur, puis envoie les données à AWS IoT. Vous créez une règle dansAWS IoTqui envoie un e-mail à une adresse abonnée à une rubrique Amazon SNS lorsque le niveau d'humidité tombe en dessous d'un seuil.

Table des matières

Prerequisites

Pour suivre ce didacticiel, vous devez disposer des éléments suivants :

- Un Compte AWS .
- Utilisateur IAM disposant d'autorisations de niveau administrateur.
- Un ordinateur de développement exécutant Windows, macOS, Linux ou Unix pour accéder à la console AWS IoT.
- ARaspberry Pi 3B ou 4Bexécution de la dernièreOS Raspbian. Pour obtenir des instructions d'installation, consultez Installation des images du système d'exploitation sur le site web de Rasberry Pi.
- Un écran, un clavier, une souris et un réseau Wi-Fi ou une connexion Ethernet pour votre Raspberry Pi.
- Un capteur d'humidité compatible avec Raspberry Pi. Le capteur utilisé dans ce didacticiel est un capteur d'humidité capacitif SteMMA I2C Adafruit avec un en-tête de câble à 4 broches vers connecteur femelle JST.

Configuration d'AWS IoT

Pour suivre ce didacticiel, vous devez créer les ressources suivantes. Pour connecter un appareil à AWS IoT, vous créez un objet IoT, un certificat d'appareil et une stratégie AWS IoT.

- Un objet AWS IoT.

  Un objet représente un appareil physique (dans ce cas, votre Rasberry Pi) et contient des métadonnées statiques sur l'appareil.

- Un certificat d'appareil.

  Tous les appareils doivent avoir un certificat d'appareil pour se connecter à AWS IoT et s'authentifier auprès de celui-ci.
- Une stratégie AWS IoT.

  Chaque certificat d'appareil est associé à une ou plusieurs stratégies AWS IoT. Ces stratégies déterminent les ressources AWS IoT auxquelles l'appareil peut accéder.
- Un certificat d'autorité de certification racine AWS IoT.

  Les appareils et autres clients utilisent un certificat d'autorité de certification AWS IoT racine pour authentifier le serveur AWS IoTavec lequel ils communiquent. Pour plus d'informations, consultez Authentification du serveur (p. 233).
- Une règle AWS IoT.

  Une règle contient une requête et une ou plusieurs actions de règle. La requête extrait les données des messages de l'appareil pour déterminer si les données du message doivent être traitées. L'action de règle spécifie ce qu'il faut faire si les données correspondent à la requête.
- Une rubrique et un abonnement à la rubrique Amazon SNS.

  La règle écoute les données d'humidité de votre Raspberry Pi. Si la valeur est inférieure à un seuil, elle envoie un message à la rubrique Amazon SNS. Amazon SNS envoie ce message à toutes les adresses e-mail abonnées à la rubrique.

Création de la stratégie AWS IoT

Créez une stratégie AWS IoT qui permet à votre Raspberry Pi de se connecter et d'envoyer des messages à AWS IoT.

1. Dans la console AWS IoT, si un bouton Commencer s'affiche, appuyez dessus. Dans le panneau de navigation du service, développez Sécurité, puis Stratégies.
2. Si une boîte de dialogue Vous ne possédez pas encore de stratégie s'affiche, choisissez Créer une stratégie. Sinon, cliquez sur Create.
3. Entrez un nom pour la stratégie AWS IoT(par exemple, **`MoistureSensorPolicy`**).
4. Dans la section Ajouter des instructions, remplacez la stratégie existante par le code JSON suivant. Remplacez*`region`*and*`compte`*avec vos Région AWS and Compte AWS Nombre.

```
{
   "Version": "2012-10-17",
   "Statement": [{
         "Effect": "Allow",
         "Action": "iot:Connect",
         "Resource": "arn:aws:iot:region:account:client/RaspberryPi"
      },
      {
         "Effect": "Allow",
         "Action": "iot:Publish",
         "Resource": [
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get"
         ]
      },
      {
         "Effect": "Allow",
```

```
         "Action": "iot:Receive",
         "Resource": [
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/get/
accepted",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/update/
rejected",
            "arn:aws:iot:region:account:topic/$aws/things/RaspberryPi/shadow/delete/
rejected"
         ]
      },
      {
         "Effect": "Allow",
         "Action": "iot:Subscribe",
         "Resource": [
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
update/accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
delete/accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/get/
accepted",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
update/rejected",
            "arn:aws:iot:region:account:topicfilter/$aws/things/RaspberryPi/shadow/
delete/rejected"
         ]
      },
      {
         "Effect": "Allow",
         "Action": [
            "iot:GetThingShadow",
            "iot:UpdateThingShadow",
            "iot:DeleteThingShadow"
         ],
         "Resource": "arn:aws:iot:region:account:thing/RaspberryPi"

      }
   ]
}
```

5. Sélectionnez Créer.

Création de l'objet AWS IoT, du certificat et de la clé privée

Créez un objet dans le registre AWS IoT pour représenter votre Raspberry Pi.

1. Dans la console AWS IoT, dans le panneau de navigation, choisissez Gérer, puis Objets.
2. Si une boîte de dialogue Vous n'avez pas encore d'objets s'affiche, choisissez Enregistrer un objet. Sinon, cliquez sur Create.
3. Sur la page Création d'objets AWS IoT, choisissez Créer un objet unique.
4. Sur la page Add your device to the device registry (Ajouter votre appareil au registre des appareils), entrez un nom pour votre objet IoT (par exemple, **`RaspberryPi`**), puis choisissez Next (Suivant). Vous ne pouvez pas modifier le nom d'un objet après l'avoir créé. Pour changer le nom d'un objet, vous devez créer un objet, lui donner un nouveau nom, puis supprimer l'ancien objet.
5. Sur la page Add a certificate for your thing (Ajouter un certificat pour votre objet), choisissez Create certificate (Créer un certificat).
6. Choisissez les liens Télécharger pour télécharger le certificat, la clé privée et le certificat CA racine.

> **Important**
>
> C'est la seule fois que vous pouvez télécharger votre certificat et votre clé privée.

7. Pour activer le certificat, choisissezActivate. Le certificat doit être actif pour qu'un périphérique puisse se connecter àAWS IoT.
8. Choisissez Attacher une stratégie.
9. Pour Ajouter une stratégie pour votre objet, choisissez MoistureSensorPolicy, puis Enregistrer l'objet.

Créez une rubrique Amazon SNS et un abonnement

Créez une rubrique Amazon SNS et un abonnement.

1. À partir duAWSSNS de la consoleDans le volet de navigation, choisissezRubriques, puis choisissezCréation d'une rubrique.
2. Attribuez un nom à la rubrique (par exemple, **`MoistureSensorTopic`**).
3. Entrez un nom d'affichage pour la rubrique (par exemple, **`Moisture Sensor Topic`**). Il s'agit du nom affiché pour votre rubrique dans la console Amazon SNS.
4. Choisissez Create topic.
5. Sur la page des détails de la rubrique Amazon SNS, sélectionnezCréer un abonnement.
6. Pour Protocole, choisissez E-mail.
7. Saisissez votre adresse e-mail dans Point de terminaison.
8. Choisissez Créer un abonnement.
9. Ouvrez votre client de messagerie et recherchez un message avec l'objet **`MoistureSensorTopic`**. Ouvrez cet e-mail et cliquez sur le lien Confirmer l'abonnement.

> **Important**
>
> Vous ne recevrez aucune alerte par e-mail de cette rubrique Amazon SNS tant que vous n'aurez pas confirmé l'abonnement.

Vous devriez recevoir un message électronique contenant le texte que vous avez saisi.

Création d'une règle AWS IoT pour envoyer un e-mail

Une règle AWS IoT définit une requête et une ou plusieurs actions à effectuer lorsqu'un message est reçu à partir d'un appareil. Le moteur de règles AWS IoT écoute les messages envoyés par les appareils et utilise les données des messages pour déterminer si une action doit être effectuée. Pour plus d'informations, consultez Règles pour AWS IoT (p. 394).

Dans ce didacticiel, votre Raspberry Pi publie des messages sur `aws/things/RaspberryPi/shadow/update`. Il s'agit d'une rubrique MQTT interne utilisée par les appareils et le service Thing Shadow. Le Raspberry Pi publie des messages sous la forme suivante :

```
{
    "reported": {
        "moisture" : moisture-reading,
        "temp" : temperature-reading
    }
}
```

Vous créez une requête qui extrait les données d'humidité et de température du message entrant. Vous créez également une action Amazon SNS qui prend les données et les envoie aux abonnés de la rubrique Amazon SNS si le relevé d'humidité est inférieur à un seuil.

Création d'une règle Amazon SNS

1. Dans la console AWS IoT, dans le panneau de navigation, choisissez Agir. Si une boîte de dialogue Vous ne possédez pas encore de règle s'affiche, choisissez Créer une règle. Sinon, cliquez sur Create.
2. Sur la page Créer une règle, saisissez le nom de cette règle, (par exemple, **`MoistureSensorRule`**).
3. Pour Description, décrivez cette règle de manière significative (par exemple, **`Sends an alert when soil moisture level readings are too low`**).
4. Sous Instruction de requête de règle, choisissez SQL version 2016-03-23, et entrez l'instruction de requête SQL AWS IoT suivante :

```
SELECT * FROM '$aws/things/RaspberryPi/shadow/update/accepted' WHERE
 state.reported.moisture < 400
```

   Cette instruction déclenche l'action de la règle lorsque la valeur de `moisture` est inférieure à `400`.

   > **Note**
   >
   > Vous devrez peut-être utiliser une valeur différente. Une fois que le code s'exécute sur votre Raspberry Pi, vous pouvez voir les valeurs que vous obtenez de votre capteur en touchant le capteur, en le plaçant dans l'eau ou en le plaçant dans un pot.

5. Sous Définissez une ou plusieurs actions, choisissez Ajouter une action.
6. Sur la page Sélectionner une action, choisissez Envoyer un message en tant que notification push SNS.
7. Faites défiler jusqu'en bas de la page et choisissez Configurer une action.
8. Sur la page Configurer une action, pour Cible SNS, choisissez Sélectionner, puis LowMoistureTopic.
9. Pour Format du message, choisissez RAW.
10. Sous Choisissez ou créez un rôle afin d'autoriser AWS IoT à accéder à la ressource pour effectuer cette action, sélectionnez Créer un rôle. Entrez un nom pour le rôle (par exemple, **`LowMoistureTopicRole`**), puis choisissez Créer un rôle.
11. Choisissez Add action.
12. Choisissez Create rule.

Configuration de votre Raspberry Pi et du capteur d'humidité

Insérez votre carte micro SD dans le Raspberry Pi, connectez votre écran, votre clavier, votre souris et, si vous n'utilisez pas le Wi-Fi, un câble Ethernet. Ne connectez pas encore le câble d'alimentation.

Connectez le câble jumper JST au capteur d'humidité. L'autre côté du jumper a quatre câbles :

- Green (Vert) : I2C SCL
- Blanc : I2C SDA
- Rouge : alimentation (3,5 V)
- Black : terre

Maintenez la carte Raspberry Pi enfoncée avec la prise Ethernet sur la droite. Dans cette orientation, il y a deux lignes de broches GPIO en haut. Connectez les câbles du capteur d'humidité à la ligne inférieure de broches dans l'ordre suivant. À partir du connecteur le plus à gauche, connectez rouge (alimentation), blanc (SDA) et vert (SCL). Ignorez une broche, puis connectez le fil noir (terre). Pour plus d'informations, consultez Câblage informatique Python.

Attachez le câble d'alimentation au Raspberry Pi et branchez l'autre extrémité à une prise murale pour l'allumer.

Configuration de votre Raspberry Pi

1. Sur Welcome to Raspberry Pi (Bienvenue dans Raspberry Pi), choisissez Next (Suivant).
2. Choisissez votre pays, votre langue, votre fuseau horaire et votre disposition du clavier. Choisissez Suivant.
3. Saisissez un mot de passe pour votre Raspberry Pi, puis choisissez Next (Suivant).
4. Choisissez votre réseau Wi-Fi, puis choisissez Next (Suivant). Si vous n'utilisez pas de réseau Wi-Fi, choisissez Skip (Ignorer).
5. Choisissez Next (Suivant) pour rechercher les mises à jour logicielles. Lorsque les mises à jour sont terminées, choisissez Restart (Redémarrer) pour redémarrer votre Raspberry Pi.

Une fois que votre Raspberry Pi a démarré, activez l'interface I2C.

1. Dans le coin supérieur gauche du bureau Raspbian, cliquez sur l'icône Raspberry, choisissez Preferences (Préférences), puis Raspberry Pi Configuration (Configuration du Raspberry Pi).
2. Sous l'onglet Interfaces pour I2C, choisissez Enable (Activer).
3. Choisissez OK.

Les bibliothèques du capteur d'humidité SteMMA Adafruit sont écrites pour CircuitPython. Pour les exécuter sur un Raspberry Pi, vous devez installer la dernière version de Python 3.

1. Exécutez les commandes suivantes à partir d'une invite de commande pour mettre à jour votre logiciel Raspberry Pi :

   ```
   sudo apt-get update

   sudo apt-get upgrade
   ```
2. Exécutez la commande suivante pour mettre à jour votre installation Python 3 :

   ```
   sudo pip3 install --upgrade setuptools
   ```
3. Exécutez la commande suivante pour installer les bibliothèques GPIO Raspberry Pi :

   ```
   pip3 install RPI.GPIO
   ```
4. Exécutez la commande suivante pour installer les bibliothèques Adafruit Blinka :

   ```
   pip3 install adafruit-blinka
   ```

   Pour plus d'informations, consultez Installation des bibliothèques CircuitPython sur Raspberry Pi.
5. Exécutez la commande suivante pour installer les bibliothèques Adafruit Seesaw :

   ```
   sudo pip3 install adafruit-circuitpython-seesaw
   ```
6. Exécutez la commande suivante pour installer le kit SDK AWS IoT Device pour Python :

   ```
   pip3 install AWSIoTPythonSDK
   ```

Votre Raspberry Pi dispose désormais de toutes les bibliothèques requises. Créez un fichier appelé **`moistureSensor.py`** et copiez le code Python suivant dans le fichier :

```
from adafruit_seesaw.seesaw import Seesaw
```

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient
from board import SCL, SDA

import logging
import time
import json
import argparse
import busio

# Shadow JSON schema:
#
# {
#   "state": {
#       "desired":{
#           "moisture":<INT VALUE>,
#           "temp":<INT VALUE>
#       }
#   }
# }

# Function called when a shadow is updated
def customShadowCallback_Update(payload, responseStatus, token):

    # Display status and data from update request
    if responseStatus == "timeout":
        print("Update request " + token + " time out!")

    if responseStatus == "accepted":
        payloadDict = json.loads(payload)
        print("~~~~~~~~~~~~~~~~~~~~~~~")
        print("Update request with token: " + token + " accepted!")
        print("moisture: " + str(payloadDict["state"]["reported"]["moisture"]))
        print("temperature: " + str(payloadDict["state"]["reported"]["temp"]))
        print("~~~~~~~~~~~~~~~~~~~~~~~\n\n")

    if responseStatus == "rejected":
        print("Update request " + token + " rejected!")

# Function called when a shadow is deleted
def customShadowCallback_Delete(payload, responseStatus, token):

     # Display status and data from delete request
    if responseStatus == "timeout":
        print("Delete request " + token + " time out!")

    if responseStatus == "accepted":
        print("~~~~~~~~~~~~~~~~~~~~~~~")
        print("Delete request with token: " + token + " accepted!")
        print("~~~~~~~~~~~~~~~~~~~~~~~\n\n")

    if responseStatus == "rejected":
        print("Delete request " + token + " rejected!")


# Read in command-line parameters
def parseArgs():

    parser = argparse.ArgumentParser()
    parser.add_argument("-e", "--endpoint", action="store", required=True, dest="host",
 help="Your device data endpoint")
    parser.add_argument("-r", "--rootCA", action="store", required=True, dest="rootCAPath",
 help="Root CA file path")
    parser.add_argument("-c", "--cert", action="store", dest="certificatePath",
 help="Certificate file path")
    parser.add_argument("-k", "--key", action="store", dest="privateKeyPath", help="Private
 key file path")
```

```
    parser.add_argument("-p", "--port", action="store", dest="port", type=int, help="Port
 number override")
    parser.add_argument("-n", "--thingName", action="store", dest="thingName",
 default="Bot", help="Targeted thing name")
    parser.add_argument("-id", "--clientId", action="store", dest="clientId",
 default="basicShadowUpdater", help="Targeted client id")

    args = parser.parse_args()
    return args


# Configure logging
# AWSIoTMQTTShadowClient writes data to the log
def configureLogging():

    logger = logging.getLogger("AWSIoTPythonSDK.core")
    logger.setLevel(logging.DEBUG)
    streamHandler = logging.StreamHandler()
    formatter = logging.Formatter('%(asctime)s - %(name)s - %(levelname)s - %(message)s')
    streamHandler.setFormatter(formatter)
    logger.addHandler(streamHandler)


# Parse command line arguments
args = parseArgs()

if not args.certificatePath or not args.privateKeyPath:
    parser.error("Missing credentials for authentication.")
    exit(2)

# If no --port argument is passed, default to 8883
if not args.port:
    args.port = 8883


# Init AWSIoTMQTTShadowClient
myAWSIoTMQTTShadowClient = None
myAWSIoTMQTTShadowClient = AWSIoTMQTTShadowClient(args.clientId)
myAWSIoTMQTTShadowClient.configureEndpoint(args.host, args.port)
myAWSIoTMQTTShadowClient.configureCredentials(args.rootCAPath, args.privateKeyPath,
 args.certificatePath)

# AWSIoTMQTTShadowClient connection configuration
myAWSIoTMQTTShadowClient.configureAutoReconnectBackoffTime(1, 32, 20)
myAWSIoTMQTTShadowClient.configureConnectDisconnectTimeout(10) # 10 sec
myAWSIoTMQTTShadowClient.configureMQTTOperationTimeout(5) # 5 sec

# Initialize Raspberry Pi's I2C interface
i2c_bus = busio.I2C(SCL, SDA)

# Intialize SeeSaw, Adafruit's Circuit Python library
ss = Seesaw(i2c_bus, addr=0x36)

# Connect to AWS IoT
myAWSIoTMQTTShadowClient.connect()

# Create a device shadow handler, use this to update and delete shadow document
deviceShadowHandler = myAWSIoTMQTTShadowClient.createShadowHandlerWithName(args.thingName,
 True)

# Delete current shadow JSON doc
deviceShadowHandler.shadowDelete(customShadowCallback_Delete, 5)

# Read data from moisture sensor and update shadow
while True:
```

```
    # read moisture level through capacitive touch pad
    moistureLevel = ss.moisture_read()

    # read temperature from the temperature sensor
    temp = ss.get_temp()

    # Display moisture and temp readings
    print("Moisture Level: {}".format(moistureLevel))
    print("Temperature: {}".format(temp))

    # Create message payload
    payload = {"state":{"reported":{"moisture":str(moistureLevel),"temp":str(temp)}}}

    # Update shadow
    deviceShadowHandler.shadowUpdate(json.dumps(payload), customShadowCallback_Update, 5)
    time.sleep(1)
```

Enregistrez le fichier dans un emplacement où vous le trouverez. Sur la ligne de commande, tapez `moistureSensor.py` avec les paramètres suivants :

endpoint

Votre point de terminaison AWS IoT personnalisé Pour plus d'informations, consultez API REST Device Shadow (p. 573).

rootCA

Chemin complet de votre certificat d'autorité de certification AWS IoT racine.

cert

Chemin d'accès complet au certificat de votre AWS IoT appareil.

key

Chemin d'accès complet à la clé privée de votre certificat d'appareil AWS IoT.

thingName

Votre nom d'objet (dans ce cas, `RaspberryPi`).

clientId

ID du client MQTT. Utilisez `RaspberryPi`.

La ligne de commande doit se présenter comme suit :

```
python3 moistureSensor.py --endpoint your-endpoint --rootCA ~/certs/
AmazonRootCA1.pem --cert ~/certs/raspberrypi-certificate.pem.crt --key ~/certs/
raspberrypi-private.pem.key --thingName RaspberryPi --clientId RaspberryPi
```

Essayez de toucher le capteur, de le placer dans un pot ou de le placer dans un verre d'eau pour voir comment le capteur réagit à différents niveaux d'humidité. Si nécessaire, vous pouvez modifier la valeur de seuil dans `MoistureSensorRule`. Lorsque la lecture du capteur d'humidité est inférieure à la valeur spécifiée dans l'instruction de requête SQL de votre règle,AWS IoTPublie un message dans la rubrique Amazon SNS. Vous devez recevoir un e-mail contenant les données relatives à l'humidité et à la température.

Une fois que vous avez vérifié la réception des e-mails provenant d'Amazon SNS, appuyez surCTRL +CPour arrêter le programme Python. Il est peu probable que le programme Python envoie suffisamment de messages pour générer des frais, mais il est recommandé d'arrêter le programme une fois que vous avez terminé.

Gestion des appareils avec AWS IoT

AWS IoT fournit un registre vous permettant de gérer les objets. Un objet est une représentation d'un appareil spécifique ou d'une entité logique. Il peut s'agir d'un appareil physique ou d'un capteur (par exemple, une ampoule ou un interrupteur sur un mur). Il peut également s'agir dune entité logique, comme une instance d'une application ou une entité physique qui ne se connecte pas à AWS IoT, mais qui est associée à d'autres appareils qui se connectent (par exemple, une voiture dotée de capteurs de moteur ou d'un ordinateur de bord).

Les informations concernant un objet sont stockées dans le registre en tant que données JSON. Voici un exemple d'objet :

```
{
     "version": 3,
    "thingName": "MyLightBulb",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

Les objets sont identifiés par un nom. Ils peuvent également avoir des attributs, qui sont des paires nom-valeur, que vous pouvez utiliser pour stocker des informations concernant l'objet, comme son numéro de série ou le fabricant.

Un cas d'utilisation de appareil classique consisterait à utiliser le nom d'un objet en tant qu'ID de client MQTT par défaut. Même si nous n'appliquons pas de mappage entre le nom de registre d'un objet et son utilisation des ID client MQTT, des certificats et de l'état du shadow, nous vous recommandons de choisir un nom d'objet et de l'utiliser comme ID client MQTT pour le registre et le service Device Shadow. Cela permet d'organiser votre parc IoT plus facilement, sans perdre la souplesse du modèle de certificat d'appareil ou du shadow sous-jacent.

Vous n'avez pas besoin de créer d'objet dans le registre pour connecter un appareil à AWS IoT. L'ajout d'objets au registre permet de les gérer et de rechercher des appareils plus facilement.

Comment gérer des objets avec le registre

Vous utilisez la stratégieAWS IoTconsoleAWS IoT, ou l'APIAWS CLIpour interagir avec le registre. Les sections suivantes montrent comment utiliser l'interface de ligne de commande pour qu'elle fonctionne avec le registre.

Lorsque vous nommez vos objets chose :

- Vous ne devez pas utiliser d'informations personnelles identifiables dans le nom de votre objet. Le nom de chose peut apparaître dans les communications et les rapports non chiffrés.
- Vous ne devez pas utiliser un caractère deux-points (:) dans un nom de chose. Le caractère deux-points est utilisé comme délimiteur par d'autresAWS IoTet cela peut les amener à analyser les chaînes avec des noms de choses incorrectement.

Créer un objet

La commande suivante montre comment utiliser la commande AWS IoT CreateThing à partir de l'interface de ligne de commande pour créer un objet : Vous ne pouvez pas changer le nom d'un objet une fois qu'il est créé. Pour changer le nom d'un objet, vous devez créer un objet, lui donner un nouveau nom, puis supprimer l'ancien objet.

```
$ aws iot create-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\":
 {\"wattage\":\"75\", \"model\":\"123\"}}"
```

La commande CreateThing affiche le nom et l'Amazon Resource Name (ARN) de votre nouvel objet :

```
{
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
    "thingName": "MyLightBulb",
    "thingId": "12345678abcdefgh12345678ijklmnop12345678"
}
```

> **Note**
>
> Nous vous déconseillons d'utiliser des informations personnelles identifiables dans le nom de vos objets.

Liste des objets

Vous pouvez utiliser la commande ListThings pour répertorier tous les objets présents dans votre compte :

```
$ aws iot list-things
```

```
{
    "things": [
       {
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyLightBulb"
        },
        {
            "attributes": {
                "numOfStates":"3"
             },
            "version": 11,
            "thingName": "MyWallSwitch"
        }
    ]
}
```

Vous pouvez utiliser la stratégieListThingspour rechercher tous les objets d'un type d'objet spécifique :

```
$  aws iot list-things --thing-type-name "LightBulb"
```

```
{
    "things": [
        {
            "thingTypeName": "LightBulb",
            "attributes": {
```

```
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyRGBLight"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MySecondLightBulb"
        }
    ]
}
```

Vous pouvez utiliser la stratégieListThingspour rechercher tous les objets qui ont un attribut avec une valeur spécifique. Cette commande recherche uniquement les attributs pouvant faire l'objet d'une recherche.

```
$  aws iot list-things --attribute-name "wattage" --attribute-value "75"
```

```
{
    "things": [
        {
            "thingTypeName": "StopLight",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3,
            "thingName": "MyLightBulb"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyRGBLight"
        },
        {
            "thingTypeName": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MySecondLightBulb"
        }
    ]
}
```

SiIndexation (p. 758)est activé, vous pouvez utiliser la stratégiesearch-indexpour effectuer une recherche sur les attributs de chose, les valeurs d'ombre de périphérique et les valeurs de connectivité accessibles et non recherchables. Pour de plus amples informations sur les informations que vous pouvez interroger en utilisantsearch-index, voirExemples de requêtes sur des objets (p. 776)et la référence de l'interface de ligne de commande sur lesearch-index command.

Décrire

Vous pouvez utiliser la stratégieDescribeThingpour afficher des informations plus détaillées sur un objet :

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "version": 3,
    "thingName": "MyLightBulb",
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",
    "thingId": "12345678abcdefgh12345678ijklmnop12345678",
    "defaultClientId": "MyLightBulb",
    "thingTypeName": "StopLight",
    "attributes": {
        "model": "123",
        "wattage": "75"
    }
}
```

Mettre à jour un objet

Vous pouvez utiliser la commande UpdateThing pour mettre à jour un objet : Notez que cette commande met à jour uniquement les attributs de l'objet. Vous ne pouvez pas changer le nom d'un objet. Pour changer le nom d'un objet, vous devez créer un objet, lui donner un nouveau nom, puis supprimer l'ancien objet.

```
$ aws iot update-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\":
 {\"wattage\":\"150\", \"model\":\"456\"}}"
```

La commande UpdateThing ne génère pas de sortie. Vous pouvez voir le résultat à l'aide de la commande DescribeThing :

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "attributes": {
        "model": "456",
        "wattage": "150"
    },
    "version": 2,
    "thingName": "MyLightBulb"
}
```

Supprimer un objet

Vous pouvez utiliser la commande DeleteThing pour supprimer un objet :

```
$ aws iot delete-thing --thing-name "MyThing"
```

Cette commande renvoie un message de succès de l'opération sans erreur si la suppression a été réussie ou que vous spécifiez un objet qui n'existe pas.

Attacher un mandataire à un objet

Un appareil physique doit posséder un certificat X.509 pour pouvoir communiquer avec AWS IoT. Vous pouvez associer le certificat de votre appareil à l'objet dans le registre qui représente votre appareil. Pour attacher un certificat à votre objet, utilisez la commande AttachThingPrincipal :

```
$ aws iot attach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-
east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

La commande AttachThingPrincipal ne génère pas de sortie.

Détacher un mandataire d'un objet

Vous pouvez utiliser la commande DetachThingPrincipal pour détacher un certificat d'un objet :

```
$ aws iot detach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-
east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

La commande DetachThingPrincipal ne génère pas de sortie.

Types d'objets

Les types d'objets permettent de stocker des informations de configuration et de description communes à tous les objets qui associés au même type d'objet. Cela simplifie la gestion des objets dans le registre. Par exemple, vous pouvez définir un type d'objet ampoule, LightBulb. Tous les objets associés au type d'objet LightBulb partagent un ensemble d'attributs : numéro de série, fabricant et puissance. Lorsque vous créez un objet de type LightBulb (ou que vous modifiez le type d'un objet existant en LightBulb), vous pouvez spécifier des valeurs pour chacun des attributs définis dans le type d'objet LightBulb.

Bien que les types d'objet soient facultatifs, leur utilisation facilite la découverte des objets.

- Les objets avec un type d'objet peuvent posséder jusqu'à 50 attributs.
- Les objets sans type d'objet peuvent posséder jusqu'à trois attributs.
- Un objet ne peut être associé qu'à un seul type d'objet.
- Il n'y a aucune limite au nombre de types d'objets que vous pouvez créer dans votre compte.

Les types d'objets sont immuables. Vous ne pouvez pas modifier un nom de type d'objet après sa création. Vous pouvez rendre obsolète un type d'objet à tout moment pour empêcher de nouveaux objets d'y être associés. Vous pouvez aussi supprimer les types d'objets qui n'ont aucun objet associé.

Créer un type d'objet

Vous pouvez utiliser la commande CreateThingType pour créer un type d'objet :

```
$ aws iot create-thing-type

                --thing-type-name "LightBulb" --thing-type-properties
 "thingTypeDescription=light bulb type, searchableAttributes=wattage,model"
```

La commande CreateThingType renvoie une réponse qui contient le type d'objet et son ARN :

```
{
    "thingTypeName": "LightBulb",
    "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb"
}
```

Liste des types d'objets

Vous pouvez utiliser la commande ListThingTypes pour répertorier les types d'objets :

```
$ aws iot list-thing-types
```

La .ListThingTypesLa commande renvoie une liste des types d'objets définis dans Compte AWS :

```
{
    "thingTypes": [
        {
            "thingTypeName": "LightBulb",
            "thingTypeProperties": {
                "searchableAttributes": [
                    "wattage",
                    "model"
                ],
                "thingTypeDescription": "light bulb type"
            },
            "thingTypeMetadata": {
                "deprecated": false,
                "creationDate": 1468423800950
            }
        }
    ]
}
```

Décrire un type d'objet

Vous pouvez utiliser la commande DescribeThingType pour obtenir des informations sur un type d'objet :

```
$ aws iot describe-thing-type --thing-type-name "LightBulb"
```

La commande DescribeThingType renvoie des informations sur le type spécifié :

```
{
    "thingTypeProperties": {
        "searchableAttributes": [
            "model",
            "wattage"
        ],
        "thingTypeDescription": "light bulb type"
    },
    "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb",
    "thingTypeName": "LightBulb",
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1544466338.399
    }
}
```

Associer un type d'objet à un objet

Vous pouvez utiliser la commande CreateThing pour spécifier un type d'objet lorsque vous créez un objet :

```
$ aws iot create-thing --thing-name "MyLightBulb" --thing-type-name "LightBulb" --
attribute-payload "{\"attributes\": {\"wattage\":\"75\", \"model\":\"123\"}}"
```

Vous pouvez utiliser la commande UpdateThing à tout moment pour modifier le type d'objet associé à un objet :

```
$ aws iot update-thing --thing-name "MyLightBulb"
                --thing-type-name "LightBulb" --attribute-payload  "{\"attributes\":
 {\"wattage\":\"75\", \"model\":\"123\"}}"
```

Vous pouvez également utiliser la commande UpdateThing pour dissocier un objet d'un type d'objet.

Rendre obsolète un type d'objet

Les types d'objets sont immuables. Il est impossible de les modifier une fois qu'ils sont définis. Vous pouvez, toutefois, rendre obsolète un type d'objet pour empêcher les utilisateurs de lui associer de nouveaux objets. Tous les objets existants associés au type d'objet restent inchangés.

Pour rendre obsolète un type d'objet, utilisez la commande DeprecateThingType :

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType"
```

Vous pouvez voir le résultat à l'aide de la commande DescribeThingType :

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{
    "thingTypeName": "StopLight",
    "thingTypeProperties": {
        "searchableAttributes": [
            "wattage",
            "numOfLights",
            "model"
        ],
        "thingTypeDescription": "traffic light type",
    },
    "thingTypeMetadata": {
        "deprecated": true,
        "creationDate": 1468425854308,
        "deprecationDate": 1468446026349
    }
}
```

Rendre obsolète un type d'objet est une opération réversible. Vous pouvez annuler une obsolescence en utilisant l'indicateur `--undo-deprecate` avec la commande CLI DeprecateThingType :

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType" --undo-deprecate
```

Vous pouvez voir le résultat à l'aide de la commande CLI DescribeThingType :

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{
    "thingTypeName": "StopLight",
    "thingTypeArn": "arn:aws:iot:us-east-1:123456789012:thingtype/StopLight",
    "thingTypeId": "12345678abcdefgh12345678ijklmnop12345678"
    "thingTypeProperties": {
        "searchableAttributes": [
```

```
            "wattage",
            "numOfLights",
            "model"
        ],
        "thingTypeDescription": "traffic light type"
    },
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1468425854308,
    }
}
```

Supprimer un type d'objet

Vous pouvez supprimer des types d'objet uniquement une fois qu'ils sont obsolètes. Pour supprimer un type d'objet, utilisez la commande DeleteThingType :

```
$ aws iot delete-thing-type --thing-type-name "StopLight"
```

> **Note**
>
> Après avoir rendu obsolète un type d'objet, vous devez attendre 5 minutes avant de le supprimer.

Groupes d’objets statiques

Les groupes d'objets permettent de gérer plusieurs objets simultanément en les classant dans des groupes. Les groupes d’objets statiques contiennent des objets gérés grâce à la console, l'interface de ligne de commande ou l'API. Les groupes d’objets dynamiques (p. 220), en revanche, contiennent des objets qui correspondent à une requête spécifiée. Les groupes d'objets statiques peuvent eux aussi contenir d'autres groupes d'objets statiques ; vous pouvez ainsi créer une hiérarchie de groupes. Vous pouvez attacher à un groupe parent une stratégie dont héritent tous ses groupes enfants et tous les objets du groupe, ainsi que leurs groupes enfants. Cela facilite le contrôle des autorisations pour les grands nombres d'objets.

Voici ce que vous pouvez faire avec les groupes d'objets statiques :

- Créer, décrire ou supprimer un groupe.
- Ajouter un objet à un ou plusieurs groupes.
- Supprimer un objet d'un groupe.
- Répertorier les groupes que vous avez créés.
- Répertorier tous les groupes enfants d'un groupe (ses descendants directs et indirects).
- Répertorier les objets d'un groupe, y compris tous les objets de ses groupes enfants.
- Répertorier tous les groupes ascendants d'un groupe (ses parents directs et indirects).
- Ajouter, supprimer ou mettre à jour les attributs d'un groupe. Les attributs sont des paires nom-valeur que vous pouvez utiliser afin de stocker des informations relatives à un groupe.
- Attacher une stratégie à un groupe ou la détacher de celui-ci.
- Répertorier les stratégies attachées à un groupe.
- Répertorier les stratégies dont un objet hérite (en fonction des stratégies attachées à son groupe ou à l'un de ses groupes parents).
- Configurer les options de journalisation des objets d'un groupe. Voir Configurer la journalisation AWS IoT (p. 353).

- Créer des tâches qui sont envoyées vers et exécutées sur chaque objet d'un groupe et de ses groupes enfants. Voir Jobs (p. 595).

Voici quelques restrictions relatives aux groupes d'objets statiques :

- Un groupe peut avoir un parent direct au maximum.
- Si un groupe doit être utilisé comme enfant d'un autre groupe, vous devez le spécifier au moment de sa création.
- Vous ne pouvez pas modifier le parent d'un groupe ultérieurement. Veillez donc à planifier votre hiérarchie de groupe et à créer un groupe parent avant de créer les groupes enfants qu'il contient.
- Le nombre de groupes auxquels un objet peut appartenir est limité.
- Vous ne pouvez pas ajouter un objet à plusieurs groupes de la même hiérarchie. (En d'autres termes, vous ne pouvez pas ajouter un objet à deux groupes qui partagent un même parent).
- Vous ne pouvez pas renommer un groupe.
- Les noms des groupes d'objets ne peuvent contenir aucun caractère international, comme û, é et ñ.
- Vous ne devez pas utiliser d'informations personnelles identifiables dans le nom de votre groupe d'objets. Le nom du groupe de choses peut apparaître dans les communications et les rapports non chiffrés.
- Vous ne devez pas utiliser un caractère deux-points (:) dans un nom de groupe de choses. Le caractère deux-points est utilisé comme délimiteur par d'autresAWS IoTet cela peut les amener à analyser les chaînes avec les noms de groupes de choses de manière incorrecte.

Le fait d'attacher des stratégies aux groupes ou de les détacher de ceux-ci vous permet d'améliorer la sécurité des opérations AWS IoT de nombreuses façons importantes. La méthode par appareil qui consiste à attacher une stratégie à un certificat, qui est lui-même attaché ensuite à un objet, prend du temps et complique la mise à jour ou la modification rapide des stratégies pour tout un parc d'appareils. Le fait d'attacher une stratégie au groupe de l'objet permet d'éviter certaines étapes lors de la rotation des certificats pour un objet. De plus, les stratégies sont appliquées dynamiquement aux objets lorsqu'elles changent d'appartenance à un groupe, ce qui signifie que vous n'avez pas besoin de recréer un ensemble complexe d'autorisations chaque fois qu'un appareil change d'appartenance dans un groupe.

Créer un groupe d’objets statiques

Utilisez la commande CreateThingGroup pour créer un groupe d'objets statiques.

```
$ aws iot create-thing-group --thing-group-name LightBulbs
```

La commande CreateThingGroup renvoie une réponse qui contient le nom, l'ID et l'ARN du groupe d’objets statiques :

```
{
    "thingGroupName": "LightBulbs",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
}
```

> **Note**
>
> Nous vous déconseillons d'utiliser des informations personnelles identifiables dans le nom de vos groupes d'objets.

Voici un exemple qui spécifie un parent du groupe d'objets statiques lors de sa création :

```
$ aws iot create-thing-group --thing-group-name RedLights --parent-group-name LightBulbs
```

Comme auparavant, la commande CreateThingGroup renvoie une réponse qui contient le nom, l'ID et l'ARN du groupe d'objets statiques :

```
{
    "thingGroupName": "RedLights",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",
}
```

Important

Gardez à l'esprit les limites suivantes lorsque vous créez des hiérarchies de groupes d'objets :

- Un groupe d'objets ne peut avoir qu'un seul parent direct.
- Le nombre de groupes d'enfants directs qu'un groupe d'objets peut avoir est limité.
- Le nombre maximal de niveaux d'une hiérarchie de groupes d'objets est limité.
- Le nombre d'attributs qu'un groupe d'objets peut avoir est limité. Les attributs sont des paires nom-valeur que vous pouvez utiliser afin de stocker des informations relatives à un groupe. La longueur du nom de chaque attribut et de chaque valeur est également limitée.

Décrire un groupe d'objets

Pour obtenir des informations sur un groupe d'objets, vous pouvez utiliser la commande DescribeThingGroup :

```
$ aws iot describe-thing-group --thing-group-name RedLights
```

La commande DescribeThingGroup renvoie des informations sur le groupe spécifié :

```
{
    "thingGroupName": "RedLights",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",
    "thingGroupId": "12345678abcdefgh12345678ijklmnop12345678",
    "version": 1,
    "thingGroupMetadata": {
        "creationDate": 1478299948.882
        "parentGroupName": "Lights",
        "rootToParentThingGroups": [
            {
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ShinyObjects",
                "groupName": "ShinyObjects"
            },
            {
                "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs",
                "groupName": "LightBulbs"
            }
        ]
    },
    "thingGroupProperties": {
        "attributePayload": {
            "attributes": {
                "brightness": "3400_lumens"
            },
        },
        "thingGroupDescription": "string"
```

```
    },
}
```

Ajouter un objet à un groupe d’objets statiques

Vous pouvez utiliser la commande AddThingToThingGroup pour ajouter un objet à un groupe d’objets statiques :

```
$ aws iot add-thing-to-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

La commande AddThingToThingGroup ne génère pas de sortie.

> Important
>
> Vous pouvez ajouter un objet à un maximum de 10 groupes. Mais vous ne pouvez pas ajouter un objet à plusieurs groupes de la même hiérarchie. En d'autres termes, vous ne pouvez pas ajouter un objet à deux groupes qui partagent un même parent.
> Si un objet appartient à autant de groupes d'objets que possible et qu'un ou plusieurs de ces groupes est un groupe d'objets dynamiques, vous pouvez utiliser l'indicateur overrideDynamicGroups pour que les groupes statiques soient prioritaires par rapport aux groupes dynamiques.

Supprimer un objet d'un groupe d’objet statiques

Vous pouvez utiliser la commande RemoveThingFromThingGroup pour supprimer un objet d'un groupe :

```
$ aws iot remove-thing-from-thing-group --thing-name MyLightBulb --thing-group-name
 RedLights
```

La commande RemoveThingFromThingGroup ne génère pas de sortie.

Répertorier les objets d'un groupe d'objets

Vous pouvez utiliser la commande ListThingsInThingGroup pour répertorier les objets appartenant à un groupe :

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs
```

La commande ListThingsInThingGroup renvoie une liste des objets d'un groupe donné :

```
{
    "things":[
        "TestThingA"
    ]
}
```

Le paramètre `--recursive` vous permet de répertorier les objets appartenant à un groupe et ceux figurant dans ses groupes enfants :

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs --recursive
```

```
{
    "things":[
```

```
        "TestThingA",
        "MyLightBulb"
    ]
}
```

Note

Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au groupe d'objets peuvent ne pas être reflétées immédiatement.

Répertorier les groupes d'objets

Vous pouvez utiliser la commande ListThingGroups pour répertorier les groupes d'objets de votre compte :

```
$ aws iot list-thing-groups
```

La .ListThingGroupsLa commande renvoie une liste des groupes d'objets dans votre Compte AWS :

```
{
    "thingGroups": [
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        },
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "RedLEDLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLEDLights"
        },
        {
            "groupName": "RedIncandescentLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedIncandescentLights"
        }
        {
            "groupName": "ReplaceableObjects",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ReplaceableObjects"
        }
    ]
}
```

Utilisez les filtres facultatifs pour répertorier les groupes ayant un groupe donné comme parent (`--parent-group`) ou ceux dont le nom commence par un préfixe spécifique (`--name-prefix-filter`). Le paramètre `--recursive` vous permet de répertorier tous les groupes enfants, et pas seulement les groupes enfants directs d'un groupe d'objets :

```
$ aws iot list-thing-groups --parent-group LightBulbs
```

Dans ce cas,ListThingGroupsLa commande renvoie une liste des groupes enfants directs du groupe d'objets défini dans Compte AWS :

```
{
    "childGroups":[
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
```

```
        }
    ]
}
```

Utilisez le paramètre `--recursive` avec la commande ListThingGroups pour répertorier tous les groupes enfants d'un groupe d'objets, et pas seulement les enfants directs :

```
$ aws iot list-thing-groups --parent-group LightBulbs --recursive
```

La commande ListThingGroups renvoie une liste de tous les groupes enfants du groupe d'objets :

```
{
    "childGroups":[
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "RedLEDLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLEDLights"
        },
        {
            "groupName": "RedIncandescentLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedIncandescentLights"
        }
    ]
}
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au groupe d'objets peuvent ne pas être reflétées immédiatement.

Répertorier les groupes d'un objet

Vous pouvez utiliser la commande ListThingGroupsForThing afin de répertorier les groupes auxquels appartient un objet, dont tous les groupes parents :

```
$ aws iot list-thing-groups-for-thing --thing-name MyLightBulb
```

La commande ListThingGroupsForThing renvoie une liste des groupes d'objets auxquels appartient un objet, dont tous les groupes parents :

```
{
    "thingGroups":[
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        },
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "ReplaceableObjects",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ReplaceableObjects"
        }
    ]
```

```
}
```

Mettre à jour un groupe d'objets statiques

Vous pouvez utiliser la commande UpdateThingGroup afin de mettre à jour les attributs d'un groupe d'objets statiques :

```
$ aws iot update-thing-group --thing-group-name "LightBulbs" --thing-group-properties
 "thingGroupDescription=\"this is a test group\", attributePayload=\"{\"attributes
\"={\"Owner\"=\"150\",\"modelNames\"=\"456\"}}"
```

La commande UpdateThingGroup renvoie une réponse qui contient le numéro de version du groupe après la mise à jour :

```
{
    "version": 4
}
```

> **Note**
>
> Le nombre d'attributs qu'un objet peut avoir est limité.

Supprimer un groupe d'objets

Pour supprimer un groupe d'objets, utilisez la commande DeleteThingGroup :

```
$ aws iot delete-thing-group --thing-group-name "RedLights"
```

La commande DeleteThingGroup ne génère pas de sortie.

> **Important**
>
> Si vous essayez de supprimer un groupe d'objets qui comporte des groupes d'objets enfants, vous recevez une erreur :
>
> ```
> A client error (InvalidRequestException) occurred when calling the
>  DeleteThingGroup
> operation: Cannot delete thing group : RedLights when there are still child groups
>  attached to it.
> ```
>
> Vous devez supprimer tous les groupes enfants avant de supprimer le groupe.

Vous pouvez supprimer un groupe qui a des objets enfants, mais les autorisations accordées aux objets par appartenance au groupe ne s'appliqueront plus. Avant de supprimer un groupe auquel une stratégie est attachée, vérifiez que la suppression de ces autorisations n'empêchera pas les objets du groupe de fonctionner correctement. Notez également que les commandes qui affichent à quels groupes un objet appartient (par exemple, ListGroupsForThing) peuvent continuer à afficher le groupe pendant que les enregistrements sont mis à jour sur le cloud.

Attacher une stratégie à un groupe d'objets statiques

Vous pouvez utiliser la commande AttachPolicy pour attacher une stratégie à un groupe d'objets statiques et, par extension, à tous les objets de ce groupe et de ses groupes enfants :

```
$ aws iot attach-policy \
```

```
  --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \
  --policy-name "myLightBulbPolicy"
```

La commande AttachPolicy ne génère pas de sortie

> Important
>
> Vous pouvez attacher au maximum deux stratégies à un groupe.
>
> Note
>
> Nous vous déconseillons d'utiliser des informations personnelles identifiables dans le nom de vos stratégies.

La `--target`peut être un ARN de groupe d'objets (comme ci-dessus), un ARN de certificat ou une identité Amazon Cognito. Pour plus d'informations sur les stratégies, les certificats et l'authentification, consultez Authentication (p. 232).

Détacher une stratégie d'un groupe d’objets statiques

Vous pouvez utiliser la commande DetachPolicy pour détacher une stratégie d'un groupe d'objets et, par extension, de tous les objets de ce groupe et de ses groupes enfants :

```
$ aws iot detach-policy --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
 --policy-name "myLightBulbPolicy"
```

La commande DetachPolicy ne génère pas de sortie.

Répertorier les stratégies attachées à un groupe d'objets statiques

Vous pouvez utiliser la commande ListAttachedPolicies pour répertorier les stratégies attachées à un groupe d’objets statiques :

```
$ aws iot list-attached-policies --target "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedLights"
```

La `--target`peut être un ARN de groupe d'objets (comme ci-dessus), un ARN de certificat ou une identité Amazon Cognito.

Ajoutez le paramètre facultatif `--recursive` afin d'inclure toutes les stratégies attachées aux groupes parents du groupe.

La commande ListAttachedPolicies renvoie une liste de stratégies :

```
{
    "policies": [
        "MyLightBulbPolicy"
        ...
    ]
}
```

Répertorier les groupes d'une stratégie

Vous pouvez utiliser la commande ListTargetsForPolicy afin de répertorier les cibles, y compris tous les groupes éventuels, auxquelles une stratégie est attachée :

```
$ aws iot list-targets-for-policy --policy-name "MyLightBulbPolicy"
```

Ajoutez le paramètre facultatif `--page-size` *`number`* afin de spécifier le nombre maximal de résultats renvoyés par chaque demande, et le paramètre `--marker` *`string`* sur les appels suivants afin d'extraire l'ensemble de résultats suivant, le cas échéant.

La commande ListTargetsForPolicy renvoie une liste des cibles et des jetons à utiliser pour extraire davantage de résultats :

```
{
    "nextMarker": "string",
    "targets": [ "string" ... ]
}
```

Obtenir des stratégies efficaces pour un objet

Vous pouvez utiliser la commande GetEffectivePolicies afin de répertorier les stratégies en vigueur pour un objet, y compris les stratégies attachées aux groupes auxquels l'objet appartient (que le groupe soit un parent direct ou un ancêtre indirect) :

```
$ aws iot get-effective-policies \
  --thing-name "MyLightBulb" \
  --principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

Utilisez le paramètre `--principal` afin de spécifier l'ARN du certificat attaché à l'objet. Si vous utilisez l'authentification d'identité Amazon Cognito, utilisez le`--cognito-identity-pool-id`et, éventuellement, ajoutez le paramètre`--principal`pour spécifier une identité Amazon Cognito. Si vous spécifiez uniquement le `--cognito-identity-pool-id`, les stratégies associées à ce rôle du pool d'identités pour les utilisateurs non authentifiés sont renvoyées. Si vous utilisez les deux, les stratégies associées à ce rôle du pool d'identités pour les utilisateurs authentifiés sont renvoyées.

Le paramètre `--thing-name` est facultatif et peut être utilisé à la place du paramètre `--principal`. Lorsqu'il est utilisé, les stratégies attachées aux groupes auxquels l'objet appartient et les stratégies attachées aux groupes parents de ces groupes (jusqu'au groupe racine dans la hiérarchie) sont renvoyées.

La commande GetEffectivePolicies renvoie une liste de stratégies :

```
{
    "effectivePolicies": [
        {
            "policyArn": "string",
            "policyDocument": "string",
            "policyName": "string"
        }
        ...
    ]
}
```

Tester l'autorisation pour les actions MQTT

Vous pouvez utiliser la commande TestAuthorization afin de tester si une action MQTT est autorisée pour un objet :

```
aws iot test-authorization \
```

```
    --principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847" \
    --auth-infos "{\"actionType\": \"PUBLISH\", \"resources\": [ \"arn:aws:iot:us-
east-1:123456789012:topic/my/topic\"]}"
```

Utilisez le paramètre `--principal` afin de spécifier l'ARN du certificat attaché à l'objet. Si vous utilisez l'authentification d'identité Amazon Cognito, spécifiez une identité Cognito comme`--principal`ou utilisez`--cognito-identity-pool-id`, ou les deux. Si vous spécifiez uniquement le paramètre `--cognito-identity-pool-id`, les stratégies associées à ce rôle du pool d'identités pour les utilisateurs non authentifiés sont appliquées. Si vous utilisez les deux, les stratégies associées à ce rôle du pool d'identités pour les utilisateurs authentifiés sont appliquées.

Spécifiez une ou plusieurs actions MQTT que vous souhaitez tester en répertoriant des ensembles de ressources et de types d'actions après le paramètre `--auth-infos`. Le champ `actionType` doit contenir « PUBLISH », « SUBSCRIBE », « RECEIVE » ou « CONNECT ». Le champ `resources` doit contenir une liste d'ARN des ressources. Pour plus d'informations, consultez Stratégies AWS IoT Core (p. 270).

Vous pouvez tester les conséquences de l'ajout des stratégies en les spécifiant avec le paramètre `--policy-names-to-add`. Ou vous pouvez tester les conséquences de la suppression des stratégies en les spécifiant avec le paramètre `--policy-names-to-skip`.

Vous pouvez utiliser le paramètre facultatif `--client-id` pour affiner vos résultats.

La commande TestAuthorization renvoie des détails sur les actions qui ont été autorisées ou refusées pour chaque ensemble de requêtes `--auth-infos` que vous avez spécifié :

```
{
    "authResults": [
        {
            "allowed": {
                "policies": [
                    {
                        "policyArn": "string",
                        "policyName": "string"
                    }
                ]
            },
            "authDecision": "string",
            "authInfo": {
                "actionType": "string",
                "resources": [ "string" ]
            },
            "denied": {
                "explicitDeny": {
                    "policies": [
                        {
                            "policyArn": "string",
                            "policyName": "string"
                        }
                    ]
                },
                "implicitDeny": {
                    "policies": [
                        {
                            "policyArn": "string",
                            "policyName": "string"
                        }
                    ]
                }
            },
            "missingContextValues": [ "string" ]
        }
    ]
```

```
}
```

Groupes d'objets dynamiques

Les groupes d'objets dynamiques mettent à jour l'appartenance à un groupe par le biais de requêtes de recherche. Les groupes d'objets dynamiques vous permettent de modifier la façon dont vous interagissez avec les objets en fonction de leur connectivité, de leur registre ou de leurs données de shadows.

Étant donné que les groupes d'objets dynamiques sont liées à votre index de flotte, vous devez activer le service d'indexation de flotte pour les utiliser. Vous pouvez prévisualiser les objets d'un groupe d'objets dynamique avant de créer le groupe avec une requête de recherche d'indexation de flotte. Pour plus d'informations, consultez Service d'indexation de flotte (p. 758) et Syntaxe de requête (p. 775).

Vous pouvez spécifier un groupe d'objets dynamique en tant que cible d'une tâche. Seuls les objets qui répondent aux critères définissant le groupe d'objets dynamique exécutent la tâche.

Par exemple, supposons que vous souhaitiez mettre à jour le microprogramme sur vos appareils, mais que, pour minimiser le risque d'interruption de cette mise à jour, vous souhaitiez uniquement mettre à jour le microprogramme sur les appareils dont l'autonomie de batterie est supérieure à 80 %. Vous pouvez créer un groupe d'objets dynamique incluant uniquement les appareils avec une autonomie de batterie signalée au-dessus de 80 % et vous pouvez utiliser ce groupe d'objets dynamique comme cible de votre tâche de mise à jour du microprogramme. Seuls les appareils répondant à vos critères d'autonomie de batterie reçoivent la mise à jour du microprogramme. Au fur et à mesure que les appareils atteignent les critères d'autonomie de batterie de 80 %, ils sont ajoutés au groupe d'objets dynamique et reçoivent la mise à jour du microprogramme.

Pour de plus amples informations sur la spécification de groupes de choses en tant que cibles de travail, veuillez consulter CreateJob (p. 633).

Voici les raisons qui différencient les groupes d'objets dynamiques des groupes d'objets statiques :

- L'appartenance des objets n'est pas explicitement définie. Pour créer un groupe d'objets dynamique, vous devez définir une chaîne de requête qui définit l'appartenance au groupe.
- Les groupes d'objets dynamiques ne peuvent pas faire partie d'une hiérarchie.
- Les groupes d’objets dynamiques ne peuvent pas avoir de stratégies appliquées.
- Vous utilisez un ensemble de commandes différent pour créer, mettre à jour et supprimer des groupes d'objets dynamiques. Pour tous les autres opérations, les commandes que vous utilisez pour interagir avec les groupes d'objets statiques peuvent également être utilisées pour interagir avec les groupes d'objets dynamiques.
- Le nombre de groupes dynamiques qu'un seul compte peut avoir est limité.

- Vous ne devez pas utiliser d'informations personnelles identifiables dans le nom de votre groupe d'objets. Le nom du groupe de choses peut apparaître dans les communications et les rapports non chiffrés.
- Vous ne devez pas utiliser un caractère deux-points (:) dans un nom de groupe de choses. Le caractère deux-points est utilisé comme délimiteur par d'autresAWS IoTet cela peut les amener à analyser les chaînes avec les noms de groupes de choses de manière incorrecte.

Pour plus d'informations sur les groupes d'objets statiques, consultez Groupes d’objets statiques (p. 210).

Par exemple, supposons que nous créons un groupe dynamique contenant toutes les salles d'un entrepôt dont la température est supérieure à 60°F. Si la température d'une salle atteint ou dépasse 61°, la salle est ajoutée au groupe d'objets dynamique RoomTooWarm. Les ventilateurs de refroidissement se déclenchent

dans toutes les salles appartenant au groupe dynamique RoomTooWarm. Si la température d'une salle atteint ou descend sous 60°F, la salle est retirée du groupe d'objets dynamique et son ventilateur est désactivé.

Créer un groupe d'objets dynamique

Utilisez la commande CreateDynamicThingGroup pour créer un groupe d'objets dynamique. Pour créer un groupe d'objets dynamique dans le cadre du scénario de la salle où la température est trop élevée, utilisez la commande de l'interface de ligne de commande create-dynamic-thing-group :

```
$ aws iot create-dynamic-thing-group --thing-group-name "RoomTooWarm" --query-string
 "attributes.temperature>60"
```

> **Note**
>
> Nous vous déconseillons d'utiliser des informations personnelles identifiables dans le nom de vos groupes d'objets dynamiques.

La commande CreateDynamicThingGroup renvoie une réponse qui contient le nom de l'index, la chaîne de requête, la version de la requête, ainsi que le nom, l'ID et l'ARN du groupe d'objets :

```
{
    "indexName": "AWS_Things",
    "queryVersion": "2017-09-30",
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "queryString": "attributes.temperature>60\n",
    "thingGroupId": "abcdefgh12345678ijklmnop12345678qrstuvwx"
}
```

La création d'un groupe d'objets dynamique n'est pas instantanée. Le remplissage complet d'un groupe d'objets dynamique prend un certain temps. Lorsqu'un groupe d'objets dynamique est créé, l'état du groupe est défini sur `BUILDING`. Une fois le remplissage terminé, l'état passe à `ACTIVE`. Pour vérifier l'état de votre groupe d'objets dynamique, utilisez la commande DescribeThingGroup.

Décrire un groupe d'objets dynamique

Utilisez la commande DescribeThingGroup pour obtenir des informations sur un groupe d'objets dynamique :

```
$ aws iot describe-thing-group --thing-group-name "RoomTooWarm"
```

La commande DescribeThingGroup renvoie des informations sur le groupe spécifié :

```
{
    "status": "ACTIVE",
    "indexName": "AWS_Things",
    "thingGroupName": "RoomTooWarm",
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RoomTooWarm",
    "queryString": "attributes.temperature>60\n",
    "version": 1,
    "thingGroupMetadata": {
        "creationDate": 1548716921.289
    },
    "thingGroupProperties": {},
    "queryVersion": "2017-09-30",
    "thingGroupId": "84dd9b5b-2b98-4c65-84e4-be0e1ecf4fd8"
```

```
}
```

L'exécution de DescribeThingGroup sur un groupe d'objets dynamique renvoie les attributs qui sont spécifiques aux groupes d'objets dynamiques, tels que l'attribut queryString et le statut.

Le statut d'un groupe d'objets dynamique peut avoir les valeurs suivantes :

`ACTIVE`

Le groupe d'objets dynamique est prêt à l'emploi.

`BUILDING`

Le groupe d'objets dynamique est en cours de création et l'appartenance des objets est en cours de traitement.

`REBUILDING`

L'appartenance au groupe d'objets dynamique est en cours de mise à jour, en fonction de l'ajustement de la requête de recherche du groupe.

Note

Une fois que vous avez créé un groupe d'objets dynamique, vous pouvez l'utiliser, quel que soit son statut. Seuls les groupes d'objets dynamiques ayant le statut `ACTIVE` incluent tous les objets qui correspondent à la requête de recherche de ce groupe d'objets dynamique. Les groupes d'objets dynamiques ayant les statuts `BUILDING` et `REBUILDING` peuvent ne pas inclure tous les objets correspondant à la requête de recherche.

Mettre à jour un groupe d'objets dynamique

Utilisez la commande UpdateDynamicThingGroup pour mettre à jour les attributs d'un groupe d'objets dynamique, y compris la requête de recherche du groupe. La commande suivante met à jour la description du groupe d'objets et la chaîne de recherche en modifiant les critères d'appartenance sur une température supérieure à 65 :

```
$ aws iot update-dynamic-thing-group --thing-group-name "RoomTooWarm" --thing-group-
properties "thingGroupDescription=\"This thing group contains rooms warmer than 65F.\"" --
query-string "attributes.temperature>65"
```

La commande UpdateDynamicThingGroup renvoie une réponse qui contient le numéro de version du groupe après la mise à jour :

```
{
    "version": 2
}
```

Les mises à jour des groupes d'objets dynamiques ne sont pas instantanées. Le remplissage complet d'un groupe d'objets dynamique prend un certain temps. Lorsqu'un groupe d'objets dynamique est mis à jour, le statut du groupe devient `REBUILDING` pendant la mise à jour de l'appartenance au groupe. Une fois le remplissage terminé, l'état passe à `ACTIVE`. Pour vérifier l'état de votre groupe d'objets dynamique, utilisez la commande DescribeThingGroup.

Supprimer un groupe d'objets dynamique

Utilisez la commande DeleteDynamicThingGroup pour supprimer un groupe d'objets dynamique :

```
$ aws iot delete-dynamic-thing-group --thing-group-name "RoomTooWarm"
```

La commande DeleteDynamicThingGroup ne génère pas de sortie.

Les commandes qui affichent à quels groupes un objet appartient (par exemple, ListGroupsForThing) peuvent continuer à afficher le groupe pendant que les enregistrements sont mis à jour sur le cloud.

Limitations et conflits

Les groupes d'objets dynamiques partagent les limitations suivantes avec les groupes d'objets statiques :

- Le nombre d'attributs qu'un groupe d'objets peut avoir est limité.
- Le nombre de groupes auxquels un objet peut appartenir est limité.
- Les groupes d'objets ne peuvent pas être renommés.
- Les noms des groupes d'objets ne peuvent contenir aucun caractère international, comme û, é et ñ.

Lorsque vous utilisez des groupes d'objets dynamiques, gardez à l'esprit ce qui suit :

Le service d'indexation de flotte doit être activé

Le service d'indexation de flotte doit être activé et le remplissage de l'indexation de flotte doit être terminé pour que vous puissiez créer et utiliser les groupes d'objets dynamiques. Vous devez attendre un peu après avoir activé le service d'indexation de flotte. Le remplissage peut prendre un certain temps. Plus vous avez enregistré d'objets, plus le processus de remplissage est long. Une fois que vous avez activé le service d'indexation de flotte pour les groupes d'objets dynamiques, vous ne pouvez pas le désactiver tant que vous n'avez pas supprimé tous vos groupes d'objets dynamiques.

> Note
>
> Si vous disposez d'autorisations pour interroger l'index de la flotte, vous pouvez accéder aux données d'objets dans la totalité de la flotte.

Le nombre de groupes d'objets dynamiques est limité

Le nombre de groupes dynamiques est limité.

Les commandes réussies peuvent enregistrer les erreurs

Lors de la création ou de la mise à jour d'un groupe d'objets dynamiques, il est possible que certains objets soient éligibles à un groupe d'objets dynamiques sans toutefois y être ajoutés. Toutefois, la commande permettant de créer ou de mettre à jour un groupe d'objets dynamique réussit toujours dans ces cas lors de l'enregistrement d'une erreur et de la génération d'une métrique `AddThingToDynamicThingGroupsFailed` (p. 364).

UnEntrée de journalDans le journal CloudWatch est créé pour chaque objet lorsqu'il est impossible d'ajouter un objet éligible à un groupe d'objets dynamiques ou qu'un objet est supprimé d'un groupe d'objets dynamiques pour être ajouté à un autre groupe. Lorsqu'un objet ne peut pas être ajouté à un groupe dynamique, une métrique `AddThingToDynamicThingGroupsFailed` (p. 364) est également créée. Toutefois, une seule métrique peut représenter plusieurs entrées de journal.

Lorsqu'un objet devient admissible pour être ajouté à un groupe d'objets dynamiques, tenez compte des éléments suivants :

- Est-ce que l'objet est déjà contenu dans autant de groupes que possible ? (Consultez la section limites)

- NON : L'objet est ajouté au groupe d'objets dynamiques.
- YES Est-ce que l'objet est un membre d'un groupe d'objets dynamiques ?
  - NON : L'objet ne peut pas être ajouté au groupe d'objets dynamiques, une erreur est enregistrée et un `AddThingToDynamicThingGroupsFailed`Métrique (p. 364)est généré.
  - YES Le groupe d'objets dynamiques à rejoindre est-il plus ancien que n'importe quel groupe d'objets dynamiques dont l'objet est déjà membre ?
    - NON : L'objet ne peut pas être ajouté au groupe d'objets dynamiques, une erreur est enregistrée et un `AddThingToDynamicThingGroupsFailed`Métrique (p. 364)est généré.
    - YES Supprimez l'objet du groupe d'objets dynamiques le plus récent dont il est membre, enregistrez une erreur et ajoutez l'objet au groupe d'objets dynamiques. Cela génère une erreur et une métrique `AddThingToDynamicThingGroupsFailed` (p. 364) pour le groupe d'objets dynamiques duquel l'objet a été supprimé.

Lorsqu'un objet dans un groupe d'objets dynamiques ne correspond plus à la requête de recherche, celui-ci est supprimé du groupe d'objets dynamiques. De même, lorsqu'un objet est mis à jour pour correspondre à la requête de recherche d'un groupe d'objets dynamiques, il est ensuite ajouté au groupe, comme décrit ci-dessus. Ces ajouts et suppressions sont normaux et ne produisent pas d'entrées de journal des erreurs.

Si l'attribut `overrideDynamicGroups` est activé, les groupes statiques sont prioritaires par rapport aux groupes dynamiques

Le nombre de groupes auxquels un objet peut appartenir est limité. Lorsque vous mettez à jour l'appartenance d'objets à l'aide des commandes AddThingToThingGroup ou UpdateThingGroupsForThing l'ajout du paramètre `--overrideDynamicGroups` donne la priorité aux groupes d'objets statiques par rapport aux groupes d'objets dynamiques.

Lors de l'ajout d'un objet à un groupe d'objets statiques, les éléments suivants doivent être pris en compte :

- Est-ce que l'objet appartient déjà au nombre maximal de groupes ?
  - NON : La chose est ajoutée au groupe d'objets statiques.
  - YES Est-ce que l'objet est contenu dans des groupes dynamiques ?
    - NON : L'objet ne peut pas être ajouté au groupe d'objets. La commande déclenche une exception.
    - YES Était--overrideDynamicGroupsActivé
      - NON : L'objet ne peut pas être ajouté au groupe d'objets. La commande déclenche une exception.
      - YES L'objet est supprimé du groupe d'objets dynamiques le plus récent, une erreur est enregistrée et un `AddThingToDynamicThingGroupsFailed`Métrique (p. 364)est généré pour le groupe d'objets dynamiques dont l'objet a été supprimé. Ensuite, l'objet est ajouté au groupe d'objets statiques.

Les groupes d'objets dynamiques plus anciens sont prioritaires par rapport aux groupes d'objets plus récents.

Le nombre de groupes auxquels un objet peut appartenir est limité. Lorsqu'un objet devient éligible pour être ajouté à un groupe d'objets dynamiques parce qu'il s'agit d'une opération de création ou de mise à jour et que l'objet est déjà dans autant de groupes que possible, il peut être supprimé d'un autre groupe d'objets dynamiques pour permettre cet ajout. Pour plus d'informations sur la façon de procéder, consultez Les commandes réussies peuvent enregistrer les erreurs (p. 223) et Si l'attribut `overrideDynamicGroups` est activé, les groupes statiques sont prioritaires par rapport aux groupes dynamiques (p. 224) pour obtenir des exemples.

Lorsqu'un objet est supprimé d'un groupe d'objets dynamiques, une erreur est enregistrée et un événement déclenché.

Vous ne pouvez pas appliquer de stratégies à des groupes d’objets dynamiques

Le fait de tenter d’appliquer une stratégie à un groupe d’objets dynamiques génère une exception.

L'appartenance à un groupe d'objets dynamique est cohérente à terme

Seul le dernier état d'un objet est évalué pour le registre. Les états intermédiaires peuvent être ignorés s'ils sont mis à jour rapidement. Évitez d'associer une règle ou une tâche à un groupe d'objets dynamique dont l'appartenance dépend d'un état intermédiaire.

Balisage de vos ressources AWS IoT

Pour vous aider à gérer et à organiser vos groupes d'objets, vos types d'objet, vos règles de rubrique, vos tâches, vos audits planifiés ainsi que vos profils de sécurité, vous pouvez, le cas échéant, attribuer vos propres métadonnées à chacune de ces ressources sous la forme de balises. Cette section décrit les balises et vous montre comment les créer.

Pour vous aider à gérer vos coûts liés aux objets, vous pouvez créer des groupes de facturation (p. 229) qui contiennent des objets. Vous pouvez ensuite affecter des balises contenant vos métadonnées à chacun de ces groupes de facturation. Cette section décrit également les groupes de facturation, ainsi que les commandes disponibles pour les créer et les gérer.

Principes de base des balises

Vous pouvez utiliser des balises pour classer vos ressources AWS IoT de différentes manières (par exemple, par objectif, propriétaire ou environnement). Cette approche est utile lorsque vous avez de nombreuses ressources du même type : elle vous permet d'identifier rapidement une ressource 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 types d'objet vous permettant de suivre les appareils par type. Nous vous recommandons de créer un ensemble de clés de balise 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 balises que vous ajoutez ou appliquez. Vous pouvez également utiliser des balises de groupe de facturation pour classer les coûts par catégorie et en effectuer le suivi. Vous pouvez également utiliser des balises pour contrôler l'accès à vos ressources, comme décrit dans Utilisation des balises avec des stratégies IAM (p. 227).

Pour plus de facilité d'utilisation, l'Éditeur de balises dans leAWSManagement Console permet de centraliser et d'unifier la création et la gestion de vos balises. Pour de plus amples informations, veuillez consulterUtilisation de Tag EditorinUtilisation deAWSManagement Console.

Vous pouvez également gérer les balises à l'aide de l'outilAWS CLIet l'AWS IoTAPI. Vous pouvez associer des balises à des groupes d'objets, des types d'objet, des règles de rubrique, des tâches, des profils de sécurité, des stratégies ainsi que des groupes de facturation lorsque vous les créez en utilisant l'outil`Tags`dans les commandes suivantes :

- CreateBillingGroup
- CreateDestination
- CreateDeviceProfile
- CreateDynamicThingGroup
- CreateJob
- CreateOTAUpdate
- CreatePolicy
- CreateScheduledAudit
- CreateSecurityProfile
- CreateServiceProfile
- CreateStream
- CreateThingGroup

- CreateThingType
- CreateTopicRule
- CreateWirelessGateway
- CreateWirelessDevice

Vous pouvez ajouter, modifier ou supprimer des balises pour les ressources existantes qui prennent en charge le balisage à l'aide des 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.

Limites et restrictions liées aux balises

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 de balise sont sensibles à la casse.
- N'utilisez pas le`aws:`préfixe dans vos noms ou valeurs de balise. Il est réservé pour un usage par AWS. Vous ne pouvez pas modifier ou supprimer des noms ou valeurs de balise ayant ce préfixe. Les balises avec ce préfixe ne sont pas comptabilisées comme vos balises pour la limite de ressources.
- 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 autorisés incluent les lettres, les espaces et les chiffres représentables en UTF-8, ainsi que les caractères spéciaux suivants : + - = . _ : / @.

Utilisation des balises avec des stratégies IAM

Vous pouvez appliquer des autorisations de niveau ressource basées sur des balises dans les stratégies IAM que vous utilisez pour les actions d'API AWS IoT. Vous bénéficiez ainsi d'un meilleur contrôle sur les ressources qu'un utilisateur peut créer, modifier ou utiliser. Vous pouvez utiliser l'élément `Condition` (également appelé bloc `Condition`) avec les clés et valeurs de contexte de condition suivantes dans une stratégie IAM pour contrôler l'accès des utilisateurs (autorisations) en fonction des balises d'une ressource :

- Utilisez `aws:ResourceTag/`*`tag-key`*`: `*`tag-value`* pour accorder ou refuser aux utilisateurs des actions sur des ressources ayant des 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**
>
> Les clés et les valeurs de contexte de condition dans une stratégie IAM s'appliquent uniquement aux actions AWS IoT dans lesquelles un identifiant pour une ressource pouvant être balisée est un paramètre obligatoire. Par exemple, l'utilisation de DescribeEndpoint n'est pas autorisée ou refusée sur la base de clés et de valeurs de contexte de condition, car aucune ressource pouvant être balisée (groupe d'objets, type d'objet, règle de rubrique, tâche ou profil de sécurité) n'est référencée dans cette demande.

Pour plus d'informations sur l'utilisation des balises, consultezContrôle de l'accès à l'aide de balisesdans leAWS Identity and Access ManagementGuide de l'utilisateur. La section Référence de stratégie JSON IAM de ce guide fournit la syntaxe détaillée, des descriptions, ainsi que des exemples des éléments, des variables et de la logique d'évaluation des stratégies JSON dans IAM.

L'exemple de stratégie suivant applique deux restrictions basées sur des balises. Un utilisateur IAM restreint par cette stratégie :

- ne peut pas attribuer à une ressource la balise « env=prod » (dans l'exemple, voir la ligne)`"aws:RequestTag/env" : "prod"`
- ne peut pas modifier une ressource qui a une balise existante « env=prod » ou y accéder (dans l'exemple, voir la ligne)`"aws:ResourceTag/env" : "prod"`).

```
{
    "Version" : "2012-10-17",
    "Statement" : [
        {
          "Effect" : "Deny",
          "Action" : "iot:*",
          "Resource" : "*",
          "Condition" : {
            "StringEquals" : {
              "aws:RequestTag/env" : "prod"
            }
          }
        },
        {
          "Effect" : "Deny",
          "Action" : "iot:*",
          "Resource" : "*",
          "Condition" : {
            "StringEquals" : {
              "aws:ResourceTag/env" : "prod"
            }
          }
        },
        {
          "Effect": "Allow",
          "Action": [
            "iot:*"
          ],
          "Resource": "*"
        }
    ]
}
```

Vous pouvez également spécifier plusieurs valeurs de balise pour une clé de balise donnée en les plaçant dans une liste, comme suit :

```
            "StringEquals" : {
```

```
            "aws:ResourceTag/env" : ["dev", "test"]
        }
```

Note

Si vous autorisez ou refusez à des utilisateurs l'accès à des ressources en fonction de balises, vous devez envisager de refuser de manière explicite la possibilité pour les utilisateurs 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.

Groupes de facturation

AWS IoT ne vous autorise pas à appliquer directement des balises à des objets individuels, mais vous pouvez placer des objets dans des groupes de facturation et appliquer des balises à ces groupes. Pour AWS IoT, les données de répartition des coûts et d'utilisation basées sur des balises sont limitées aux groupes de facturation.

AWS IoT Core pour les ressources LoRaWan, telles que les appareils sans fil et les passerelles, ne peuvent pas être ajoutées aux groupes de facturation. Cependant, ils peuvent être associés àAWS IoTchoses, qui peuvent être ajoutées aux groupes de facturation.

Les commandes suivantes sont disponibles :

- AddThingToBillingGroup ajoute un objet à un groupe de facturation.
- CreateBillingGroup crée un groupe de facturation.
- DeleteBillingGroup supprime le groupe de facturation.
- DescribeBillingGroup renvoie des informations sur un groupe de facturation.
- ListBillingGroups répertorie les groupes de facturation que vous avez créés.
- ListThingsInBillingGroup répertorie les objets que vous avez ajoutés dans le groupe de facturation donné.
- RemoveThingFromBillingGroup supprime l'objet donné du groupe de facturation.
- UpdateBillingGroup met à jour les informations sur le groupe de facturation.
- CreateThing vous permet de spécifier un groupe de facturation pour l'objet lorsque vous le créez.
- DescribeThing renvoie la description d'un objet, y compris le groupe de facturation auquel l'objet appartient, le cas échéant.

La .AWS IoTL'API sans fil fournit ces actions pour associer des périphériques et des passerelles sans fil àAWS IoTObjets.

- AssociateWirelessDeviceWiththing
- AssociateWirelessGatewayWiththing

Affichage des données de répartition des coûts et d'utilisation

Vous pouvez utiliser des balises de groupe de facturation pour classer les coûts par catégorie et en effectuer le suivi. Lorsque vous appliquez des balises aux groupes de facturation (et donc aux éléments qu'ils incluent),AWSgénère un rapport de répartition des coûts sous forme de fichier CSV faisant apparaître l'utilisation et les coûts regroupés par les 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, consultezUtiliser les balises de répartition desdans le AWSGuide de l'utilisateur Billing and Cost Management.

> Note
>
> Pour associer avec précision les données d'utilisation et de coût aux objets que vous avez placés dans des groupes de facturation, chaque appareil ou application doit :
>
> - être enregistré en tant qu'objet dans AWS IoT, Pour plus d'informations, consultez Gestion des appareils avec AWS IoT (p. 203).
> - Connectez-vous au courtier de messages AWS IoT via MQTT en utilisant uniquement le nom de l'objet comme ID client. Pour plus d'informations, consultez the section called “Protocoles de communication de périphérique” (p. 79).
> - S'authentifier à l'aide d'un certificat client associé à l'objet.

Les dimensions de tarification suivantes sont disponibles pour les groupes de facturation (en fonction de l'activité des objets associés au groupe de facturation) :

- Connectivité (en fonction du nom d'objet utilisé comme ID client pour la connexion).
- Messagerie (en fonction des messages entrants provenant d'un objet et sortants vers un objet, MQTT uniquement).
- Opérations de shadow (en fonction de l'objet dont le message a déclenché une mise à jour de shadow).
- Règles déclenchées (selon l'objet dont le message entrant a déclenché la règle ; ne s'applique pas aux règles déclenchées par des événements de cycle de vie MQTT).
- Mises à jour d'index d'objets (en fonction de l'objet qui a été ajouté à l'index).
- Actions distantes (en fonction de l'objet mis à jour).
- Rapports Detect (p. 922) (en fonction de l'objet dont l'activité est signalée).

Les données d'utilisation et de coût basées sur des balises (et signalées pour un groupe de facturation) ne reflètent pas les activités suivantes :

- Opérations de registre d'appareils (y compris les mises à jour d'objets, de groupes d'objets et de types d'objet). Pour de plus amples informations, veuillez consulter Gestion des appareils avec AWS IoT (p. 203)).
- Mises à jour d'index de groupes d'objets (lors de l'ajout d'un groupe d'objets).
- Requêtes de recherche d'index.
- Mise en service des appareils (p. 728).
- Rapports d'Audit (p. 857).

Sécurité dans AWS IoT

Chez AWS, la sécurité du cloud est la priorité absolue. En tant que client AWS, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des organisations les plus pointilleuses en termes de sécurité.

La sécurité est une responsabilité partagée entreAWS et vous. Le modèle de responsabilité partagée décrit cette notion par les termes sécurité du cloud et sécurité dans le cloud :

- Sécurité du cloud – AWS est responsable de la protection de l'infrastructure qui exécute des services AWS dans le AWS Cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de conformité AWS. Pour en savoir plus sur les programmes de conformité qui s'appliquent à AWS IoT, consultez Services AWS concernés par le programme de conformité.
- Sécurité dans le cloud – Votre responsabilité est déterminée par le service AWS que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise, ainsi que la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS IoT. Les rubriques suivantes vous montrent comment configurer AWS IoT pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres services AWS pour surveiller et sécuriser vos ressources AWS IoT.

Rubriques

Sécurité AWS IoT

Chaque appareil ou client connecté doit disposer d'informations d'identification pour interagir avec AWS IoT. Tout le trafic à destination et en provenanceAWS IoTest envoyé en toute sécurité via Transport Layer Security (TLS).AWSLes mécanismes de sécurité du cloud protègent les données pendant leur transfert entreAWS IoTet autresAWSServices .

- Vous êtes responsable de la gestion des informations d'identification de périphérique (certificats X.509,AWSInformations d'identification, identités Amazon Cognito, identités fédérées ou jetons d'authentification personnalisés) et stratégies dansAWS IoT. Pour plus d'informations, consultez Gestion des clés dans AWS IoT (p. 315). Vous êtes responsable de l'affectation d'identités uniques à chaque appareil et de la gestion des autorisations de chaque appareil ou groupe d'appareils.
- Vos appareils se connectent àAWS IoTL'aide de certificats X.509 ou d'identités Amazon Cognito via une connexion TLS sécurisée. En phase de recherche et de développement, ainsi que pour certaines applications qui effectuent des appels d'API ou qui utilisent des WebSockets, vous pouvez également effectuer l'authentification à l'aide d'utilisateurs et de groupes IAM ou de jetons d'authentification personnalisés. Pour plus d'informations, consultez Utilisateurs, groupes et rôles IAM (p. 255).
- Lorsque vous utilisez l'authentification AWS IoT, l'agent de messages est responsable de l'authentification de vos appareils, de l'intégration sécurisée des données des appareils, ainsi que de l'octroi ou du refus des autorisations d'accès que vous spécifiez pour vos appareils à l'aide de stratégies AWS IoT.
- Lorsque vous utilisez l'authentification personnalisée, un mécanisme d'autorisation personnalisé est responsable de l'authentification de vos appareils et de l'octroi ou du refus des autorisations d'accès que vous spécifiez pour vos appareils à l'aide deAWS IoTou stratégies IAM.
- La .AWS IoTLe moteur de règles transfère les données du dispositif vers d'autres appareils ou d'autres appareilsAWSselon les règles que vous définissez. Il utilise AWS Identity and Access Management pour transférer de façon sécurisée les données vers leur destination finale. Pour plus d'informations, consultez Gestion des identités et des accès pour AWS IoT (p. 315).

Authentication

L'authentification est un mécanisme permettant de vérifier l'identité d'un client ou d'un serveur. L'authentification du serveur est le processus au cours duquel les appareils ou d'autres clients s'assurent qu'ils communiquent avec un point de terminaison AWS IoT réel. L'authentification du client est le processus au cours duquel les appareils ou d'autres clients s'authentifient eux-même auprès d'AWS IoT.

Formation et certification AWS

Suivez le cours suivant pour en savoir plus sur l'authentification dansAWS IoT : Plongez profond dansAWS IoTAuthentification et autorisation.

Présentation des certificats X.509

Les certificats X.509 sont des certificats numériques qui font appel à la norme d'infrastructure de clé publique X.509 pour associer une clé publique à une identité contenue dans un certificat. Les certificats X.509 sont émis par une entité de confiance appelée autorité de certification (CA). Celle-ci gère un ou plusieurs certificats spéciaux appelés certificats d'autorité de certification, qu'elle utilise pour émettre des certificats X.509. Seule l'autorité de certification a accès aux certificats d'autorité de certification (CA). Les chaînes de certificats X.509 sont utilisées à la fois pour l'authentification du serveur par les clients et pour l'authentification du client par le serveur.

Authentification du serveur

Lorsque votre appareil ou un autre client tente de se connecter à AWS IoT Core , le serveur AWS IoT Core envoie un certificat X.509 que votre appareil utilise pour authentifier le serveur. L'authentification est effectuée au niveau de la couche TLS par la validation de la chaîne de certificats X.509 (p. 236). C'est la même méthode que celle utilisée par votre navigateur lorsque vous visitez une adresse URL HTTPS. Si vous souhaitez utiliser des certificats de votre propre autorité de certification, veuillez consulter Gestion de vos certificats d'autorité de certification (p. 240).

Lorsque vos appareils ou d'autres clients établissent une connexion TLS à un point de terminaison AWS IoT Core , AWS IoT Core présente une chaîne de certificats que les appareils utilisent pour vérifier qu'ils communiquent avec AWS IoT Core et non avec un autre serveur qui emprunte l'identité d' AWS IoT Core . La chaîne présentée dépend de la combinaison du type de point de terminaison auquel l'appareil se connecte et de la suite de chiffrement (p. 313) négociée par le client et AWS IoT Core au cours de la négociation TLS.

Types de point de terminaison

AWS IoT Core prend en charge deux types de point de terminaison de données différents, `iot:Data` et `iot:Data-ATS`. Les points de terminaison `iot:Data` présentent un certificat signé par le certificat d'autorité de certification racine VeriSign Class 3 Public Primary G5. Les points de terminaison `iot:Data-ATS` présentent un certificat de serveur signé par une autorité de certification Amazon Trust Services.

Les certificats présentés par les points de terminaison ATS ont été signés par Starfield (signature croisée). Certaines mises en œuvre de client TLS nécessitent la validation de la racine de confiance et exigent que les certificats d'autorité de certification Starfield soient installés dans les magasins d'approbations du client.

> Warning
>
> L'utilisation d'une méthode d'épinglage de certificat qui hache l'ensemble du certificat (y compris le nom de l'émetteur, etc.) n'est pas recommandée car cela entraînera l'échec de la vérification du certificat, étant donné que les certificats ATS que nous fournissons sont signés par Starfield (signature croisée) et ont un nom d'émetteur différent.

Utilisez des points de terminaison `iot:Data-ATS`, sauf si votre appareil nécessite des certificats CA Symantec ou Verisign. Les certificats Symantec et Verisign sont obsolètes et ne sont plus pris en charge par la plupart des navigateurs Web.

Vous pouvez utiliser la commande `describe-endpoint` pour créer votre point de terminaison ATS.

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

Cette commande `describe-endpoint` renvoie un point de terminaison au format suivant.

```
account-specific-prefix.iot.your-region.amazonaws.com
```

La première fois que la commande `describe-endpoint` est appelée, un point de terminaison est créé. Tous les appels suivants de `describe-endpoint` renvoient le même point de terminaison.

Pour garantir la rétrocompatibilité, AWS IoT Core prend toujours en charge les points de terminaison Symantec. Pour de plus amples informations, veuillez consulter l'article de blog How AWS IoT Core is Helping Customers Navigate the Upcoming Distrust of Symantec Certificate Authorities. Les appareils fonctionnant sur les points de terminaison ATS sont entièrement compatibles avec les appareils fonctionnant sur les points de terminaison Symantec dans le même compte et ne nécessitent pas de réenregistrement.

> **Note**
>
> Pour afficher votre point de terminaison `iot:Data-ATS` dans la console AWS IoT Core , choisissez Paramètres. La console affiche uniquement le point de terminaison `iot:Data-ATS`. Par défaut, la commande `describe-endpoint`affiche le point de terminaison `iot:Data` pour une compatibilité descendante. Pour afficher le point de terminaison `iot:Data-ATS`, spécifiez le paramètre `--endpointType`, comme dans l'exemple précédent.

Création d'un `IotDataPlaneClient` avec le SDK AWS pour Java

Par défaut, le SDK AWS pour Java - Version 2 crée un `IotDataPlaneClient` à l'aide d'un point de terminaison `iot:Data`. Pour créer un client qui utilise un point de terminaison `iot:Data-ATS`, procédez comme suit.

- Créez un point de terminaison `iot:Data-ATS` à l'aide de l'API DescribeEndpoint .
- Spécifiez ce point de terminaison lorsque vous créez le `IotDataPlaneClient`.

L'exemple suivant exécute ces deux opérations.

```
public void setup() throws Exception {
        IotClient client =
 IotClient.builder().credentialsProvider(CREDENTIALS_PROVIDER_CHAIN).region(Region.US_EAST_1).build();
        String endpoint = client.describeEndpoint(r -> r.endpointType("iot:Data-
ATS")).endpointAddress();
        iot = IotDataPlaneClient.builder()
                                .credentialsProvider(CREDENTIALS_PROVIDER_CHAIN)
                                .endpointOverride(URI.create("https://" + endpoint))
                                .region(Region.US_EAST_1)
                                .build();
}
```

Certificats d'autorité de certification pour l'authentification du serveur

Selon le type de point de terminaison de données que vous utilisez et la suite de chiffrement négociée, les certificats d'authentification du serveur AWS IoT Core sont signés par l'un des certificats d'autorité de certification racine suivants :

Points de terminaison VeriSign (hérités)

- Clé RSA 2048 bits : Certificat racine de l'autorité de certification primaire publique de classe 3 G5 de VeriSign

Points de terminaison Amazon Trust Services (préférés)

> **Note**
>
> Vous devrez peut-être faire un clic droit sur ces liens et sélectionner Enregistrer le lien sous... pour enregistrer ces certificats sous forme de fichiers.

- Clé RSA 2048 bits :Amazon Root CA 1.
- Clé RSA 4096 bits : Amazon Root CA 2. Réservé pour un usage futur.
- Clé ECC 256 bits :Amazon Root CA 3.
- Clé ECC 384 bits : Amazon Root CA 4. Réservé pour un usage futur.

Ces certificats sont tous signés (signature croisée) par le certificat d'autorité de certification racine Starfield. Depuis le lancement d' AWS IoT Core dans la région Asie-Pacifique (Mumbai) le 9 mai 2018, toutes les nouvelles régions AWS IoT Core traitent uniquement les certificats ATS.

Instructions d'authentification du serveur

De nombreuses variables peuvent affecter la capacité d'un appareil à valider le certificat d'authentification du serveur AWS IoT Core . Par exemple, les appareils peuvent être trop limités en mémoire pour contenir tous les certificats d'autorité de certification racine possibles, ou les appareils peuvent mettre en œuvre une méthode non standard de validation de certificat. Pour ces raisons, nous suggérons de suivre les instructions suivantes :

- Nous vous recommandons d'utiliser votre point de terminaison ATS et d'installer tous lesAmazon Root CACertificats.
- Si vous ne pouvez pas stocker tous ces certificats sur votre appareil et si vos appareils n'utilisent pas la validation basée sur ECC, vous pouvez omettre l'objetAmazon Root CA 3andAmazon Root CA 4Certificats ECC. Si vos appareils n'implémentent pas la validation de certificats basée sur RSA, vous pouvez omettre l'outilAmazon Root CA 1andAmazon Root CA 2Certificats RSA. Vous devrez peut-être faire un clic droit sur ces liens et sélectionner Enregistrer le lien sous... pour enregistrer ces certificats sous forme de fichiers.
- Si vous rencontrez des problèmes de validation de certificat de serveur lorsque vous vous connectez à votre point de terminaison ATS, essayez d'ajouter le certificat d'autorité de certification racine Amazon à signature croisée pertinent à votre magasin d'approbations. Vous devrez peut-être faire un clic droit sur ces liens et sélectionner Enregistrer le lien sous... pour enregistrer ces certificats sous forme de fichiers.
  - Signature croisée Amazon Root CA 1
  - Signature croiséeAmazon Root CA 2- Réservé pour un usage futur.
  - Signature croisée Amazon Root CA 3
  - Signature croiséeAmazon Root CA 4- Réservé pour un usage futur.
- Si vous rencontrez des problèmes de validation de certificat de serveur, votre appareil devra peut-être explicitement approuver l'autorité de certification racine. Essayez d'ajouter la stratégieStarfield Root CA Certificateà votre magasin de confiance.
- Si vous rencontrez toujours des problèmes après avoir exécuté les étapes ci-dessus, contactezAWSDeveloper Support.

> Note
>
> Les certificats CA ne peuvent pas être utilisés au-delà de leur date d'expiration pour valider un certificat de serveur. Les certificats CA peuvent devoir être remplacés avant leur date d'expiration. Vérifiez que vous pouvez mettre à jour les certificats d'autorité de certification racine sur tous vos appareils ou vos clients afin d'assurer une connectivité permanente et la conformité aux bonnes pratiques de sécurité du moment.

> Note
>
> Lorsque vous vous connectez à AWS IoT Core dans le code de votre appareil, transmettez le certificat dans l'API que vous utilisez pour vous connecter. L'API que vous utilisez varie selon le kit SDK. Pour de plus amples informations, veuillez consulter Kits SDK pour les appareils AWS IoT Core (p. 1140).

Authentification client

AWS IoT prend en charge trois types de mandataire d'identité pour l'authentification de l'appareil ou du client :

- Certificats client X.509 (p. 236)
- Utilisateurs, groupes et rôles IAM (p. 255)
- Identity Amazon Cognito (p. 255)

Ces identités peuvent être utilisées avec des appareils et des applications mobiles, Web ou de bureau. Elles peuvent même être utilisées par un utilisateur saisissant des commandes d'interface de ligne de commande (CLI) AWS IoT. Typiquement,AWS IoTLes appareils utilisent des certificats X.509, alors que les applications mobiles utilisent des identités Amazon Cognito. Les applications web et de bureau utilisent des identités fédérées ou IAM.AWS CLIutilisent IAM. Pour plus d'informations sur les identités IAM, consultezGestion des identités et des accès pour AWS IoT (p. 315).

Certificats client X.509

Les certificats X.509 fournissent à AWS IoT la possibilité d'authentifier les connexions client et d'appareil. Les certificats clients doivent être enregistrés auprès d'AWS IoT avant qu'un client puisse communiquer avec AWS IoT. Un certificat client peut être enregistré dans plusieurs Compte AWS s dans le même Région AWS pour faciliter le déplacement d'appareils entre votre Compte AWS s dans la même région. Pour plus d'informations, consultez Utilisation de certificats clients X.509 dans plusieurs Compte AWS s avec enregistrement de plusieurs comptes (p. 237).

Nous vous recommandons d'attribuer à chaque appareil ou client un certificat unique de manière à permettre des actions de gestion des clients bien définies, y compris la révocation de certificats. Les appareils et les clients doivent également prendre en charge la rotation et le remplacement des certificats afin d'aider à assurer un bon fonctionnement à l'expiration de ces derniers.

Pour de plus amples informations sur l'utilisation des certificats X.509 pour la prise en charge d'un grand nombre d'appareils, veuillez consulter Mise en service des appareils (p. 728) pour prendre connaissance des différentes options de gestion des certificats et de mise en service prises en charge par AWS IoT.

AWS IoT prend en charge les types de certificat client X.509 suivants :

- Certificats X.509 générés par AWS IoT
- Certificats X.509 signés par une autorité de certification enregistrée auprès d'AWS IoT.
- Certificats X.509 signés par une autorité de certification non enregistrée auprès d'AWS IoT.

Cette section décrit comment gérer les certificats X.509 dans AWS IoT. Vous pouvez utiliser la console AWS IoT ou l'AWS CLI pour effectuer ces opérations de certificat :

- Création de certificats clients AWS IoT (p. 238)
- Création de vos propres certificats clients (p. 239)
- Enregistrement d'un certificat client (p. 244)
- Activation ou désactivation d'un certificat client (p. 249)
- Révocation d'un certificat client (p. 251)

Pour de plus amples informations sur les commandes AWS CLI permettant d'effectuer ces opérations, veuillez consulter la référence CLI AWS IoT.

Utilisation des certificats clients X.509

Les certificats X.509 authentifient les connexions de client et d'appareil à AWS IoT. Les certificats X.509 offrent plusieurs avantages par rapport à d'autres mécanismes d'identification et d'authentification. Les certificats X.509 activent des clés asymétriques à utiliser avec les appareils. Par exemple, vous pouvez graver des clés privées dans un stockage sécurisé sur un appareil afin que le matériel cryptographique sensible ne quitte jamais l'appareil. Les certificats X.509 offrent une authentification du client plus fiable que d'autres méthodes, telles que le nom d'utilisateur et le mot de passe ou les jetons de porteur, car la clé privée ne quitte jamais l'appareil.

AWS IoT authentifie les certificats client à l'aide du mode d'authentification du client du protocole TLS. La prise en charge de TLS est disponible dans de nombreux langages de programmation et systèmes d'exploitation ; ce protocole est couramment utilisée pour le chiffrement des données. Dans l'authentification client TLS,AWS IoTdemande un certificat client X.509 et valide l'état du certificat et Compte AWS contre un registre de certificats. Il interroge ensuite le client pour obtenir la preuve de propriété de la clé privée qui correspond à la clé publique contenue dans le certificat. AWS IoT exige que les clients envoient l'extension SNI (Server Name Indication) au protocole TLS (Transport Layer Security). Pour de plus amples informations sur la configuration de l'extension SNI, veuillez consulter Sécurité du transport dans AWS IoT (p. 313).

Les certificats X.509 peuvent être vérifiés par rapport à une autorité de certification approuvée. Vous pouvez créer des certificats clients qui utilisent l'autorité de certification racine Amazon et utiliser vos propres certificats clients signés par une autre autorité de certification. Pour plus d'informations sur l'utilisation de vos propres certificats X.509, consultez Création de vos propres certificats clients (p. 239).

La date et l'heure d'expiration des certificats signés par un certificat d'autorité de certification sont définies au moment de la création du certificat. Les certificats X.509 générés par AWS IoT expirent à minuit (heure UTC) le 31 décembre 2049 (2049-12-31T23:59:59 Z). Pour de plus amples informations sur l'utilisation de la console AWS IoT pour créer des certificats utilisant l'autorité de certification racine Amazon, veuillez consulter Création de certificats clients AWS IoT (p. 238).

Utilisation de certificats clients X.509 dans plusieurs Compte AWS s avec enregistrement de plusieurs comptes

L'inscription à plusieurs comptes permet de déplacer des appareils entre votre Compte AWS s dans la même Région ou dans différentes Régions. Avec cela, vous pouvez enregistrer, tester et configurer un périphérique dans un compte de pré-production, puis enregistrer et utiliser le même périphérique et le même certificat de périphérique dans un compte de production. Vous pouvez également enregistrer le certificat client sur le périphérique (les certificats de périphérique) sans une autorité de certification (p. 247) enregistrée auprès de AWS IoT.

> Note
>
> Les certificats utilisés pour l'enregistrement multi compte sont pris en charge sur le`iot:Data-ATS`,`iot:Data`(hérité), et`iot:Jobs`Types de point de terminaison. Les certificats utilisés pour l'enregistrement multi compte ne peuvent pas être utilisés sur le`iot:CredentialProvider`Type de point de terminaison. Pour plus d'informations surAWS IoTpoints de terminaison de l'appareil, consultezAWS IoTdonnées de périphérique et points de terminaison de service (p. 76).

Les périphériques qui utilisent l'enregistrement multi-compte doivent envoyer l'extension SNI (Server Name Indication) au protocole TLS (Transport Layer Security) et fournir l'adresse complète du point de terminaison dans le champ `host_name`, lorsqu'ils se connectent à AWS IoT. AWS IoT utilise l'adresse du point de terminaison dans `host_name` pour acheminer la connexion vers le compte AWS IoT correct. Les périphériques existants qui n'envoient pas d'adresse de point de terminaison valide dans `host_name` continueront de fonctionner, mais ils ne pourront pas utiliser les fonctionnalités qui nécessitent ces informations. Pour de plus amples informations sur l'extension SNI et pour savoir comment identifier l'adresse du point de terminaison pour le champ `host_name`, veuillez consulter Sécurité du transport dans AWS IoT (p. 313).

Pour utiliser l'enregistrement de plusieurs comptes

1. N'enregistrez pas l'autorité de certification qui a signé les certificats d'appareil avec AWS IoT.
2. Enregistrez les certificats d'appareil sans autorité de certification. Voir Enregistrement d'un certificat client signé par une autorité de certification non enregistrée (CLI) (p. 247).
3. Utilisez le bon `host_name` dans l'extension SNI de TLS lorsque l'appareil se connecte à AWS IoT. Voir Sécurité du transport dans AWS IoT (p. 313).

Algorithmes de signature de certificat pris en charge par AWS IoT

AWS IoT prend en charge les algorithmes de signature de certificat suivants :

- SHA256WITHRSA
- SHA384WITHRSA
- SHA512WITHRSA
- DSA_WITH_SHA256
- ECDSA-WITH-SHA256
- ECDSA-WITH-SHA384
- ECDSA-WITH-SHA512

Création de certificats clients AWS IoT

AWS IoT fournit des certificats clients signés par l'autorité de certification racine Amazon.

Cette rubrique décrit comment créer un certificat client signé par l'autorité de certification racine Amazon et télécharger les fichiers de certificat. Après avoir créé les fichiers de certificat client, vous devez les installer sur le client.

> Note
>
> Chaque certificat client X.509 fourni parAWS IoTcontient les attributs de l'émetteur et du sujet qui sont définis au moment de la création du certificat. Les attributs de certificat ne sont immuables qu'après la création du certificat.

Vous pouvez utiliser la console AWS IoT ou l'AWS CLI pour créer un certificat AWS IoT signé par l'autorité de certification racine Amazon.

Création d'un certificat AWS IoT (console)

Pour créer un certificat AWS IoT à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console, puis ouvrez l'AWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez successivement Sécurité, Certificats et Créer.
3. Choisissez Création d'un certificat en un clic (recommandé) - Créer un certificat.
4. À partir de la page Certificat créé !, téléchargez les fichiers du certificat client pour l'objet, la clé publique et la clé privée sur un emplacement sécurisé.

   Si vous avez aussi besoin du fichier de certificat de l'autorité de certification racine Amazon, cette page contient également le lien vers la page à partir de laquelle vous pouvez le télécharger.
5. Un certificat client est désormais créé et enregistré auprès d'AWS IoT. Vous devez activer le certificat avant de l'utiliser dans un client.

ChoisissezActivatePour activer le certificat client maintenant. Si vous ne souhaitez pas activer le certificat maintenant, Activation d'un certificat client (console) (p. 249) décrit comment l'activer ultérieurement.

6. Si vous souhaitez attacher une stratégie au certificat, choisissezAttacher une stratégie.

   Si vous ne souhaitez pas attacher de stratégie maintenant, choisissezTerminéPour terminer. Vous pouvez attacher une stratégie ultérieurement.

Après avoir terminé la procédure, installez les fichiers de certificat sur le client.

Création d'un certificat AWS IoT (CLI)

L'AWS CLI fournit la commande create-keys-and-certificate qui permet de créer des certificats clients signés par l'autorité de certification racine Amazon. Toutefois, cette commande ne télécharge pas le fichier de certificat de l'autorité de certification racine Amazon. Vous pouvez télécharger le fichier de certificat de l'autorité de certification racine Amazon à partir de Certificats d'autorité de certification pour l'authentification du serveur (p. 234).

Cette commande crée les fichiers de clé privée, de clé publique et de certificat X.509. Elle enregistre et active également le certificat auprès d'AWS IoT.

```
aws iot create-keys-and-certificate \
    --set-as-active \
    --certificate-pem-outfile certificate_filename \
    --public-key-outfile public_key_filename \
    --private-key-outfile private_key_filename
```

Si vous ne souhaitez pas activer le certificat lorsque vous le créez et l'enregistrez, cette commande crée les fichiers de clé privée, de clé publique et de certificat X.509, enregistre le certificat, mais ne l'active pas. Activation d'un certificat client (CLI) (p. 250) décrit comment activer le certificat ultérieurement.

```
aws iot create-keys-and-certificate \
    --no-set-as-active \
    --certificate-pem-outfile certificate_filename \
    --public-key-outfile public_key_filename \
    --private-key-outfile private_key_filename
```

Installez les fichiers de certificat sur le client.

Création de vos propres certificats clients

AWS IoT prend en charge les certificats clients signés par d'autres autorités de certification racine. Vous pouvez enregistrer des certificats clients signés par une autre autorité de certification racine. Toutefois, si vous souhaitez que l'appareil ou le client enregistre son certificat client lors de sa première connexion à AWS IoT, l'autorité de certification racine doit être enregistrée auprès d'AWS IoT.

> **Note**
>
> Un certificat d'autorité de certification peut être enregistré par un seul compte dans une région.

Pour plus d'informations sur l'utilisation des certificats X.509 pour la prise en charge de plusieurs appareils, veuillez consulter Mise en service des appareils (p. 728) pour prendre connaissance des différentes options de gestion et de mise en service des certificats prises en charge par AWS IoT.

Rubriques

Gestion de vos certificats d'autorité de certification

Cette section décrit les tâches courantes de gestion de vos propres certificats d'autorité de certification.

Création d'un certificat d'autorité de certification

Si vous n'avez pas de certificat CA, vous pouvez utiliserOpenSSL v1.1.1ipour en créer un.

> **Note**
>
> Vous ne pouvez pas effectuer cette procédure dans la console AWS IoT.

Pour créer un certificat d'autorité de certification à l'aideOpenSSL v1.1.1iOutils

1. Générez une paire de clés.

```
openssl genrsa -out root_CA_key_filename 2048
```

2. Utilisez la clé privée de la paire de clés pour générer un certificat CA.

```
openssl req -x509 -new -nodes \
    -key root_CA_key_filename \
    -sha256 -days 1024 \
    -out root_CA_pem_filename
```

Enregistrement de votre certificat d'autorité de certification

Vous devrez peut-être enregistrer votre autorité de certification auprès d'AWS IoT si vous utilisez des certificats clients signés par une autorité de certification non reconnue par AWS IoT.

Si vous souhaitez que les clients enregistrent automatiquement leurs certificats clients auprès d'AWS IoT lors de leur première connexion, l'autorité de certification qui a signé les certificats clients doit être enregistrée auprès d'AWS IoT. Sinon, vous n'avez pas besoin d'enregistrer le certificat de l'autorité de certification qui a signé les certificats clients.

> **Note**
>
> Un certificat d'autorité de certification peut être enregistré par un seul compte dans une région.

Enregistrement d'un certificat d'autorité de certification (console)

> **Note**
>
> Assurez-vous de disposer du fichier de certificat et du fichier de clé privée de l'autorité de certification racine avant de commencer.

Cette procédure nécessite également l'utilisation de l'interface de ligne de commande pour exécuter des commandes OpenSSL v1.1.1i.

Pour enregistrer un certificat d'autorité de certification à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le panneau de navigation de gauche, choisissezSecure, choisissezAC, puisS'inscrire.
3. DansSélectionner une autorité de certification, choisissezEnregistrer CA.
4. Dans Enregistrer un certificat d'autorité de certification, suivez les étapes affichées.

   Les étapes 1 à 4 sont effectuées dans l'interface de ligne de commande.

   Les étapes 5 et 6 requièrent les fichiers créés aux étapes 3 et 4.
5. Si vous souhaitez activer ce certificat lorsque vous l'enregistrez, cochez Activer le certificat de CA.

   Le certificat de l'autorité de certification doit être actif avant que vous puissiez enregistrer les certificats clients signés par celui-ci.
6. Si vous souhaitez activer le certificat pour enregistrer automatiquement des certificats clients signés par celui-ci, sélectionnez Activer l'enregistrement automatique des certificats d'appareils.
7. ChoisissezEnregistrement d'un certificat d'autoritéPour compléter l'enregistrement.

Le certificat de l'autorité de certification apparaît dans la liste des autorités de certification avec son statut actuel.

Enregistrement d'un certificat d'autorité de certification (CLI)

Note

Assurez-vous de disposer du fichier de certificat et du fichier de clé privée de l'autorité de certification racine avant de commencer.

Pour enregistrer un certificat d'autorité de certification à l'aide de l'AWS CLI

1. Utilisez get-registration-code pour obtenir un code d'enregistrement auprès d'AWS IoT. Enregistrez le `registrationCode` renvoyé à utiliser en tant que `Common Name` du certificat de vérification de clé privée.

```
aws iot get-registration-code
```

2. Générez une paire de clés pour le certificat de vérification de clé privée :

```
openssl genrsa -out verification_cert_key_filename 2048
```

3. Créez une demande de signature de certificat (CSR) pour le certificat de vérification de clé privée. Dans le champ `Common Name` du certificat, indiquez la valeur `registrationCode` renvoyée par get-registration-code.

```
openssl req -new \
    -key verification_cert_key_filename \
    -out verification_cert_csr_filename
```

Vous êtes invité à entrer certaines informations, notamment le `Common Name` du certificat.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
```

```
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name) []:your_registration_code
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

4. Utilisez la CSR pour créer un certificat de vérification de clé privée :

```
openssl x509 -req \
                                        -in verification_cert_csr_filename \
    -CA root_CA_pem_filename \
    -CAkey root_CA_key_filename \
    -CAcreateserial \
    -out verification_cert_pem_filename \
    -days 500 -sha256
```

5. Enregistrez le certificat CA auprès d'AWS IoT. Transmettez le nom de fichier du certificat de l'autorité de certification et le nom de fichier du certificat de vérification de clé privée à la commande register-ca-certificate :

```
aws iot register-ca-certificate \
    --ca-certificate file://root_CA_pem_filename \
    --verification-cert file://verification_cert_pem_filename
```

   Cette commande, si elle aboutit, renvoie l'*ID de certificat*.

6. À ce stade, le certificat d'autorité de certification a été enregistré auprès d'AWS IoT, mais n'est pas actif. Le certificat de l'autorité de certification doit être actif avant que vous puissiez enregistrer les certificats clients signés par celui-ci.

   Cette étape active le certificat de l'autorité de certification.

   Utilisez la commande CLI update-certificate pour activer le certificat de l'autorité de certification :

```
aws iot update-ca-certificate \
    --certificate-id certificateId \
    --new-status ACTIVE
```

Utilisez la commande describe-ca-certificate pour voir le statut du certificat d'autorité de certification.

Désactivation d'un certificat d'autorité de certification

Lorsqu'un certificat d'autorité de certification est activé pour l'enregistrement automatique du certificat client, AWS IoT vérifie le certificat de l'autorité de certification utilisé pour signer le certificat client afin de s'assurer que l'autorité de certification est `ACTIVE`. Si le certificat de l'autorité de certification est `INACTIVE`, AWS IoT n'autorise pas l'enregistrement du certificat client.

En définissant le certificat de l'autorité de certification comme `INACTIVE`, vous empêchez l'enregistrement automatique de tout nouveau certificat client émis par l'autorité de certification.

Note

Les certificats clients enregistrés qui ont été signés par le certificat d'autorité de certification compromis continuent de fonctionner jusqu'à ce que vous les révoquiez explicitement.

Désactivation d'un certificat d'autorité de certification (console)

Pour désactiver un certificat d'autorité de certification à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis CA.
3. Dans la liste des autorités de certification, recherchez celle que vous souhaitez désactiver et ouvrez le menu d'options à l'aide de l'icône représentant des points de suspension.
4. Dans le menu d'options, choisissez Désactiver.

L'autorité de certification doit apparaître comme Inactive dans la liste.

Note

La console AWS IoT ne permet pas de répertorier les certificats signés par l'autorité de certification que vous avez désactivée. Pour connaître l'option d'AWS CLI permettant de répertorier ces certificats, veuillez consulter Désactivation d'un certificat d'autorité de certification (CLI) (p. 243).

Désactivation d'un certificat d'autorité de certification (CLI)

L'AWS CLI fournit la commande update-ca-certificate pour désactiver un certificat d'autorité de certification.

```
aws iot update-ca-certificate \
    --certificate-id certificateId \
    --new-status INACTIVE
```

Utilisez la commande list-certificates-by-ca pour obtenir la liste de tous les certificats clients enregistrés qui ont été signés par l'autorité de certification spécifiée. Pour chaque certificat client signé par le certificat d'autorité de certification spécifié, utilisez la commande update-certificate pour révoquer le certificat client afin d'empêcher son utilisation.

Utilisez la commande describe-ca-certificate pour voir le statut du certificat d'autorité de certification.

Création d'un certificat client à l'aide de votre certificat d'autorité de certification

Vous pouvez utiliser votre propre autorité de certification pour créer des certificats clients. Le certificat client doit être enregistré auprès d'AWS IoT avant d'être utilisé. Pour de plus amples informations sur les options d'enregistrement pour vos certificats clients, veuillez consulter Enregistrement d'un certificat client (p. 244).

Création d'un certificat client (CLI)

Note

Vous ne pouvez pas effectuer cette procédure dans la console AWS IoT.

Pour créer un certificat client à l'aide de l'AWS CLI

1. Générez une paire de clés.

```
openssl genrsa -out device_cert_key_filename 2048
```

2. Créez une demande de signature de certificat (CSR) pour le certificat client.

```
openssl req -new \
    -key device_cert_key_filename \
    -out device_cert_csr_filename
```

Vous êtes invité à entrer certaines informations, comme illustré ici :

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:
    State or Province Name (full name) []:
    Locality Name (for example, city) []:
    Organization Name (for example, company) []:
    Organizational Unit Name (for example, section) []:
    Common Name (e.g. server FQDN or YOUR name) []:
    Email Address []:

    Please enter the following 'extra' attributes
    to be sent with your certificate request
    A challenge password []:
    An optional company name []:
```

3. Créez un certificat client à partir de la CSR.

```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_CA_pem_filename \
    -CAkey root_CA_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 500 -sha256
```

À ce stade, le certificat client a été créé, mais il n'a pas encore été enregistré auprès d'AWS IoT. Pour de plus amples informations sur la façon et le moment d'enregistrer le certificat client, veuillez consulter Enregistrement d'un certificat client (p. 244).

Enregistrement d'un certificat client

Les certificats clients doivent être enregistrés auprès d'AWS IoT pour activer les communications entre le client et AWS IoT. Vous pouvez enregistrer chaque certificat client manuellement ou configurer les certificats clients de sorte qu'ils soient enregistrés automatiquement lorsque le client se connecte à AWS IoT pour la première fois.

Si vous souhaitez que vos clients et appareils enregistrent leurs certificats clients lors de leur première connexion, vous devez utiliser Enregistrement de votre certificat d'autorité de certification (p. 240) pour signer le certificat client auprès d'AWS IoT dans les régions où vous souhaitez l'utiliser. L'autorité de certification racine Amazon est automatiquement enregistrée auprès d'AWS IoT.

Les certificats clients peuvent être partagés par Compte AWS s et régions. Les procédures décrites dans ces rubriques doivent être effectuées dans chaque compte et chaque région où vous souhaitez

utiliser le certificat client. L'enregistrement d'un certificat client dans un compte ou une région n'est pas automatiquement reconnu par un autre compte ou une autre région.

> **Note**
>
> Les clients qui utilisent le protocole TLS (Transport Layer Security) pour se connecter à AWS IoT doivent prendre en charge l'extension SNI (Server Name Indication) de TLS. Pour plus d'informations, consultez Sécurité du transport dans AWS IoT (p. 313).

Rubriques

- Enregistrement manuel d’un certificat client (p. 245)
- Enregistrement d’un certificat client lorsque le client se connecte à AWS IoT (enregistrement juste-à-temps) (p. 247)

Enregistrement manuel d’un certificat client

Vous pouvez enregistrer manuellement un certificat client à l'aide de la console AWS IoT et de l’AWS CLI.

La procédure d'enregistrement à utiliser dépend si le certificat sera partagé par Compte AWS s et régions. L'enregistrement d'un certificat client dans un compte ou une région n'est pas automatiquement reconnu par un autre compte ou une autre région.

Les procédures décrites dans cette rubrique doivent être effectuées dans chaque compte et chaque région où vous souhaitez utiliser le certificat client. Les certificats clients peuvent être partagés par Compte AWS s et régions, mais uniquement si le certificat client est signé par une autorité de certification qui n'est PAS enregistrée auprès d'.AWS IoT.

Enregistrement d'un certificat client signé par une autorité de certification enregistrée (console)

> **Note**
>
> Avant d'effectuer cette procédure, assurez-vous de disposer du fichier .pem du certificat client et vérifiez que le certificat client a été signé par une autorité de certification que vous avez enregistrée auprès d’AWS IoT (p. 240).

Pour enregistrer un certificat existant auprès d’AWS IoT à l'aide de la console

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez successivement Sécurité, Certificats et Créer.
3. SurCréer un certificat, localisez l'Utiliser mon certificat, puis choisissezMise en route.
4. Sur Sélectionner une CA :

   - si les certificats clients sont signés par une autorité de certification enregistrée auprès d’AWS IoT

     Choisissez cette autorité de certification dans la liste, puis choisissezSuivant.
   - Si les certificats clients ne sont pas signés par une autorité de certification enregistrée auprès d’AWS IoT

     Voir Enregistrement d'un certificat client signé par une autorité de certification non enregistrée (console) (p. 246).
   - Si les certificats client sont signés par l'autorité de certification d'Amazon

     Ne sélectionnez aucune autorité de certification, choisissez simplementSuivant.

   Si les certificats clients ne sont pas signés par une autorité de certification enregistrée auprès d’AWS IoT, veuillez consulter Enregistrement d'un certificat client signé par une autorité de certification non enregistrée (console) (p. 246).

5. SurEnregistrer les certificats de périphérique existants, choisissezSélectionner des certificatset sélectionnez jusqu'à 10 fichiers de certificats à enregistrer.
6. Après avoir fermé la boîte de dialogue du fichier, indiquez si vous souhaitez activer ou révoquer les certificats clients lorsque vous les enregistrez.

   Si vous n'activez pas un certificat lors de son enregistrement, Activation d’un certificat client (console) (p. 249) décrit comment l'activer ultérieurement.

   Si un certificat est révoqué lors de son enregistrement, il ne peut pas être activé ultérieurement.

   Après avoir sélectionné les fichiers de certificat à enregistrer et sélectionné les actions à effectuer après l'enregistrement, sélectionnezEnregistrer les certificats.

Les certificats clients enregistrés apparaissent dans la liste des certificats.

Enregistrement d'un certificat client signé par une autorité de certification non enregistrée (console)

Note

Avant d'effectuer cette procédure, assurez-vous de disposer du fichier .pem du certificat client.

Pour enregistrer un certificat existant auprès d’AWS IoT à l'aide de la console

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez successivement Sécurité, Certificats et Créer.
3. SurCréer un certificat, localisez l'Utiliser mon certificat, puis choisissezMise en route.
4. SurSélectionner une autorité de certification, choisissezSuivant.
5. SurEnregistrer les certificats de périphérique existants, choisissezSélectionner des certificatset sélectionnez jusqu'à 10 fichiers de certificats à enregistrer.
6. Après avoir fermé la boîte de dialogue du fichier, indiquez si vous souhaitez activer ou révoquer les certificats clients lorsque vous les enregistrez.

   Si vous n'activez pas un certificat lors de son enregistrement, Activation d’un certificat client (console) (p. 249) décrit comment l'activer ultérieurement.

   Si un certificat est révoqué lors de son enregistrement, il ne peut pas être activé ultérieurement.

   Après avoir sélectionné les fichiers de certificat à enregistrer et sélectionné les actions à effectuer après l'enregistrement, sélectionnezEnregistrer les certificats.

Les certificats clients enregistrés apparaissent dans la liste des certificats.

Enregistrement d'un certificat client signé par une autorité de certification enregistrée (CLI)

Note

Avant d'effectuer cette procédure, assurez-vous de disposer du fichier .pem de l'autorité de certification et du fichier .pem du certificat client. Le certificat client doit être signé par une autorité de certification que vous avez enregistrée auprès d’AWS IoT (p. 240).

Utilisez la commande register-certificate pour enregistrer un certificat client sans l’activer.

```
aws iot register-certificate \
    --certificate-pem file://device_cert_pem_filename \
    --ca-certificate-pem file://ca_cert_pem_filename
```

Le certificat client est enregistré auprès d'AWS IoT, mais il n'est pas encore actif. Veuillez consulter Activation d'un certificat client (CLI) (p. 250) pour de plus amples informations sur la façon de l'activer ultérieurement.

Vous pouvez également activer le certificat client lorsque vous l'enregistrez à l'aide de cette commande.

```
aws iot register-certificate \
    --set-as-active \
    --certificate-pem file://device_cert_pem_filename \
    --ca-certificate-pem file://ca_cert_pem_filename
```

Pour plus d'informations sur l'activation du certificat afin qu'il puisse être utilisé pour la connexion à AWS IoT, veuillez consulter Activation ou désactivation d'un certificat client (p. 249)

Enregistrement d'un certificat client signé par une autorité de certification non enregistrée (CLI)

> **Note**
>
> Avant d'effectuer cette procédure, assurez-vous de disposer du fichier .pem du certificat.

Utilisez la commande register-certificate-without-ca pour enregistrer un certificat client sans l'activer.

```
aws iot register-certificate-without-ca \
    --certificate-pem file://device_cert_pem_filename
```

Le certificat client est enregistré auprès d'AWS IoT, mais il n'est pas encore actif. Veuillez consulter Activation d'un certificat client (CLI) (p. 250) pour de plus amples informations sur la façon de l'activer ultérieurement.

Vous pouvez également activer le certificat client lorsque vous l'enregistrez à l'aide de cette commande.

```
aws iot register-certificate-without-ca \
    --status ACTIVE \
    --certificate-pem file://device_cert_pem_filename
```

Pour de plus amples informations sur l'activation du certificat afin qu'il puisse être utilisé pour la connexion àAWS IoT, voirActivation ou désactivation d'un certificat client (p. 249).

Enregistrement d'un certificat client lorsque le client se connecte à AWS IoT (enregistrement juste-à-temps)

Vous pouvez configurer un certificat d'autorité de certification de manière à ce que les certificats clients qu'il a signés s'enregistrent automatiquement auprès d'AWS IoT lors de la première connexion du client à AWS IoT.

Pour enregistrer des certificats client lorsqu'un client se connecte à AWS IoT pour la première fois, vous devez activer l'enregistrement automatique du certificat d'autorité de certification et configurer la première connexion par le client de manière à fournir les certificats requis.

Configuration d'un certificat d'autorité de certification pour la prise en charge de l'enregistrement automatique (console)

Pour configurer un certificat d'autorité de certification de sorte qu'il prenne en charge l'enregistrement automatique du certificat client à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis CA.

3. Dans la liste des autorités de certification, recherchez celle pour laquelle vous souhaitez activer l'enregistrement automatique et ouvrez le menu d'options à l'aide de l'icône représentant des points de suspension.
4. Dans le menu d'options, choisissez Activer l'enregistrement automatique.

   Note

   Le statut d'enregistrement automatique n'apparaît pas dans la liste des autorités de certification. Pour voir le statut d'enregistrement automatique d'une autorité de certification, vous devez ouvrir la page Détails de l'autorité de certification.

Configuration d'un certificat d'autorité de certification pour la prise en charge de l'enregistrement automatique (CLI)

Si vous avez déjà enregistré votre certificat d'autorité de certification auprès d'AWS IoT, utilisez la commande update-ca-certificate pour définir `autoRegistrationStatus` du certificat d'autorité de certification sur `ENABLE`.

```
aws iot update-ca-certificate \
--certificate-id caCertificateId \
--new-auto-registration-status ENABLE
```

Si vous souhaitez activer `autoRegistrationStatus` lors de l'enregistrement du certificat d'autorité de certification, utilisez la commande register-ca-certificate.

```
aws iot register-ca-certificate \
--allow-auto-registration  \
--ca-certificate file://root_CA_pem_filename \
--verification-cert file://verification_cert_pem_filename
```

Utilisez la commande describe-ca-certificate pour voir le statut du certificat d'autorité de certification.

Configuration de la première connexion par un client pour l'enregistrement automatique

Lorsqu'un client tente de se connecter à AWS IoT pour la première fois, il doit présenter un fichier contenant à la fois votre certificat d'autorité de certification enregistré et le certificat client signé par votre certificat d'autorité de certification dans le cadre de la poignée de main TLS. Vous pouvez combiner les deux fichiers à l'aide d'une commande comme la suivante :

```
cat device_cert_filename ca_certificate_pem_filename > combined_filename
```

Lorsque le client se connecte à AWS IoT, utilisez le fichier *combined_filename* en tant que fichier de certificat. AWS IoT reconnaît le certificat d'autorité de certification en tant que certificat d'autorité de certification enregistré, enregistre le certificat client et définit son statut sur `PENDING_ACTIVATION`. Cela signifie que le certificat client a été enregistré automatiquement et qu'il est en attente d'activation. L'état du certificat client doit être `ACTIVE` avant de pouvoir être utilisé pour la connexion à AWS IoT.

Lorsqu'AWS IoT enregistre automatiquement un certificat ou lorsqu'un client présente un certificat avec le statut `PENDING_ACTIVATION`, AWS IoT publie un message dans la rubrique MQTT suivante :

`$aws/events/certificates/registered/`*caCertificateId*

Où *caCertificateId* est l'ID du certificat d'autorité de certification ayant émis le certificat client.

Le message publié sur cette rubrique a la structure suivante :

```
{
        "certificateId": "certificateId",
        "caCertificateId": "caCertificateId",
        "timestamp": timestamp,
        "certificateStatus": "PENDING_ACTIVATION",
        "awsAccountId": "awsAccountId",
        "certificateRegistrationTimestamp": "certificateRegistrationTimestamp"
}
```

Vous pouvez créer une règle qui écoute cette rubrique et effectue certaines actions. Nous vous recommandons de créer une règle Lambda qui vérifie que le certificat client ne figure pas sur une liste de révocation de certificat (CRL), qui active le certificat et crée et attache une stratégie au certificat. La stratégie détermine les ressources auxquelles le client peut accéder. Pour plus d'informations sur la création d'une règle Lambda qui écoute sur le`$aws/events/certificates/registered/caCertificateID`et effectue ces actions, voirEnregistrement juste-à-temps des certificats clients surAWS IoT.

Si une erreur ou une exception se produit pendant l'enregistrement automatique des certificats clients,AWS IoTenvoie des événements ou des messages à vos journaux dans CloudWatch Logs. Pour plus d'informations sur la configuration des journaux pour votre compte, consultez leDocumentation Amazon CloudWatch.

Activation ou désactivation d’un certificat client

AWS IoT vérifie qu'un certificat client est actif lorsqu'il authentifie une connexion.

Vous pouvez créer et enregistrer des certificats clients sans les activer afin qu'ils ne puissent pas être utilisés tant que vous ne le souhaitez pas. Vous pouvez également désactiver temporairement des certificats clients actifs. Enfin, vous pouvez révoquer des certificats clients pour empêcher toute future utilisation.

Activation d’un certificat client (console)

Pour activer un certificat client à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.
3. Dans la liste des certificats, recherchez le certificat que vous souhaitez activer et ouvrez le menu d'options à l'aide de l'icône représentant des points de suspension.
4. Dans le menu d'options, choisissez Activer.

Le certificat doit apparaître comme Active dans la liste des certificats.

Désactivation d’un certificat client (console)

Pour désactiver un certificat client à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.
3. Dans la liste des certificats, recherchez le certificat que vous souhaitez désactiver et ouvrez le menu d'options à l'aide de l'icône représentant des points de suspension.
4. Dans le menu d'options, choisissez Désactiver.

Le certificat doit apparaître comme Inactive dans la liste des certificats.

Activation d'un certificat client (CLI)

L'AWS CLI fournit la commande update-certificate pour activer un certificat.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status ACTIVE
```

Si la commande aboutit, le statut du certificat devient `ACTIVE`. Exécutez describe-certificate pour voir le statut du certificat.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Désactivation d'un certificat client (CLI)

L'AWS CLI fournit la commande update-certificate pour désactiver un certificat.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status INACTIVE
```

Si la commande aboutit, le statut du certificat devient `INACTIVE`. Exécutez describe-certificate pour voir le statut du certificat.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Attacher un objet ou une stratégie à un certificat client

Lorsque vous créez et enregistrez un certificat séparé d'un objet AWS IoT, il n'aura aucune stratégie qui autorise des opérations AWS IoT et il ne sera associé à aucun objet d'objet AWS IoT. Cette section décrit comment ajouter ces relations à un certificat enregistré.

> **Important**
>
> Pour effectuer ces procédures, vous devez avoir déjà créé l'objet ou la stratégie que vous souhaitez attacher au certificat.

Le certificat authentifie un périphérique avecAWS IoTafin qu'il puisse se connecter. L'attachement du certificat à une ressource de chose établit la relation entre le périphérique (par le biais du certificat) et la ressource de chose. Pour autoriser le périphérique à effectuerAWS IoT, par exemple pour permettre à l'appareil de se connecter et de publier des messages, une stratégie appropriée doit être attachée au certificat de l'appareil.

Attacher un objet à un certificat client (console)

Vous aurez besoin du nom de l'objet d'objet pour réaliser cette procédure.

Pour attacher un objet d'objet à un certificat enregistré

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

3. Dans la liste des certificats, recherchez le certificat auquel vous souhaitez attacher une stratégie, ouvrez le menu d'options du certificat en choisissant l'icône représentant trois points de suspension, puis choisissez Attacher un objet.
4. Dans la fenêtre contextuelle, recherchez le nom de l'objet à attacher au certificat, cochez sa case et choisissezAttacher.

L'objet d'objet doit désormais apparaître dans la liste des objets sur la page de détails du certificat.

Attacher une stratégie à un certificat client (console)

Vous aurez besoin du nom de l'objet de stratégie pour réaliser cette procédure.

Pour attacher un objet de stratégie à un certificat enregistré

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.
3. Dans la liste des certificats, recherchez le certificat auquel vous souhaitez attacher une stratégie, ouvrez le menu d'options du certificat en choisissant l'icône représentant trois points de suspension, puis choisissez Attacher une stratégie.
4. Dans la fenêtre contextuelle, recherchez le nom de la stratégie à attacher au certificat, cochez sa case et choisissezAttacher.

L'objet de stratégie doit désormais apparaître dans la liste des stratégies de la page de détails du certificat.

Attacher un objet à un certificat client (interface de ligne de commande)

L'AWS CLI fournit la commande attach-thing-principal pour attacher un objet d'objet à un certificat.

```
aws iot attach-thing-principal \
    --principal certificateArn \
    --thing-name thingName
```

Attacher une stratégie à un certificat client (interface de ligne de commande)

L'AWS CLI fournit la commande attach-policy pour attacher un objet de stratégie à un certificat.

```
aws iot attach-policy \
    --target certificateArn \
    --policy-name policyName
```

Révocation d'un certificat client

Si vous détectez une activité suspecte sur un certificat client enregistré, vous pouvez révoquer celui-ci afin qu'il ne puisse plus être utilisé.

Révocation d'un certificat client (console)

Pour révoquer un certificat client à l'aide de la console AWS IoT

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.
3. Dans la liste des certificats, recherchez le certificat que vous souhaitez révoquer et ouvrez le menu d'options à l'aide de l'icône représentant des points de suspension.

4. Dans le menu d'options, choisissez Révoquer.

Si la révocation aboutit, il s'affichera comme Revoked (Révoqué) dans la liste des certificats.

Révocation d'un certificat client (CLI)

L'AWS CLI fournit la commande update-certificate pour révoquer un certificat.

```
aws iot update-certificate \
    --certificate-id certificateId \
    --new-status REVOKED
```

Si la commande aboutit, le statut du certificat devient `REVOKED`. Exécutez describe-certificate pour voir le statut du certificat.

```
aws iot describe-certificate \
    --certificate-id certificateId
```

Transférer un certificat vers un autre compte

Certificats X.509 qui appartiennent à un Compte AWS peut être transféré à un autre Compte AWS .

Pour transférer un certificat X.509 à partir d'un Compte AWS à un autre

1. the section called "Commencer un transfert de certificat" (p. 252)

   Le certificat doit être désactivé et détaché de toutes les stratégies et choses avant de lancer le transfert.
2. the section called "Accepter ou rejeter un transfert de certificat" (p. 254)

   Le compte de réception doit explicitement accepter ou rejeter le certificat transféré. Une fois que le compte de réception accepte le certificat, le certificat doit être activé avant utilisation.
3. the section called "Annulation d'un transfert de certificat" (p. 254)

   Le compte d'origine peut annuler un virement, si le certificat n'a pas été accepté.

Commencer un transfert de certificat

Vous pouvez commencer à transférer un certificat vers un autre Compte AWS à l'aide du kitAWS IoTconsoleou leAWS CLI.

Commencer un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'ID du certificat à transférer.

Effectuez cette procédure à partir du compte avec le certificat à transférer.

Pour commencer à transférer un certificat à un autre Compte AWS

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

   Choisissez le certificat avec unActifouInactifL'état que vous souhaitez transférer et ouvrez la page de détails le concernant.

3. Sur le certificatDétails, dans laActions, si l'Deactivateest disponible, choisissez l'optionDeactivatepour désactiver le certificat.
4. Sur le certificatDétailsDans le menu de gauche, choisissezStratégies.
5. Sur le certificatStratégies, si des stratégies sont attachées au certificat, détachez-les en ouvrant le menu Options de la stratégie et en sélectionnantDetach.

   Aucune stratégie ne doit être attachée au certificat avant de continuer.
6. Sur le certificatStratégiesDans le menu de gauche, choisissezObjets.
7. Sur le certificatObjets, s'il y a des éléments attachés au certificat, détachez chacun en ouvrant le menu d'options de la chose et en choisissantDetach.

   Aucun objet ne doit être attaché au certificat avant de continuer.
8. Sur le certificatObjetsDans le menu de gauche, choisissezDétails.
9. Sur le certificatDétails, dans laActions, choisissezDémarrer le transfertpour ouvrir l'Démarrer le transfertBoîte de dialogue.
10. DansDémarrer le transfert, entrez la stratégie Compte AWS du compte pour recevoir le certificat et un court message facultatif.
11. ChoisissezDémarrer le transfertpour transférer le certificat.

La console doit afficher un message indiquant la réussite ou l'échec du transfert. Si le transfert a été démarré, le statut du certificat est mis à jour surTransmis.

Commencer un transfert de certificat (CLI)

Pour suivre cette procédure, vous aurez besoin de l'outil*certificateId*et l'*certificateArn*du certificat que vous souhaitez transférer.

Effectuez cette procédure à partir du compte avec le certificat à transférer.

Pour commencer à transférer un certificat à un autreAWScompte

1. Utilisation de l'update-certificatepour désactiver le certificat.

   ```
   aws iot update-certificate --certificate-id certificateId --new-status INACTIVE
   ```
2. Détachez toutes les politiques.

   1. Utilisation de l'list-attached-policiesPour répertorier les stratégies attachées au certificat.

      ```
      aws iot list-attached-policies --target certificateArn
      ```
   2. Pour chaque stratégie attachée, utilisez la méthodedetach-policypour détacher la stratégie.

      ```
      aws iot detach-policy --target certificateArn --policy-name policy-name
      ```
3. Détachez toutes les choses.

   1. Utilisation de l'list-principal-thingspour répertorier les éléments attachés au certificat.

      ```
      aws iot list-principal-things --principal certificateArn
      ```
   2. Pour chaque chose attachée, utilisez la méthodedetach-thing-principalpour détacher la chose.

      ```
      aws iot detach-thing-principal --principal certificateArn --thing-name thing-name
      ```
4. Utilisation de l'transfer-certificatepour démarrer le transfert de certificat.

```
aws iot transfer-certificate --certificate-id certificateId --target-aws-
account account-id
```

Accepter ou rejeter un transfert de certificat

Vous pouvez accepter ou refuser un certificat qui vous a été transféré Compte AWS d'un autre Compte AWS à l'aide du kitAWS IoTconsoleou leAWS CLI.

Accepter ou rejeter un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'ID du certificat qui a été transféré à votre compte.

Effectuez cette procédure à partir du compte recevant le certificat qui a été transféré.

Pour accepter ou refuser un certificat qui a été transféré à votre Compte AWS

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

   Choisissez le certificat dont le statut estTransfert en attenteVous souhaitez accepter ou refuser et ouvrir la page de détails le concernant.
3. Sur le certificatDétails, dans laActionsMenu,

   - Pour accepter le certificat, choisissezAccept le transfert.
   - Pour ne pas accepter le certificat, choisissezTransfert de rejet.

Accepter ou rejeter un transfert de certificat (CLI)

Pour suivre cette procédure, vous aurez besoin de l'outil*certificateId*Le transfert de certificat que vous souhaitez accepter ou refuser.

Effectuez cette procédure à partir du compte recevant le certificat qui a été transféré.

Pour accepter ou refuser un certificat qui a été transféré à votre Compte AWS

1. Utilisation de l'accept-certificate-transferpour accepter le certificat.

   ```
   aws iot accept-certificate-transfer --certificate-id certificateId
   ```
2. Utilisation de l'reject-certificate-transferpour rejeter le certificat.

   ```
   aws iot reject-certificate-transfer --certificate-id certificateId
   ```

Annulation d'un transfert de certificat

Vous pouvez annuler un transfert de certificat avant qu'il n'ait été accepté à l'aide de l'outilAWS IoTconsoleou leAWS CLI.

Annuler un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'ID du transfert de certificat à annuler.

Effectuez cette procédure à partir du compte qui a initié le transfert de certificat.

Pour annuler un transfert de certificat

1. Connectez-vous à la consoleAWSManagement Console et ouvrez la fenêtreAWS IoTconsole.
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

   Choisissez le certificat avecTransmisdont vous souhaitez annuler le transfert et ouvrir son menu d'options.
3. Dans le menu Options du certificat, choisissez l'optionAnnuler le transfertpour annuler le transfert de certificat.

   Important

   Veillez à ne pas confondre laAnnuler le transfertavec la stratégieRévoquéoption.
   La .Annuler le transfertannule le transfert de certificat, tandis que l'optionRévoquérend le certificat irréversible inutilisable parAWS IoT.

Annuler un transfert de certificat (CLI)

Pour suivre cette procédure, vous aurez besoin de l'outil*certificateId*du transfert de certificat que vous souhaitez annuler.

Effectuez cette procédure à partir du compte qui a initié le transfert de certificat.

Utilisation de l'cancel-certificate-transferpour annuler le transfert de certificat.

```
aws iot cancel-certificate-transfer --certificate-id certificateId
```

Utilisateurs, groupes et rôles IAM

Les utilisateurs, groupes et rôles IAM constituent les mécanismes standard pour la gestion des identités et de l'authentification dansAWS. Vous pouvez les utiliser pour vous connecter àAWS IoTInterfaces HTTP utilisant leAWSkit SDK etAWS CLI.

Les rôles IAM permettent égalementAWS IoTPour accéder à d'autresAWSdans votre compte en votre nom. Par exemple, si vous souhaitez qu'un appareil publie son état dans une table DynamoDB, les rôles IAM permettentAWS IoTPour interagir avec Amazon DynamoDB. Pour de plus amples informations, veuillez consulter Rôles IAM.

Pour les connexions du courtier de messages via HTTP,AWS IoTauthentifie les utilisateurs, groupes et rôles IAM à l'aide du processus Signature Version 4. Pour plus d'informations, consultezSignatureAWSDemandes d'API.

Lorsque vous utilisezAWSSignature Version 4 avecAWS IoTLes clients doivent prendre en charge les éléments suivants dans leur implémentation TLS :

- TLS 1.2, TLS 1.1, TLS 1.0
- Validation de la signature de certificat RSA SHA-256
- Une des suites de chiffrement de la section Prise en charge des suites de chiffrement TLS

Pour plus d'informations, consultez Gestion des identités et des accès pour AWS IoT (p. 315).

Identity Amazon Cognito

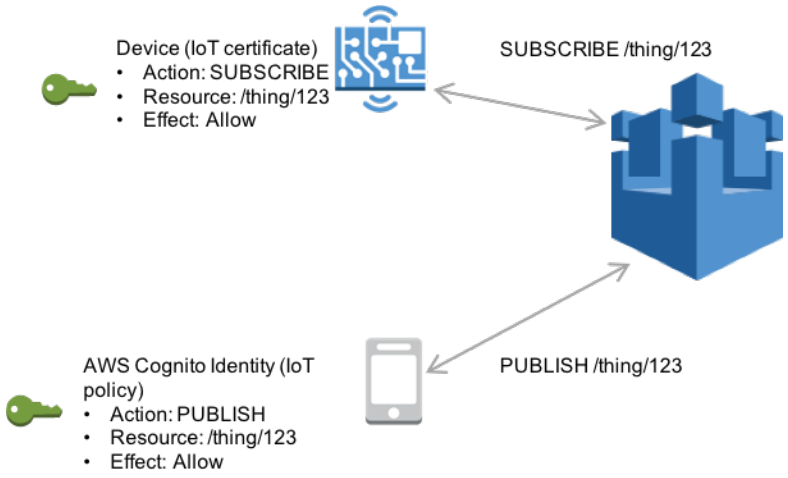
Amazon Cognito Identity vous permet de créer des privilèges limités temporairesAWSLes informations d'identification à utiliser dans des applications mobiles et Web. Lorsque vous utilisez Amazon Cognito Identity, vous créez des groupes d'identités qui créent des identités uniques pour vos utilisateurs et les

authentifient auprès de fournisseurs d'identités tels que Login with Amazon, Facebook et Google. Vous pouvez également utiliser les identités Amazon Cognito avec vos propres identités authentifiées par le développeur. Pour de plus amples informations, veuillez consulterAmazon Cognito Identity.

Pour utiliser Amazon Cognito Identity, vous définissez un groupe d'identités Amazon Cognito associé à un rôle IAM. Le rôle IAM est associé à une stratégie IAM qui accorde aux identités de votre groupe d'identités l'autorisation d'accéder à l'adresse IAM.AWSressources comme l'appelAWSServices .

Amazon Cognito Identity crée des identités non authentifiées et authentifiées. Les identités non authentifiées sont utilisées pour les utilisateurs invités d'une application mobile ou Web qui souhaitent utiliser l'application sans se connecter. Les utilisateurs non authentifiés ne bénéficient que des autorisations spécifiées dans la stratégie IAM associée au groupe d'identités.

Lorsque vous utilisez des identités authentifiées, en plus de la stratégie IAM attachée au groupe d'identités, vous pouvez attacher un objetAWS IoTà une identité Amazon Cognito à l'aide de l'outilAttachPolicyet donnez des autorisations précises d'accès à un utilisateur de votreAWS IoTapplication. De cette manière, vous pouvez attribuer des autorisations pour des clients spécifiques et leurs appareils. Pour de plus amples informations sur la création de stratégies pour les identités Amazon Cognito, veuillez consulterExemples de stratégie de publication/abonnement (p. 287).



Authentification personnalisée

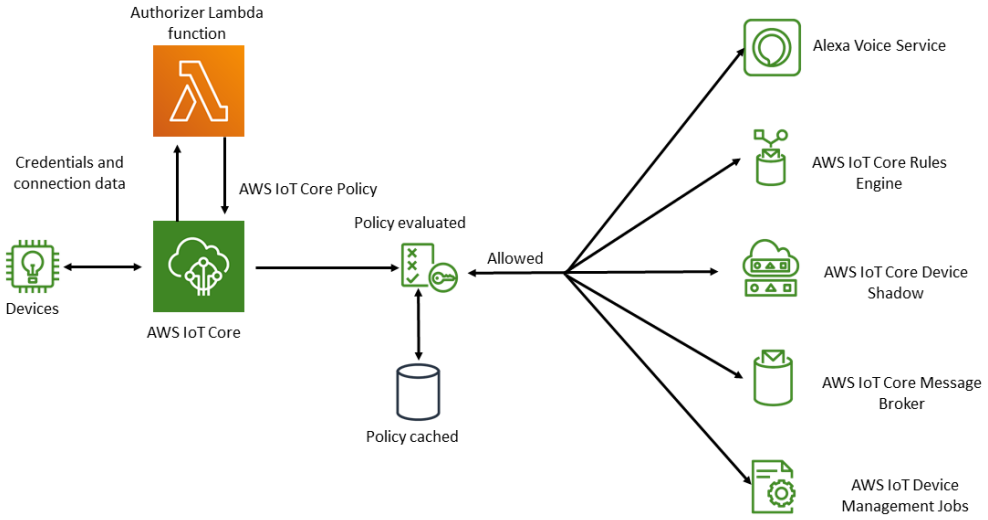
AWS IoT Core vous permet de définir des mécanismes d'autorisation personnalisée afin que vous puissiez gérer votre propre authentification et autorisation de client. Ceci est utile lorsque vous devez utiliser des mécanismes d'authentification autres que ceux qui AWS IoT Core prend en charge de manière native. (Pour plus d'informations sur les mécanismes pris en charge nativement, consultez.the section called “Authentification client” (p. 236)).

Par exemple, si vous migrez des périphériques existants dans le champ vers AWS IoT Core et que ces appareils utilisent un jeton de porteur personnalisé ou un nom d'utilisateur et un mot de passe MQTT pour s'authentifier, vous pouvez les migrer vers AWS IoT Core sans avoir à leur fournir de nouvelles identités. Vous pouvez utiliser l'authentification personnalisée avec n'importe quel protocole de communication AWS IoT Core prend en charge. Pour de plus amples informations sur les protocoles que AWS IoT Core prend en charge, voirthe section called “Protocoles de communication de périphérique” (p. 79).

Rubriques

Présentation du flux de travail d'authentification personnalisé

L'authentification personnalisée vous permet de définir comment authentifier et autoriser les clients à l'aide deressources de l'autorisation. Chaque autorisation contient une référence à une fonction Lambda gérée par le client, une clé publique facultative pour valider les informations d'identification du périphérique et des informations de configuration supplémentaires. Le schéma suivant illustre le flux de travail d'autorisation de l'authentification personnalisée dans AWS IoT Core .



AWS IoT Core Authentification et autorisation personnalisées

La liste suivante explique chaque étape du workflow d'authentification et d'autorisation personnalisées.

1. Un appareil se connecte au AWS IoT Core Point de terminaison de données en utilisant l'une des prises en chargethe section called “Protocoles de communication de périphérique” (p. 79). Le périphérique transmet des informations d'identification dans les champs d'en-tête de la requête ou les paramètres de requête (pour les protocoles HTTP Publish ou MQTT sur WebSockets) ou dans le champ nom d'utilisateur et mot de passe du message MQTT CONNECT (pour les protocoles MQTT et MQTT sur WebSockets).
2. AWS IoT Core vérifie l'une des deux conditions suivantes :
   - La demande entrante spécifie un autorisation.
   - La . AWS IoT Core le point de terminaison de données recevant la demande dispose d'un autorisation par défaut configuré pour lui.

   Si AWS IoT Core trouve un mécanisme d'autorisation de l'une ou l'autre des façons suivantes, AWS IoT Core déclenche la fonction Lambda associée à l'autorisation.
3. (Facultatif) Si vous avez activé la signature de jetons, AWS IoT Core valide la signature de la requête à l'aide de la clé publique stockée dans l'autorisateur avant de déclencher la fonction Lambda. Si la validation échoue, AWS IoT Core Arrête la demande sans invoquer la fonction Lambda.
4. La fonction Lambda reçoit les informations d'identification et les métadonnées de connexion dans la demande et prend une décision d'authentification.
5. La fonction Lambda renvoie les résultats de la décision d'authentification et un AWS IoT Core qui spécifie les actions autorisées dans la connexion. La fonction Lambda renvoie également des

informations qui spécifient à quelle fréquence AWS IoT Core Revalide les informations d'identification dans la requête en appelant la fonction Lambda.

6. AWS IoT Core évalue l'activité sur la connexion par rapport à la politique qu'elle a reçue de la fonction Lambda.

Considérations relatives à la mise à l'échelle

Étant donné qu'une fonction Lambda gère l'authentification et l'autorisation pour votre autorisateur, la fonction est soumise aux limites de tarification et de service Lambda, telles que le taux d'exécution simultanée. Pour de plus amples informations sur la tarification Lambda, veuillez consulterTarification Lambda. Vous pouvez gérer la charge de votre fonction Lambda en ajustant la`refreshAfterInSeconds`and`disconnectAfterInSeconds`dans votre réponse de fonction Lambda. Pour plus d'informations sur le contenu de votre réponse de fonction Lambda, consultez.the section called “Définition de votre fonction Lambda” (p. 258).

> **Note**
>
> Si vous laissez la signature activée, vous pouvez empêcher le déclenchement excessif de votre Lambda par des clients non reconnus. Considérez cela avant de désactiver la signature dans votre autorisateur.

Création et gestion d'autorisations personnalisées

AWS IoT Core implémente des schémas d'authentification et d'autorisation personnalisés en utilisantressources de l'autorisation. Chaque autorisation comprend les composants suivants :

- Nom : Une chaîne définie par l'utilisateur unique qui identifie l'autorisation.
- Lambda, fonction ARN : L'Amazon Resource Name (ARN) de la fonction Lambda qui implémente la logique d'autorisation et d'authentification.
- Nom de la clé de jeton : Nom de clé utilisé pour extraire le jeton des en-têtes HTTP, des paramètres de requête ou du nom d'utilisateur MQTT CONNECT afin d'effectuer la validation de signature. Cette valeur est obligatoire si la signature est activée dans votre autorisation.
- Signature de l'indicateur désactivé (facultatif) : Valeur booléenne qui spécifie s'il convient de désactiver l'exigence de signature sur les informations d'identification. Ceci est utile pour les scénarios où la signature des informations d'identification n'a pas de sens, tels que les schémas d'authentification utilisant le nom d'utilisateur et le mot de passe MQTT. La valeur par défaut est `.false`, la signature est donc activée par défaut.
- Clé publique : Clé publique AWS IoT Core utilise pour valider la signature du jeton. Sa longueur minimale est 2 048 bits. Cette valeur est obligatoire si la signature est activée dans votre autorisation.

Lambda vous facture pour le nombre de fois que votre fonction Lambda s'exécute et pour le temps nécessaire à l'exécution du code de votre fonction. Pour de plus amples informations sur la tarification Lambda, veuillez consulterTarification Lambda. Pour plus d'informations sur la création de fonctions Lambda, consultez leLambda.

> **Note**
>
> Si vous laissez la signature activée, vous pouvez empêcher le déclenchement excessif de votre Lambda par des clients non reconnus. Considérez cela avant de désactiver la signature dans votre autorisateur.

Définition de votre fonction Lambda

Quand AWS IoT Core appelle votre autorisateur, il déclenche le Lambda associé associé à l'autorisateur avec un événement qui contient l'objet JSON suivant. L'exemple d'objet JSON contient tous les champs possibles. Tous les champs qui ne sont pas pertinents pour la demande de connexion ne sont pas inclus.

```
{
    "token" :"aToken",
    "signatureVerified": Boolean, // Indicates whether the device gateway has validated the
 signature.
    "protocols": ["tls", "http", "mqtt"], // Indicates which protocols to expect for the
 request.
    "protocolData": {
        "tls" : {
            "serverName": "serverName" // The server name indication (SNI) host_name
 string.
        },
        "http": {
            "headers": {
                "#{name}": "#{value}"
            },
            "queryString": "?#{name}=#{value}"
        },
        "mqtt": {
            "username": "myUserName",
            "password": "myPassword", // A base64-encoded string.
            "clientId": "myClientId" // Included in the event only when the device sends
 the value.
        }
    },
    "connectionMetadata": {
        "id": UUID // The connection ID. You can use this for logging.
    },
}
```

La fonction Lambda doit utiliser ces informations pour authentifier la connexion entrante et décider quelles actions sont autorisées dans la connexion. La fonction doit envoyer une réponse qui contient les valeurs suivantes.

- `isAuthenticated` : valeur booléenne qui indique si la demande est authentifiée.
- `principalId` : chaîne alphanumérique qui agit comme un identifiant pour le jeton envoyé par la demande d'autorisation personnalisée. Sa longueur minimale est 1 caractère. Sa longueur maximale est de 128 caractères.
- `policyDocuments` : Une liste de formats JSON AWS IoT Core Documents de stratégie Pour de plus amples informations sur la création AWS IoT Core Stratégies, consultezthe section called “Stratégies AWS IoT Core ” (p. 270). Le nombre maximal de documents de politique est de 10 documents de politique. Chaque document de stratégie peut contenir 2 048 caractères au maximum.
- `disconnectAfterInSeconds` : nombre entier qui spécifie la durée maximale (en secondes) de la connexion à l'identité AWS IoT Core Passerelle. La valeur minimale est de 300 secondes et la valeur maximale de 86 400 secondes.
- `refreshAfterInSeconds` : Entier qui spécifie l'intervalle entre les actualisations de stratégie. Lorsque cet intervalle passe, AWS IoT Core appelle la fonction Lambda pour permettre des actualisations de stratégie. La valeur minimale est de 300 secondes et la valeur maximale de 86 400 secondes.

L'objet JSON suivant contient un exemple de réponse que votre fonction Lambda peut envoyer.

```
{
"isAuthenticated":true, //A Boolean that determines whether client can connect.
"principalId": "xxxxxxxx",  //A string that identifies the connection in logs.
"disconnectAfterInSeconds": 86400,
"refreshAfterInSeconds": 300,
 "policyDocuments": [
      {
```

```
        "Version": "2012-10-17",
        "Statement": [
           {
              "Action": "iot:Publish",
              "Effect": "Allow",
              "Resource": "arn:aws:iot:us-east-1:<your_aws_account_id>:topic/
customauthtesting"
            }
         ]
       }
    ]
}
```

La .`policyDocument`doit contenir une valeur valide AWS IoT Core Document de stratégie. Pour plus d'informations sur AWS IoT Core Stratégies, consultezthe section called “Stratégies AWS IoT Core ” (p. 270). Dans les connexions MQTT sur TLS et MQTT sur WebSockets, AWS IoT Core met en cache cette stratégie pour l'intervalle spécifié dans la valeur de la propriété`refreshAfterInSeconds`field. Au cours de cet intervalle, AWS IoT Core autorise des actions dans une connexion établie contre cette stratégie mise en cache sans déclencher à nouveau votre fonction Lambda. Si des échecs se produisent lors de l'authentification personnalisée, AWS IoT Core met fin à la connexion. AWS IoT Core met également fin à la connexion si elle a été ouverte depuis plus longtemps que la valeur spécifiée dans la`disconnectAfterInSeconds` Paramètre .

Le JavaScript suivant contient un exemple de fonction Node.js Lambda qui recherche un mot de passe dans le message MQTT Connect avec une valeur de`test`et renvoie une stratégie accordant à l'autorisation de se connecter à AWS IoT Core avec un client nommé`myClientName`et publiez dans une rubrique qui contient le même nom de client. S'il ne trouve pas le mot de passe attendu, il renvoie une stratégie qui refuse ces deux actions.

```
// A simple Lambda function for an authorizer. It demonstrates
// how to parse an MQTT password and generate a response.

exports.handler = function(event, context, callback) {
    var uname = event.protocolData.mqtt.username;
    var pwd = event.protocolData.mqtt.password;
    var buff = new Buffer(pwd, 'base64');
    var passwd = buff.toString('ascii');
    switch (passwd) {
        case 'test':
            callback(null, generateAuthResponse(passwd, 'Allow'));
        default:
            callback(null, generateAuthResponse(passwd, 'Deny'));
    }
};

// Helper function to generate the authorization response.
var generateAuthResponse = function(token, effect) {
    var authResponse = {};
    authResponse.isAuthenticated = true;
    authResponse.principalId = 'TEST123';

    var policyDocument = {};
    policyDocument.Version = '2012-10-17';
    policyDocument.Statement = [];
    var publishStatement = {};
    var connectStatement = {};
    connectStatement.Action = ["iot:Connect"];
    connectStatement.Effect = effect;
    connectStatement.Resource = ["arn:aws:iot:us-east-1:123456789012:client/myClientName"];
    publishStatement.Action = ["iot:Publish"];
    publishStatement.Effect = effect;
```

```
    publishStatement.Resource = ["arn:aws:iot:us-east-1:123456789012:topic/telemetry/
myClientName"];
    policyDocument.Statement[0] = connectStatement;
    policyDocument.Statement[1] = publishStatement;
    authResponse.policyDocuments = [policyDocument];
    authResponse.disconnectAfterInSeconds = 3600;
    authResponse.refreshAfterInSeconds = 300;

    return authResponse;
}
```

Cette fonction Lambda renvoie les valeurs suivantes lorsqu'elle reçoit la valeur attendue de`testdans` le champ MQTT Connect mot de passe.

```
{
  "password": "password",
  "isAuthenticated": true,
  "principalId": "principalId",
  "policyDocuments": [
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "iot:Connect",
          "Effect": "Allow",
          "Resource": "*"
        },
        {
          "Action": "iot:Publish",
          "Effect": "Allow",
          "Resource": "arn:aws:region:accountId:topic/telemetry/${iot:ClientId}"
        },
        {
          "Action": "iot:Subscribe",
          "Effect": "Allow",
          "Resource": "arn:aws:iot:region:accountId:topicfilter/telemetry/${iot:ClientId}"
        },
        {
          "Action": "iot:Receive",
          "Effect": "Allow",
          "Resource": "arn:aws:iot:region:accountId:topic/telemetry/${iot:ClientId}"
        }
      ]
    }
  ],
  "disconnectAfterInSeconds": 3600,
  "refreshAfterInSeconds": 300
}
```

Crée un mécanisme d'autorisation

Vous pouvez créer un mécanisme d'autorisation à l'aide du mécanisme d'autorisationAPI CreateAuthorizer. L'exemple suivant illustre la marche à suivre.

```
 aws iot create-authorizer
--authorizer-name MyAuthorizer
--authorizer-function-arn arn:aws:lambda:us-
west-2:<account_id>:function:MyAuthorizerFunction  //The ARN of the Lambda function.
[--token-key-name MyAuthorizerToken //The key used to extract the token from headers.
[--token-signing-public-keys FirstKey=
```

```
 "-----BEGIN PUBLIC KEY-----
  [...insert your public key here...]
  -----END PUBLIC KEY-----"
[--status ACTIVE]
[--tags <value>]
[--signing-disabled | --no-signing-disabled]
```

Vous pouvez utiliser la stratégie`signing-disabled`pour désactiver la validation de signature pour chaque appel de votre autorisateur. Nous vous recommandons fortement de ne pas désactiver la signature sauf si vous le devez. La validation de signature vous protège contre les invocations excessives de votre fonction Lambda à partir d'appareils inconnus. Vous ne pouvez pas mettre à jour les`signing-disabled`d'un autorisateur après l'avoir créé. Pour modifier ce comportement, vous devez créer un autre mécanisme d'autorisation personnalisée avec une valeur différente pour l'identité`signing-disabled` Paramètre .

Valeurs pour la stratégie`tokenKeyName`and`tokenSigningPublicKeys`sont facultatifs si vous avez désactivé la signature. Ce sont des valeurs obligatoires si la signature est activée.

Une fois que vous avez créé votre fonction Lambda et l'autorisation personnalisée, vous devez explicitement accorder la AWS IoT Core Pour invoquer la fonction en votre nom. Vous pouvez le faire à l'aide de la commande suivante.

```
aws lambda add-permission --function-name <lambda_function_name> --principal
iot.amazonaws.com --source-arn <authorizer_arn> --statement-id Id-123 --action
"lambda:InvokeFunction"
```

Test de vos autorisations

Vous pouvez utiliser la stratégieTestInvokeAuthorizerpour tester les valeurs d'invocation et de retour de votre autorisateur. Cette API vous permet de spécifier des métadonnées de protocole et de tester la validation de la signature dans votre autorisation.

Les onglets suivants montrent comment utiliser laAWS CLIpour tester votre autorisation.

Unix-like

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER \
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

Windows CMD

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER ^
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

Windows PowerShell

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER `
--token TOKEN_VALUE --token-signature TOKEN_SIGNATURE
```

La valeur de l'`token-signature`est le jeton signé. Pour savoir comment obtenir cette valeur, consultezthe section called “Signature du jeton” (p. 265).

Si votre organisme d'autorisation prend un nom d'utilisateur et un mot de passe, vous pouvez transmettre ces informations à l'aide de l'outil`--mqtt-context` Paramètre . Les onglets suivants montrent comment

utiliser la`TestInvokeAuthorizer`pour envoyer un objet JSON contenant un nom d'utilisateur, un mot de passe et un nom de client à votre autorisation personnalisée.

Unix-like

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  \
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

Windows CMD

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  ^
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

Windows PowerShell

```
aws iot test-invoke-authorizer --authorizer-name NAME_OF_AUTHORIZER  `
--mqtt-context '{"username": "USER_NAME", "password": "dGVzdA==",
 "clientId":"CLIENT_NAME"}'
```

Le mot de passe doit être codé en base64. L'exemple suivant montre comment encoder un mot de passe dans un environnement de type UNIX.

```
echo -n PASSWORD | base64
```

Gestion des autorisations personnalisées

Vous pouvez gérer vos autorisations à l'aide des API suivantes.

- ListAuthorizers : Afficher tous les agents d'autorisation de votre compte.
- DescribeAuthorizer : Affiche les propriétés de l'mécanisme d'autorisation spécifié. Ces valeurs incluent la date de création, la date de dernière modification et d'autres attributs.
- SetDefaultAuthorizer : Spécifie l'autorisation par défaut de votre AWS IoT Core points de terminaison de données. AWS IoT Core utilise cet autorisation si un périphérique ne passe pas AWS IoT Core et ne spécifie pas d'autorisation. Pour plus d'informations sur l'utilisation AWS IoT Core informations d'identification, voirthe section called “Authentification client” (p. 236).
- UpdateAuthorizer : Modifie l'état, le nom de la clé de jeton ou les clés publiques de l'autorisation spécifié.
- DeleteAuthorizer : Supprime le mécanisme d'autorisation spécifié.

> **Note**
>
> Vous ne pouvez pas mettre à jour l'exigence de signature d'un autorisateur. Cela signifie que vous ne pouvez pas désactiver la signature dans un autorisateur existant qui en a besoin. Vous ne pouvez pas non plus exiger la connexion d'un autorisateur existant qui n'en a pas besoin.

Connexion à AWS IoT Core en utilisant l'authentification personnalisée

Les périphériques peuvent se connecter à AWS IoT Core en utilisant l'authentification personnalisée avec n'importe quel protocole AWS IoT Core prend en charge la messagerie des appareils. Pour plus d'informations sur les protocoles de communication pris en charge, consultez.the section called “Protocoles de communication de périphérique” (p. 79). Les données de connexion que vous transmettez à votre fonction Lambda d'autorisation dépendent du protocole que vous utilisez. Pour plus d'informations sur la création de votre fonction Lambda d'autorisation, consultez.the section called “Définition de votre fonction Lambda” (p. 258). Les sections suivantes expliquent comment vous connecter à l'authentification à l'aide de chaque protocole pris en charge.

HTTP

Appareils envoient des données à AWS IoT Core à l'aide du kitAPI de publication HTTPpeuvent transmettre des informations d'identification via des en-têtes de requête ou des paramètres de requête dans leurs requêtes HTTP POST. Les périphériques peuvent spécifier un autorisation à appeler à l'aide de la commande`x-amz-customauthorizer-name`ou paramètre de requête. Si vous avez activé la signature de jeton dans votre autorisateur, vous devez passer la`token-key-name`and`x-amz-customauthorizer-signature`dans les en-têtes de requête ou les paramètres de requête. Les exemples de requêtes suivants montrent comment vous transmettez ces paramètres dans les en-têtes de requête et les paramètres de requête.

```
//Passing credentials via headers
POST /topics/topic?qos=qos HTTP/1.1
Host: your-endpoint
x-amz-customauthorizer-signature: token-signature
token-key-name: some-token
x-amz-customauthorizer-name: <authorizer-name>

//Passing credentials via query parameters
POST /topics/topic?qos=qos&x-amz-customauthorizer-signature=${sign}&token-name=${token-value} HTTP/1.1
```

MQTT

Appareils se connectent à AWS IoT Core à l'aide d'une connexion MQTT peut transmettre des informations d'identification via le`username`and`password`des messages MQTT. La .`username`La valeur peut également contenir une chaîne de requête qui transmet des valeurs supplémentaires (y compris un jeton, une signature et un nom d'autorisation) à votre mécanisme d'autorisation. Vous pouvez utiliser cette chaîne de requête si vous souhaitez utiliser un schéma d'authentification basé sur des jetons au lieu de`username`and`password`Valeurs.

> **Note**
>
> Les données figurant dans le champ mot de passe sont codées en base64 par AWS IoT Core . Votre fonction Lambda doit le décoder.

L'exemple suivant contient une`username`qui contient des paramètres supplémentaires qui spécifient un jeton et une signature.

```
username?x-amz-customauthorizer-name=${name}&x-amz-customauthorizer-signature=${sign}&token-name=${token-value}
```

Afin d'appeler un autorisateur, les périphériques se connectant à AWS IoT Core à l'aide de MQTT et l'authentification personnalisée doit se connecter sur le port 443. Ils doivent également transmettre

l'extension TLS ALPN (Application Layer Protocol Negotiation) avec la valeur de`mqtt`et l'extension SNI () avec le nom d'hôte de leur AWS IoT Core point de terminaison de données. Pour plus d'informations sur ces valeurs, consultezthe section called “Protocoles de communication de périphérique” (p. 79).
Le V2AWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil (p. 1140)peut configurer ces deux extensions.

MQTT via WebSockets

Appareils se connectent à AWS IoT Core en utilisant MQTT sur WebSockets peut transmettre des informations d'identification de l'une des deux manières suivantes.

- Par le biais des en-têtes de requête ou des paramètres de requête dans la requête HTTP UPPLATE pour établir la connexion WebSockets.
- Par l'intermédiaire du`username`and`password`dans le message MQTT CONNECT.

Si vous transmettez des informations d'identification via le message de connexion MQTT, les extensions ALPN et SNI TLS sont requises. Pour plus d'informations sur ces extensions, consultezthe section called “MQTT” (p. 264). L'exemple suivant montre comment transmettre les informations d'identification via la demande de mise à jour HTTP.

```
GET /mqtt HTTP/1.1
 Host: your-endpoint
Upgrade: WebSocket
Connection: Upgrades
x-amz-customauthorizer-signature: token-signature
token-key-name: some-token
sec-WebSocket-Key: any random base64 value
sec-websocket-protocol: mqtt
sec-WebSocket-Version: websocket version
```

Signature du jeton

Vous devez signer le jeton avec la clé privée de la key pair publique/privée que vous avez utilisée dans l'outil`create-authorizer`Appelez. Les exemples suivants montrent comment créer la signature de jeton à l'aide d'une commande de type UNIX et de JavaScript. Ils utilisent l'algorithme de hachage SHA-256 pour encoder la signature.

Command line

```
echo -n TOKEN_VALUE | openssl dgst -sha256 -sign PEM encoded RSA private key | openssl
 base64
```

JavaScript

```
const crypto = require('crypto')

const key = "PEM encoded RSA private key"

const k = crypto.createPrivateKey(key)
let sign = crypto.createSign('SHA256')
sign.write(t)
sign.end()
const s = sign.sign(k, 'base64')
```

Résolution des problèmes de vos autorisations

Cette rubrique décrit les problèmes courants qui peuvent causer des problèmes dans les workflows d'authentification personnalisés et les étapes à suivre pour les résoudre. Pour résoudre les problèmes de manière plus efficace, activez les journaux CloudWatch pour AWS IoT Core et définissez le niveau de journalisation surDEBUG. Vous pouvez activer les journaux CloudWatch dans la AWS IoT Core Console (https://console.aws.amazon.com/iot/). Pour plus d'informations sur l'activation et la configuration des journaux pour AWS IoT Core , voirthe section called “Configurer la journalisation AWS IoT” (p. 353).

> Note
>
> Si vous quittez le niveau du journal àDEBUGpendant de longues périodes, CloudWatch peut stocker de grandes quantités de données de journalisation. Cela peut augmenter vos frais CloudWatch. Envisagez d'utiliser la journalisation basée sur les ressources pour augmenter la verbosité pour uniquement les périphériques d'un groupe de choses particulier. Pour plus d'informations sur la journalisation basée sur les ressources, consultez.the section called “Configurer la journalisation AWS IoT” (p. 353). En outre, lorsque vous avez terminé le dépannage, réduisez le niveau du journal à un niveau moins détaillé.

Avant de commencer le dépannage, consultezthe section called “Présentation du flux de travail d'authentification personnalisé” (p. 257)Pour une vue globale du processus d'authentification personnalisé. Cela vous aide à comprendre où rechercher la source d'un problème.

Cette rubrique traite des deux domaines suivants que vous devez examiner.

- Problèmes liés à la fonction Lambda de votre mécanisme d'autorisation.
- Problèmes liés à votre appareil.

Vérifiez les problèmes dans la fonction Lambda de votre autorisateur

Effectuez les étapes suivantes pour vous assurer que les tentatives de connexion de vos appareils appellent votre fonction Lambda.

1. Vérifiez quelle fonction Lambda est associée à votre autorisation.

   Pour ce faire, appelez l'DescribeAuthorizerou en cliquant sur l'autorisation souhaitée dans la sectionSecureSection du AWS IoT Core console
2. Vérifiez les métriques d'appel pour la fonction Lambda. Effectuez les étapes suivantes pour procéder.

   a. Ouverture d'AWS LambdaConsole (https://console.aws.amazon.com/lambda/) et sélectionnez la fonction qui est associée à votre autorisation.

   b. Cliquez sur l'ongletContrôleet affichez les mesures correspondant à la période pertinente à votre problème.
3. Si vous ne voyez aucune invocation, vérifiez que AWS IoT Core a l'autorisation d'appeler votre fonction Lambda. Si vous voyez des appels d'appel, passez à l'étape suivante. Effectuez les étapes suivantes pour vérifier que votre fonction Lambda dispose des autorisations requises.

   a. Cliquez sur l'ongletAutorisationspour votre fonction dans l'ongletAWS Lambdaconsole

   b. Recherchez la valeurStratégie basée sur une ressourceen bas de la page. Si votre fonction Lambda dispose des autorisations requises, la stratégie ressemble à l'exemple suivant.

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

```
      "Sid": "Id-123",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:Region:AccountID:function:FunctionName",
      "Condition": {
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:iot:Region:AccountID:authorizer/AuthorizerName"
        }
      }
    }
  ]
}
```

c. Cette stratégie accorde à la`InvokeFunction`sur votre fonction à l' AWS IoT Core principal. Si vous ne le voyez pas, vous devrez l'ajouter à l'aide de l'AddPermissionAPI. L'exemple suivant montre comment procéder à l'aide de l'AWS CLI.

```
 aws lambda add-permission --function-name FunctionName --principal
iot.amazonaws.com --source-arn AuthorizerARn --statement-id Id-123 --action
"lambda:InvokeFunction"
```

4. Si vous voyez des appels, vérifiez qu'il n'y a pas d'erreurs. Une erreur peut indiquer que la fonction Lambda ne gère pas correctement l'événement de connexion AWS IoT Core lui envoie.

   Pour plus d'informations sur la gestion de l'événement dans votre fonction Lambda, consultezthe section called “Définition de votre fonction Lambda” (p. 258). Vous pouvez utiliser la fonction de test dans laAWS LambdaConsole (https://console.aws.amazon.com/lambda/) pour coder en dur les valeurs de test dans la fonction pour s'assurer que la fonction gère correctement les événements.
5. Si vous voyez des appels sans erreur, mais que vos appareils ne sont pas en mesure de se connecter (ou de publier, de s'abonner et de recevoir des messages), le problème peut être que la stratégie renvoyée par votre fonction Lambda ne donne pas d'autorisations pour les actions que vos appareils tentent de réaliser. Effectuez les étapes suivantes pour déterminer si quelque chose ne va pas avec la stratégie renvoyée par la fonction.

   a. Utilisez une requête Amazon CloudWatch Logs Insights pour analyser les journaux sur une courte période afin de vérifier les échecs. L'exemple de requête suivant trie les événements par horodatage et recherche les échecs.

```
display clientId, eventType, status, @timestamp | sort @timestamp desc | filter
 status = "Failure"
```

   b. Mettez à jour votre fonction Lambda pour enregistrer les données dans lesquelles il retourne AWS IoT Core et l'événement qui déclenche la fonction. Vous pouvez utiliser ces journaux pour inspecter la stratégie créée par la fonction.

Examen des problèmes de périphérique

Si vous ne trouvez aucun problème avec l'appel de votre fonction Lambda ou avec la stratégie renvoyée par la fonction, recherchez les problèmes liés aux tentatives de connexion de vos appareils. Les demandes de connexion mal formées peuvent provoquer AWS IoT Core pour ne pas déclencher votre autorisation. Des problèmes de connexion peuvent survenir à la fois au niveau des couches TLS et de l'application.

Problèmes possibles de couche TLS :

- Les clients doivent transmettre soit un en-tête de nom d'hôte (HTTP, MQTT sur WebSockets), soit l'extension TLS d'indication de nom de serveur (HTTP, MQTT sur WebSockets, MQTT) dans toutes les demandes d'authentification personnalisées. Dans les deux cas, la valeur transmise doit correspondre à l'un des AWS IoT Core points de terminaison de données. Ce sont les points de terminaison qui sont renvoyés lorsque vous exécutez les commandes CLI suivantes.
  - `aws iot describe-endpoint –endpoint type iot:data-ats`
  - `aws iot describe-endpoint –endpoint type`(pour les terminaux VeriSign hérités)
- Les appareils qui utilisent l'authentification personnalisée pour les connexions MQTT doivent également transmettre l'extension TLS ALPN (Application Layer Protocol Negotiation) avec la valeur de`mqtt`.
- L'authentification personnalisée est actuellement disponible uniquement sur le port 443.

Problèmes possibles de couche d'application :

- Si la signature est activée (l'option`signingDisabled`est false dans votre autorisation), recherchez les problèmes de signature suivants.
  - Assurez-vous que vous transmettez la signature de jeton dans le`x-amz-customauthorizer-signature`ou dans un paramètre de chaîne de requête.
  - Assurez-vous que le service ne signe pas une valeur autre que le jeton.
  - Assurez-vous que vous transmettez le jeton dans l'en-tête ou le paramètre de requête que vous avez spécifié dans le paramètre`token-key-name`dans votre autorisateur.
- Assurez-vous que le nom de l'autorisation que vous transmettez dans la boîte de dialogue`x-amz-customauthorizer-name`'en-tête ou de chaîne de requête est valide ou que vous avez un autorisation par défaut défini pour votre compte.

Authorization

L'autorisation est le processus d'octroi d'autorisations à une identité authentifiée. Vous accordez des autorisations dans AWS IoT Core utilisant AWS IoT Core et stratégies IAM. Cette rubrique couvre les stratégies AWS IoT Core . Pour plus d'informations sur les stratégies IAM, consultezGestion des identités et des accès pour AWS IoT (p. 315)andStratégies IAM (p. 320).

Les stratégies AWS IoT Core déterminent ce qu'une identité authentifiée peut faire. Une identité authentifiée peut être utilisée par des appareils, des applications mobiles, des applications web et des applications de bureau. Une identité authentifiée peut même être un utilisateur qui tape des commandes CLI AWS IoT Core . Une identité ne peut exécuter des opérations AWS IoT Core que si elle dispose d'une stratégie lui octroyant l'autorisation nécessaire pour ces opérations.

les deux AWS IoT Core et les stratégies IAM sont utilisées avec AWS IoT Core Pour contrôler les opérations d'une identité (également appeléeprincipal) peut effectuer. Le type de stratégie utilisé dépend du type d'identité employé pour s'authentifier auprès d' AWS IoT Core .

Les opérations AWS IoT Core se divisent en deux groupes :

- L'API de plan de contrôle vous permet d'effectuer des tâches administratives telles que la création ou la mise à jour de certificats, d'objets, de règles, etc.
- L'API de plan de données vous permet d'envoyer des données à destination et en provenance d' AWS IoT Core .

Le type de stratégie que vous utilisez varie selon que vous utilisez l'API de plan de contrôle ou l'API de plan de données.

Le tableau suivant présente les différents types d'identité, les protocoles qu'ils utilisent, ainsi que les types de stratégie qui peuvent être utilisés à des fins d'autorisation.

API de plan de données et types de stratégie AWS IoT Core

Protocole et mécanisme d'authentification	Kit SDK	Type d'identité	Type de stratégie		
MQTT sur TLS/TCP, authentification mutuelle TLS (port 8883 ou 443)))[† (p. 80)])	SDK pour les appareils AWS IoT Core	Certificats X.509	Stratégie AWS IoT Core		
MQTT sur HTTPS/ WebSocket,AWSAuthentification SIGv4 (port 443)	AWSKit,SDK Mobile	Amazon Cognito Identity Identity Identity Identity	IAM et AWS IoT Core Stratégies		
		Amazon Cognito Identity Identity Identity Identity Identity	Stratégie IAM		
		IAM ou identité fédérée	Stratégie IAM		
HTTPS,AWSAuthentification Signature Version 4 (port 443)	AWS CLI	Amazon Cognito Identity Identity Identity ou identité fédérée identité	Stratégie IAM		
HTTPS, authentification mutuelle TLS (port 8443)	Pas de prise en charge SDK	Certificats X.509	Stratégie AWS IoT Core		
HTTPS sur authentification personnalisée (Port 443)	SDK pour les appareils AWS IoT Core	Mécanisme d'autorisation personnalisé	Stratégie d'autorisation personnalisée		

API de plan de contrôle et types de stratégie AWS IoT Core

Protocole et mécanisme d'authentification	Kit SDK	Type d'identité	Type de stratégie		
HTTPSAWSAuthentification Signature Version 4 (port 443)	AWS CLI	Amazon Cognito	Stratégie IAM		
		IAM ou identité fédérée	Stratégie IAM		

AWS IoT Core Les stratégies sont attachées à des certificats X.509 ou à des identités Amazon Cognito. Les stratégies IAM sont attachées à un utilisateur, groupe ou rôle IAM. Si vous utilisez la stratégieAWS IoTou la console AWS IoT Core L'interface de ligne de commande pour attacher la stratégie (à un certificat

ou à une identité Amazon Cognito), vous utilisez une interface de ligne de commande AWS IoT Core La stratégie. Sinon, vous utilisez une stratégie IAM.

L'autorisation basée sur la stratégie est un outil puissant. Elle vous permet de contrôler entièrement ce qu'un appareil, un utilisateur ou une application peut faire dans AWS IoT Core . Par exemple, imaginons un appareil se connectant à AWS IoT Core avec un certificat. Vous pouvez autoriser l'appareil à accéder à toutes les rubriques MQTT ou limiter son accès à une seule rubrique. Autre exemple : imaginons le cas d'un utilisateur qui tape des commandes CLI dans la ligne de commande. En utilisant une stratégie, vous pouvez lui autoriser ou refuser l'accès à une commande ou à une ressource AWS IoT Core quelconque. Vous pouvez également contrôler l'accès d'une application aux ressources AWS IoT Core .

Les modifications apportées à une stratégie peuvent prendre quelques minutes avant d'entrer en vigueur en raison de la façon dontAWS IoTmet en cache les documents de stratégie. Autrement dit, il peut prendre quelques minutes pour accéder à une ressource qui a récemment obtenu l'accès, et une ressource peut être accessible pendant plusieurs minutes après la révocation de son accès.

Formation et certification AWS

Pour plus d'informations sur l'autorisation dans AWS IoT Core , prenez laPlongez profond dans AWS IoT Core Authentification et autorisationsur leAWSSite Web de formation et de certification.

Stratégies AWS IoT Core

Les stratégies AWS IoT Core sont des documents JSON. Ils suivent les mêmes conventions que les stratégies IAM. AWS IoT Core prend en charge les stratégies nommées afin que plusieurs identités puissent être référencées au même document de stratégie. Les stratégies nommées comptent plusieurs versions afin de faciliter leur restauration.

Les stratégies AWS IoT Core vous permettent de contrôler l'accès au plan de données AWS IoT Core . La . AWS IoT Core se compose d'opérations qui vous permettent de vous connecter au AWS IoT Core L'agent de messages, d'envoyer et recevoir des messages MQTT et d'obtenir ou mettre à jour Device Shadow d'objet.

Une stratégie AWS IoT Core est un document JSON qui contient une ou plusieurs déclarations de stratégie. Chaque déclaration contient :

- `Effect`, qui indique si l'action est autorisée ou refusée.
- `Action`, qui indique l'action autorisée ou refusée par la stratégie.
- `Resource`, qui spécifie la ou les ressources sur lesquelles l'action est autorisée ou refusée.

Les modifications apportées à une stratégie peuvent prendre quelques minutes avant d'entrer en vigueur en raison de la façon dontAWS IoTmet en cache les documents de stratégie. Autrement dit, il peut prendre quelques minutes pour accéder à une ressource qui a récemment obtenu l'accès, et une ressource peut être accessible pendant plusieurs minutes après la révocation de son accès.

Rubriques

Actions de stratégie AWS IoT Core

Les actions de stratégie suivantes sont définies par AWS IoT Core :

Actions de stratégie MQTT

iot:Connect

Représente l'autorisation de se connecter à l'agent de messages AWS IoT Core . L'autorisation `iot:Connect` est vérifiée chaque fois qu'une demande `CONNECT` est envoyée à l'agent. L'agent de messages n'autorise pas deux clients avec le même ID client à rester connectés en même temps. Une fois que le deuxième client se connecte, l'agent ferme la connexion existante. L'autorisation `iot:Connect` peut être utilisée pour s'assurer que seuls les clients autorisés utilisant un ID client spécifique peuvent se connecter.

iot:Publish

Représente l'autorisation de publier dans une rubrique MQTT. Cette autorisation est vérifiée chaque fois qu'une demande PUBLISH est envoyée à l'agent. Elle peut être utilisée pour autoriser les clients à publier dans des modèles de rubrique spécifiques.

> Note
>
> Pour accorder l'autorisation `iot:Publish`, vous devez également accorder l'autorisation `iot:Connect`.

iot:Receive

Représente l'autorisation de recevoir un message provenant de AWS IoT Core . L'autorisation `iot:Receive` est vérifiée chaque fois qu'un message est remis à un client. Cette autorisation étant vérifiée à chaque remise, elle peut servir à révoquer les autorisations aux clients actuellement abonnés à une rubrique.

iot:Subscribe

Représente l'autorisation de s'abonner à un filtre de rubrique. Cette autorisation est vérifiée chaque fois qu'une demande SUBSCRIBE est envoyée à l'agent. Elle peut être utilisée pour permettre aux clients de s'abonner aux rubriques qui correspondent à des modèles de rubrique spécifiques.

> Note
>
> Pour accorder l'autorisation `iot:Subscribe`, vous devez également accorder l'autorisation `iot:Connect`.

Actions de stratégie shadow d'appareil

iot:DeleteThingShadow

Représente l'autorisation de supprimer un shadow d'appareil d'objet.
La .`iot:DeleteThingShadow`L'autorisation est vérifiée chaque fois qu'une demande est faite de supprimer le contenu Device Shadow d'objet.

iot:GetThingShadow

Représente l'autorisation de récupérer un shadow d'appareil d'objet.
La .`iot:GetThingShadow`L'autorisation est vérifiée chaque fois qu'une demande est faite de récupérer le contenu d'Device Shadow d'objet.

IOT : listNamedShadowsForthing

Représente l'autorisation de lister un objet nommé Ombres.
La .`iot:ListNamedShadowsForThing`L'autorisation est vérifiée chaque fois qu'une demande est faite de répertorier un objet nommé Shadows.

iot:UpdateThingShadow

Représente l'autorisation de mettre à jour un shadow d'appareil.
La .`iot:UpdateThingShadow`L'autorisation est vérifiée chaque fois qu'une demande est faite de mettre à jour le contenu Device Shadow d'objet.

> **Note**
>
> Les actions de stratégie d'exécution de tâche s'appliquent uniquement pour le point de terminaison HTTP TLS. Si vous utilisez le point de terminaison MQTT, vous devez utiliser les actions de stratégie MQTT définies dans cette rubrique.

Actions de stratégie AWS IoT Core d'exécution de tâche

iot:DescribeJobExecution

Représente l'autorisation de récupérer une exécution de tâche pour un objet donné. L'autorisation `iot:DescribeJobExecution` est vérifiée chaque fois qu'une demande est faite d'obtenir une exécution de tâche.

iot:GetPendingJobExecutions

Représente l'autorisation de récupérer la liste des tâches qui ne sont pas à un statut terminal pour un objet. L'autorisation `iot:GetPendingJobExecutions` est vérifiée chaque fois qu'une demande est faite de récupérer la liste.

iot:UpdateJobExecution

Représente l'autorisation de mettre à jour une exécution de tâche. L'autorisation `iot:UpdateJobExecution` est vérifiée chaque fois qu'une demande est faite de mettre à jour l'état d'une exécution de tâche.

iot:StartNextPendingJobExecution

Représente l'autorisation d'obtenir et de démarrer l'exécution de tâche en attente suivante pour un objet. (C'est-à-dire de mettre à jour une exécution de tâche en la faisant passer du statut QUEUED au statut IN_PROGRESS.) L'autorisation `iot:StartNextPendingJobExecution` est vérifiée chaque fois qu'une demande est faite de démarrer l'exécution de tâche en attente suivante.

Ressources d'action AWS IoT Core

Pour spécifier une ressource pour une action de stratégie AWS IoT Core , vous devez utiliser l'ARN de la ressource. Tous les ARN de ressources se présentent sous la forme suivante :

```
arn:aws:iot:region:AWS-account-ID:Resource-type/Resource-name
```

Le tableau suivant présente la ressource à spécifier pour chaque type d'action :

Action	Type de ressource	Nom de la ressource	Exemple d'ARN
`iot:Connect`	`client`	ID client	`arn:aws:iot:us-east-1:123456789012:client/myClientId`
`iot:DeleteThingShadow`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:DescribeJobExecution`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:GetPendingJobExecutions`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`

Action	Type de ressource	Nom de la ressource	Exemple d'ARN

`iot:GetThingShadow`	`thing`	Le nom de la chose et le nom de l'ombre, le cas échéant	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
			
			`arn:aws:iot:us-east-1:123456789012:thing/thingOne/shadowOne`

`iot:ListNamedShadowsForThing`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:Publish`	`topic`	une chaîne de rubrique	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`
`iot:Receive`	`topic`	une chaîne de rubrique	`arn:aws:iot:us-east-1:123456789012:topic/myTopicName`
`iot:StartNextPendingJobExecution`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
`iot:Subscribe`	`topicfilter`	Chaîne de filtre	`arn:aws:iot:us-east-1:123456789012:topicfilt myTopicFilter`
`iot:UpdateJobExecution`	`thing`	Nom de l'objet	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`

`iot:UpdateThingShadow`	`thing`	Le nom de la chose et le nom de l'ombre, le cas échéant	`arn:aws:iot:us-east-1:123456789012:thing/thingOne`
			
			`arn:aws:iot:us-east-1:123456789012:thing/thingOne/shadowOne`

Variables de stratégie AWS IoT Core

AWS IoT Core définit les variables de stratégie qui peuvent être utilisées dans les stratégies AWS IoT Core du bloc `Resource` ou `Condition`. Lorsqu'une stratégie est évaluée, ses variables sont remplacées par des valeurs réelles. Par exemple, si un appareil s'est connecté à l'agent de messages AWS IoT Core avec l'ID client « 100-234-3456 », la variable de stratégie `iot:ClientId` est remplacée dans le document de stratégie par « 100-234-3456 ». Pour plus d'informations sur les variables de stratégie, consultezVariables de stratégie IAMandConditions à valeurs multiples.

Variables de stratégie AWS IoT Core de base

AWS IoT Core définit les variables de stratégie de base suivantes :

- `iot:ClientId` : ID client utilisé pour se connecter à l'agent de messages AWS IoT Core .
- `aws:SourceIp` : adresse IP du client connecté à l'agent de messages AWS IoT Core .

La stratégie AWS IoT Core suivante présente une stratégie qui utilise des variables de stratégie :

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": [
          "arn:aws:iot:us-east-1:123451234510:client/${iot:ClientId}"
        ]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Publish"],
        "Resource": [
          "arn:aws:iot:us-east-1:123451234510:topic/my/topic/${iot:ClientId}"
        ]
      }
    ]
}
```

Dans le cadre de ces exemples, `${iot:ClientId}` est remplacé par l'ID du client connecté à l'agent de messages AWS IoT Core lors de l'évaluation de la stratégie. Lorsque vous utilisez des variables de stratégie telles que `${iot:ClientId}`, vous pouvez ouvrir par inadvertance l'accès à des rubriques imprévues. Par exemple, si vous utilisez un stratégie qui utilise `${iot:ClientId}` pour spécifier un filtre de rubrique :

```
{
    "Effect": "Allow",
    "Action": ["iot:Subscribe"],
    "Resource": [
      "arn:aws:iot:us-east-1:123456789012:topicfilter/my/${iot:ClientId}/topic"
    ]
}
```

Un client peut se connecter en utilisant + comme ID de client. Cela permettrait à l'utilisateur de s'abonner à n'importe quelle rubrique correspondant au filtre de rubrique `my/+/topic`. Pour assurer une protection contre ces failles de sécurité, utilisez l'action de stratégie `iot:Connect` pour contrôler les ID client qui peuvent se connecter. Par exemple, cette stratégie autorise uniquement les clients dont l'ID client est `clientid1` à se connecter :

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/clientid1"
        ]
      }
    ]
}
```

Variables de stratégie d'objet

Les variables de stratégie d'objet vous permettent d'écrire des stratégies AWS IoT Core qui accordent ou refusent des autorisations en fonction de propriétés d'objet telles que les noms d'objet, les types d'objet et

les valeurs d'attribut d'objet. Vous pouvez utiliser des variables de stratégie d'objet pour appliquer la même stratégie afin de contrôler de nombreux appareils AWS IoT Core . Pour plus d'informations sur la mise en service des appareils, veuillez consulter Mise en service des appareils. Le nom d'objet est obtenu à partir de l'ID client dans le message MQTT `Connect` envoyé lorsqu'un objet se connecte à AWS IoT Core .

Gardez ce qui suit à l'esprit lorsque vous utilisez des variables de stratégie d'objet dans les stratégies AWS IoT Core .

- Utilisation de l'AttachThingPrincipalAPI pour attacher des certificats ou des principaux (identités Amazon Cognito authentifiées) à un objet.
- Lorsque vous remplacez des noms d'objet par des variables de stratégie d'objet, la valeur de `clientId` dans le message de connexion MQTT ou la connexion TLS doit correspondre exactement au nom de l'objet.

Les variables de stratégie d'objet suivantes sont disponibles :

- `iot:Connection.Thing.ThingName`

  Cette variable est résolue en nom d'objet dans le registre AWS IoT Core pour lequel la stratégie est évaluée. AWS IoT Core utilise le certificat présenté par l'appareil lors de l'authentification afin de déterminer quel objet utiliser pour vérifier la connexion. Cette variable de stratégie est disponible uniquement lorsqu'un appareil se connecte via MQTT ou MQTT via le protocole WebSocket.
- `iot:Connection.Thing.ThingTypeName`

  Cette variable est résolue en type d'objet associé à l'objet pour lequel la stratégie est évaluée. Le nom d'objet correspond à l'ID client de la connexion MQTT/WebSocket. Cette variable de stratégie est disponible uniquement lors de la connexion via MQTT ou de MQTT via le protocole WebSocket.
- `iot:Connection.Thing.Attributes[`*`attributeName`*`]`

  Cette variable est résolue en valeur d'attribut spécifié associé à l'objet pour lequel la stratégie est évaluée. Un objet peut posséder jusqu'à 50 attributs. Chaque attribut est disponible en tant que variable de stratégie : `iot:Connection.Thing.Attributes[`*`attributeName`*`]` où *`attributeName`* est le nom de l'attribut. Le nom d'objet correspond à l'ID client de la connexion MQTT/WebSocket. Cette variable de stratégie est uniquement disponible lors de la connexion via MQTT ou MQTT via le protocole WebSocket.
- `iot:Connection.Thing.IsAttached`

  `iot:Connection.Thing.IsAttached: ["true"]`applique que seuls les périphériques qui sont tous les deux enregistrés dansAWS IoTet attaché au principal peuvent accéder aux autorisations dans la stratégie. Vous pouvez utiliser cette variable pour empêcher un périphérique de se connecter à AWS IoT Core S'il présente un certificat qui n'est pas attaché à un objet IoT dans le AWS IoT Core Registry.Cette variable a des valeurs`true`ou`false`indiquant que la chose de connexion est attachée au certificat ou à l'identité Amazon Cognito dans le Registre en utilisantAttachThingPrincipalAPI. Le nom de chose est pris comme ID client.

Variables de stratégie AWS IoT Core de certificat X.509

Les variables de stratégie de certificat X.509 vous permettent d'écrire des stratégies AWS IoT Core qui octroient des autorisations basées sur les attributs de certificat X.509. Les sections suivantes décrivent comment vous pouvez utiliser ces variables de stratégie de certificat.

CertificateId

Dans l'API RegisterCertificate, le `certificateId` apparaît dans le corps de la réponse. Pour obtenir des informations sur votre certificat, vous pouvez utiliser le `certificateId` dans DescribeCertificate.

Attributs de l'émetteur

Les variables de stratégie AWS IoT Core suivantes vous permettent d'accorder ou de refuser des autorisations en fonction des attributs de certificat définis par l'émetteur du certificat.

- `iot:Certificate.Issuer.DistinguishedNameQualifier`
- `iot:Certificate.Issuer.Country`
- `iot:Certificate.Issuer.Organization`
- `iot:Certificate.Issuer.OrganizationalUnit`
- `iot:Certificate.Issuer.State`
- `iot:Certificate.Issuer.CommonName`
- `iot:Certificate.Issuer.SerialNumber`
- `iot:Certificate.Issuer.Title`
- `iot:Certificate.Issuer.Surname`
- `iot:Certificate.Issuer.GivenName`
- `iot:Certificate.Issuer.Initials`
- `iot:Certificate.Issuer.Pseudonym`
- `iot:Certificate.Issuer.GenerationQualifier`

Attributs de l'objet

Les variables de stratégie AWS IoT Core suivantes vous permettent d'accorder ou de refuser des autorisations en fonction des attributs d'objet de certificat définis par l'auteur de certificat.

- `iot:Certificate.Subject.DistinguishedNameQualifier`
- `iot:Certificate.Subject.Country`
- `iot:Certificate.Subject.Organization`
- `iot:Certificate.Subject.OrganizationalUnit`
- `iot:Certificate.Subject.State`
- `iot:Certificate.Subject.CommonName`
- `iot:Certificate.Subject.SerialNumber`
- `iot:Certificate.Subject.Title`
- `iot:Certificate.Subject.Surname`
- `iot:Certificate.Subject.GivenName`
- `iot:Certificate.Subject.Initials`
- `iot:Certificate.Subject.Pseudonym`
- `iot:Certificate.Subject.GenerationQualifier`

Les attributs des certificats X.509 peuvent contenir une ou plusieurs valeurs. Par défaut, les variables de stratégie de chaque attribut à valeurs multiples renvoient la première valeur. Par exemple, l'attribut `Certificate.Subject.Country` peut contenir une liste de noms de pays, mais `iot:Certificate.Subject.Country` est remplacé par le nom du premier pays lorsqu'il est évalué dans une stratégie. Vous pouvez demander une valeur d'attribut spécifique autre que la première valeur en utilisant un index de base un. Par exemple, `iot:Certificate.Subject.Country.1` est remplacé par le deuxième nom de pays dans l'attribut `Certificate.Subject.Country`. Si vous spécifiez une valeur d'index qui n'existe pas (par exemple, si vous demandez une troisième valeur alors qu'il n'y a que deux valeurs affectées à l'attribut), aucune substitution n'est effectuée et l'autorisation échoue. Vous pouvez utiliser le suffixe `.List` dans le nom de la variable de stratégie pour spécifier l'ensemble des valeurs de l'attribut.

Registered devices (2)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante autorise les clients ayant un nom d'objet inscrit dans le registre AWS IoT Core à se connecter, mais limite le droit de publier dans une rubrique spécifique au nom d'objet aux seuls clients qui disposent de certificats dont l'attribut `Certificate.Subject.Organization` est défini sur `"Example Corp"` ou `"AnyCompany"`. Cette restriction est appliquée à l'aide d'un champ `"Condition"` qui spécifie une condition devant être remplie pour pouvoir autoriser l'action précédente. Dans ce cas, la condition est que l'attribut `Certificate.Subject.Organization` associé au certificat doit inclure une des valeurs figurant dans la liste :

```
{
    "Version":"2012-10-17",
    "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "iot:Connect"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      },
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Connection.Thing.ThingName}"
        ],
        "Condition":{
          "ForAllValues:StringEquals":{
            "iot:Certificate.Subject.Organization.List":[
              "Example Corp",
              "AnyCompany"
            ]
          }
        }
      }
    ]
}
```

Unregistered devices (2)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde aux ID client `client1`, `client2` et `client3` l'autorisation de se connecter à AWS IoT Core , mais limite le droit de publier dans une rubrique spécifique à l'ID client aux seuls clients qui disposent de certificats dont l'attribut `Certificate.Subject.Organization` est défini sur `"Example Corp"` ou `"AnyCompany"`. Cette restriction est appliquée à l'aide d'un champ `"Condition"` qui spécifie une condition devant être remplie pour pouvoir autoriser l'action précédente. Dans ce cas, la condition est que l'attribut `Certificate.Subject.Organization` associé au certificat doit inclure une des valeurs figurant dans la liste :

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

```
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:client/client1",
          "arn:aws:iot:us-east-1:123456789012:client/client2",
          "arn:aws:iot:us-east-1:123456789012:client/client3"
        ]
      },
      {
        "Effect":"Allow",
        "Action":[
          "iot:Publish"
        ],
        "Resource":[
          "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
        ],
        "Condition":{
          "ForAllValues:StringEquals":{
            "iot:Certificate.Subject.Organization.List":[
              "Example Corp",
              "AnyCompany"
            ]
          }
        }
      }
    ]
}
```

Attributs de nom alternatif d'émetteur

Les variables de stratégie AWS IoT Core suivantes vous permettent d'accorder ou de refuser des autorisations en fonction des attributs de nom alternatif d'émetteur définis par l'auteur de certificat.

- `iot:Certificate.Issuer.AlternativeName.RFC822Name`
- `iot:Certificate.Issuer.AlternativeName.DNSName`
- `iot:Certificate.Issuer.AlternativeName.DirectoryName`
- `iot:Certificate.Issuer.AlternativeName.UniformResourceIdentifier`
- `iot:Certificate.Issuer.AlternativeName.IPAddress`

Attributs de nom alternatif d'objet

Les variables de stratégie AWS IoT Core suivantes vous permettent d'accorder ou de refuser des autorisations en fonction des attributs de nom alternatif d'objet définis par l'auteur de certificat.

- `iot:Certificate.Subject.AlternativeName.RFC822Name`
- `iot:Certificate.Subject.AlternativeName.DNSName`
- `iot:Certificate.Subject.AlternativeName.DirectoryName`
- `iot:Certificate.Subject.AlternativeName.UniformResourceIdentifier`
- `iot:Certificate.Subject.AlternativeName.IPAddress`

Autres attributs

Vous pouvez utiliser `iot:Certificate.SerialNumber` pour autoriser ou refuser l'accès aux ressources AWS IoT Core en fonction du numéro de série d'un certificat. La variable de stratégie `iot:Certificate.AvailableKeys` contient le nom de toutes les variables de stratégie de certificat contenant des valeurs.

Limitations applicables aux variables de stratégie de certificat X.509

Les limitations suivantes s'appliquent aux variables de stratégie de certificat X.509 :

Caractères génériques

Si les attributs de certificat contiennent des caractères génériques, la variable de stratégie n'est pas remplacée par la valeur d'attribut de certificat, et le texte `${policy-variable}` figure dans le document de la stratégie. Cela risque de provoquer un échec d'autorisation. Les caractères génériques suivants peuvent être utilisés : `*`, `$`, `+`, `?` et `#`.

Champs de tableau

Les attributs de certificats qui contiennent des tableaux sont limités à cinq éléments. Les autres éléments sont ignorés.

Longueur de chaîne

Toutes les valeurs de chaîne sont limitées à 1 024 caractères. Si un attribut de certificat contient une chaîne comportant plus de 1 024 caractères, la variable de stratégie n'est pas remplacée par la valeur d'attribut de certificat et `${policy-variable}` figure dans le document de la stratégie. Cela risque de provoquer un échec d'autorisation.

Caractères spéciaux

Tout caractère spécial, tel que `,`, `"`, `\`, `+`, `=`, `<`, `>` et `;` doit être préfixé par une barre oblique inverse (`\`) lorsqu'il est utilisé dans une variable de stratégie. Par exemple, `Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US` devient `Amazon Web Service O= \Amazon.com Inc. L\=Seattle ST\=Washington C\=US`.

Exemple de stratégies AWS IoT

Consultez les rubriques suivantes pour en savoir plus sur les éléments communs dans une stratégie AWS IoT. Vous pouvez utiliser les exemples de stratégie pour effectuer des tâches courantes dans AWS IoT.

Rubriques

Exemples de stratégies

Pour obtenir d'autres exemples de stratégie, consultez les rubriques suivantes dans d'autres sections de ce guide :

- the section called “Exemples de stratégies basées sur l'identité” (p. 338)
- the section called “Variables de stratégie AWS IoT Core de base” (p. 273)
- the section called “Variables de stratégie AWS IoT Core de certificat X.509” (p. 275)

Éléments de stratégie AWS IoT

Les stratégies AWS IoT sont spécifiées dans un document JSON. Une stratégie AWS IoT comprend les éléments suivants :

Version

Doit avoir la valeur `"2012-10-17"`.

Effet

Doit avoir la valeur `"Allow"` ou `"Deny"`.

Action

Doit être défini sur « `iot:`*operation-name* », où *operation-name* est l'un des éléments suivants :

`"iot:Connect"` : connexion à AWS IoT

`"iot:Receive"` : réception de messages d'AWS IoT.

`"iot:Publish"` : publication MQTT.

`"iot:Subscribe"` : abonnement MQTT.

`"iot:UpdateThingShadow"` : mise à jour d'un shadow d'appareil.

`"iot:GetThingShadow"` : récupération d'un shadow d'appareil.

`"iot:DeleteThingShadow"` : suppression d'un shadow d'appareil.

Ressource

Doit avoir l'une des valeurs suivantes :

Client :`arn:aws:iot:`*region*`:`*account-id*`:client/`*client-id*

ARN de rubrique : `arn:aws:iot:`*region*`:`*account-id*`:topic/`*topic-name*

ARN de filtre de rubriques : `arn:aws:iot:`*region*`:`*account-id*`:topicfilter/`*topic-filter*

Exemples de stratégies de connexion

La stratégie suivante accorde à l'ID client `client1` l'autorisation de se connecter à AWS IoT Core :

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      }
    ]
}
```

La stratégie suivante refuse aux ID client `client1` et `client2` l'autorisation de se connecter à AWS IoT Core , mais autorise les appareils à se connecter à l'aide d'un ID client qui correspond au nom d'un objet enregistré dans le registre AWS IoT Core :

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

```
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1",
          "arn:aws:iot:us-east-1:123456789012:client/client2"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      }
    ]
}
```

Exemples de stratégie de sessions permanentes MQTT

`connectAttributes`vous permettent de spécifier les attributs que vous souhaitez utiliser dans votre message de connexion dans vos stratégies IAM, telles que`PersistentConnect`and`LastWill`. Pour de plus amples informations, veuillez consulterUtilisation de ConnectAttributs (p. 85)

La stratégie suivante autorise la connexion avec`PersistentConnect`Fonctionnalité :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante interdit`PersistentConnect`, d'autres fonctionnalités sont autorisées :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringNotEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
```

```
                    ]
                }
            }
        }
    ]
```

La politique ci-dessus peut également être exprimée en utilisant`StringEquals`, toute autre fonctionnalité, y compris la nouvelle fonctionnalité, est autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante permet à la fois de se connecter par`PersistentConnect`and`LastWill`, toute autre nouvelle fonctionnalité n'est pas autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante permet de nettoyer la connexion par des clients avec ou sans`LastWill`, aucune autre fonctionnalité ne sera autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante autorise uniquement la connexion à l'aide de fonctionnalités par défaut :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante autorise la connexion uniquement avec`PersistentConnect`, toute nouvelle fonctionnalité est autorisée tant que la connexion utilise`PersistentConnect` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
```

```
}
```

La stratégie suivante indique que la connexion doit avoir à la fois`PersistentConnect`and`LastWill`, aucune nouvelle fonctionnalité n'est autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                    ]
                }
            }
        }
```

```
    ]
}
```

La politique suivante ne doit pas avoir`PersistentConnect`mais peut avoir`LastWill`, toute autre nouvelle fonctionnalité n'est pas autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        }
    ]
}
```

La stratégie suivante autorise la connexion uniquement par les clients disposant d'un`LastWill`avec rubrique`"my/lastwill/topicName"`, n'importe quelle fonctionnalité est autorisée s'il utilise l'outil`LastWill`Sujet :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iot:LastWillTopic": "arn:aws:iot:*region*:*account-id*:topic/*my/
lastwill/topicName*"
                }
            }
        }
    ]
}
```

La stratégie suivante autorise uniquement une connexion propre à l'aide d'un`LastWillTopic`, n'importe quelle fonctionnalité est autorisée s'il utilise l'outil`LastWillTopic` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ArnEquals": {
                    "iot:LastWillTopic": "arn:aws:iot:*region*:*account-id*:topic/*my/
lastwill/topicName*"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        }
    ]
}
```

Registered devices (3)

La stratégie suivante accorde à un appareil l'autorisation de se connecter en utilisant son nom d'objet comme ID client et de s'abonner au filtre de rubrique `my/topic/filter`. Cet appareil doit être enregistré auprès d' AWS IoT Core . Lorsque l'appareil se connecte à AWS IoT Core , il doit fournir le certificat associé à l'objet IoT dans le registre AWS IoT Core :

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic/filter"
        ]
```

```
      }
    ]
}
```

Unregistered devices (3)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter en utilisant l'ID client `client1` et de s'abonner au filtre de rubrique `my/topic` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        }
    ]
}
```

Exemples de stratégie de publication/abonnement

La stratégie que vous utilisez dépend de la façon dont vous vous connectez à AWS IoT Core . Vous pouvez vous connecter à AWS IoT Core En utilisant un client MQTT, HTTP ou WebSocket. Lorsque vous connectez à un client MQTT, vous authentifiez avec un certificat X.509. Lorsque vous vous connectez via HTTP ou le protocole WebSocket, vous vous authentifiez avec Signature Version 4 et Amazon Cognito.

Stratégies pour les clients MQTT

MQTT et AWS IoT Core ont des caractères génériques différents et ils doivent être utilisés avec une attention particulière. Dans MQTT, les caractères génériques+and#sont utilisés dansFiltres de rubrique MQTTpour vous abonner à plusieurs noms de rubriques. AWS IoT Core suivent les mêmes conventions queStratégies IAMet utilisez*en tant que caractère générique, et les caractères génériques MQTT+and#sont traités comme des chaînes littérales. Par conséquent, pour spécifier des caractères génériques dans les noms de rubrique et les filtres de rubrique dans AWS IoT Core pour les clients MQTT, vous devez utiliser*.

Le tableau ci-dessous montre les différents caractères génériques utilisés dans MQTT et AWS IoT Core Stratégies pour les clients MQTT.

Caractère générique	Caractère générique MQTT	Exemple en MQTT	Est AWS IoT Core Caractère générique	Exemple dans AWS IoT Core Stratégies pour les clients MQTT
#	Oui	some/#	Non	S/O

Caractère générique	Caractère générique MQTT	Exemple en MQTT	Est AWS IoT Core Caractère générique	Exemple dans AWS IoT Core Stratégies pour les clients MQTT
+	Oui	`some/+/topic`	Non	S/O
*	Non	S/O	Oui	`topicfilter/some/*/topic`

Pour décrire plusieurs noms de rubriques dans la`resource`d'une stratégie, utilisez l'attribut`*`caractère générique. La stratégie suivante permet à un dispositif de publier dans toutes les sous-rubriques commençant par le même nom.

Registered devices (5)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide d'un ID client qui correspond au nom d'objet et de publier dans n'importe quelle rubrique ayant le nom d'objet en préfixe :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/*"
            ]
        }
    ]
}
```

Unregistered devices (5)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide de l'ID client `client1`, `client2` ou `client3` et de publier dans n'importe quelle rubrique ayant l'ID client en préfixe :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
```

```
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/*"
            ]
        }
    ]
}
```

Vous pouvez également utiliser le caractère générique * à la fin d'un filtre de rubrique. L'utilisation de caractères génériques pourrait conduire à accorder accidentellement certains privilèges. Veillez donc à les utiliser avec prudence. Ils peuvent être utiles dans les cas où des appareils doivent s'abonner aux messages associés à de nombreuses rubriques différentes (par exemple, si un appareil doit s'abonner aux rapports générés par des capteurs de température situés à plusieurs endroits).

Registered devices (6)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant le nom d'objet de l'appareil en tant qu'ID client, et de s'abonner à n'importe quelle rubrique ayant le nom d'objet en préfixe, suivi de `room`, lui-même suivi de n'importe quelle chaîne (ces rubriques pourraient logiquement s'appeler, par exemple, `thing1/room1`, `thing1/room2` et ainsi de suite) :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/
${iot:Connection.Thing.ThingName}/room*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}/room*"
```

```
            ]
        }
    ]
}
```

Unregistered devices (6)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant les ID client `client1`, `client2`, `client3` et de s'abonner à une rubrique ayant l'ID client en préfixe, suivi de `room`, lui-même suivi de n'importe quelle chaîne (ces rubriques pourraient logiquement s'appeler, par exemple, `client1/room1`, `client1/room2` et ainsi de suite) :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/${iot:ClientId}/room*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/room*"
            ]
        }
    ]
}
```

Lorsque vous spécifiez des filtres de rubrique dans AWS IoT Core stratégies pour les clients MQTT, caractères génériques MQTT+and#sont traités comme des chaînes littérales. Leur utilisation peut générer un comportement inattendu.

Registered devices (4)

Pour les appareils enregistrés en tant qu'objets dans le AWS IoT Core La stratégie suivante accorde à l'autorisation de se connecter à AWS IoT Core L'ID client qui correspond au nom d'objet et de s'abonner au filtre de rubrique`some/+/topic`uniquement Remarque dans`topicfilter/some/+/topic`de l'ARN de ressource, + est traité comme une chaîne littérale dans AWS IoT Core pour les clients MQTT, ce qui signifie que seule la chaîne`some/+/topic`correspond au filtre de rubriques.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/some/+/topic"
            ]
        }
    ]
}
```

Pour les appareils enregistrés en tant qu'objets dans le AWS IoT Core La stratégie suivante accorde à l'autorisation de se connecter à AWS IoT Core L'ID client qui correspond au nom d'objet et de s'abonner au filtre de rubrique`some/*/topic`. Remarque dans`topicfilter/some/*/topic`de l'ARN de ressource, * est traité comme un caractère générique dans AWS IoT Core pour les clients MQTT, ce qui signifie que toute chaîne du niveau qui contient le caractère correspond au filtre de rubrique. (Ces rubriques pourraient logiquement s'appeler, par exemple,`some/string1/topic`,`some/string2/topic`, etc.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/some/*/topic"
            ]
        }
    ]
}
```

Unregistered devices (4)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le AWS IoT Core La stratégie suivante accorde à l'autorisation de se connecter à AWS IoT Core avec ID client`client1`et abonnez-vous au filtre de rubrique`some/+/topic`uniquement Remarque dans`topicfilter/some/+/`

`topic`de l'ARN de ressource, + est traité comme une chaîne littérale dans AWS IoT Core pour les clients MQTT, ce qui signifie que seule la chaîne`some/+/topic`correspond au filtre de rubriques.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/some/+/topic"
        ]
      }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le AWS IoT Core La stratégie suivante accorde à l'autorisation de se connecter à AWS IoT Core avec ID client`client1`et abonnez-vous au filtre de rubrique`some/+/topic`. Remarque dans`topicfilter/some/*/topic`de l'ARN de ressource, * est traité comme un caractère générique dans AWS IoT Core pour les clients MQTT, ce qui signifie que toute chaîne du niveau qui contient le caractère correspond au filtre de rubrique. (Ces rubriques pourraient logiquement s'appeler, par exemple,`some/string1/topic`,`some/string2/topic`, etc.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
      },
      {
        "Effect": "Allow",
        "Action": [
          "iot:Subscribe"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:topicfilter/some/*/topic"
        ]
      }
    ]
}
```

Note

Caractères génériques MQTT+and#sont traitées comme des chaînes littérales dans un AWS IoT Core pour les clients MQTT. Pour spécifier des caractères génériques dans les noms de rubrique et les filtres de rubrique dans AWS IoT Core pour les clients MQTT, vous devez utiliser*.

Registered devices (7)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant le nom d'objet de l'appareil en tant qu'ID client, et de s'abonner aux rubriques `my/topic` et `my/othertopic` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/othertopic"
            ]
        }
    ]
}
```

Unregistered devices (7)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide de l'ID client `client1`, et de s'abonner aux rubriques `my/topic` et `my/othertopic` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
```

```
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic",
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/othertopic"
            ]
        }
    ]
}
```

Registered devices (8)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant le nom d'objet de l'appareil en tant qu'ID client, et de s'abonner à une rubrique spécifiquement réservée à ce nom d'objet/ID client :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Thing.ThingName}"
            ]
        }
    ]
}
```

Unregistered devices (8)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide de l'ID client `client1`, et de publier dans une rubrique spécifiquement réservée à cet ID client :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
```

```
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
            ]
        }
    ]
}
```

Registered devices (9)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant le nom d'objet de l'appareil en tant qu'ID client, et de publier dans n'importe quelle rubrique ayant ce nom d'objet ou ID client en préfixe, à l'exception d'une rubrique se terminant par `bar` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/bar"
            ]
        }
    ]
}
```

Unregistered devices (9)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant les ID client `client1` et `client1`, et de publier dans n'importe quelle rubrique ayant comme préfixe l'ID client utilisé pour se connecter, à l'exception d'une rubrique se terminant par `bar` :

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

```
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}/bar"
            ]
        }
    ]
}
```

Registered devices (10)

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core en utilisant le nom d'objet de l'appareil en tant qu'ID client. L'appareil peut s'abonner à la rubrique`my/topic`, mais ne peut pas publier dans le*`thing-name`* `/bar`où*`nom-objet`*est le nom de l'objet IoT se connectant à AWS IoT Core :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
```

```
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/bar"
            ]
        }
    ]
}
```

Unregistered devices (10)

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide de l'ID client `client1`, et de s'abonner à la rubrique `my/topic` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        }
    ]
}
```

Les variables de stratégie d'objet sont également remplacées lorsqu'un certificat ou une identité Amazon Cognito authentifiée est attaché (e) à un objet. La stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core avec l'ID client `client1`, et de publier dans la rubrique `iotmonitor/provisioning/987654321098` et de recevoir celle-ci. Elle autorise également le titulaire du certificat à s'abonner à cette rubrique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [
```

```
                "arn:aws:iot:us-east-1:123456789012:topic/iotmonitor/
provisioning/987654321098"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/iotmonitor/
provisioning/987654321098"
            ]
        }
    ]
}
```

Stratégies pour les clients HTTP et WebSocket

Les identités Amazon Cognito peuvent être authentifiées ou non. Les identités authentifiées appartiennent aux utilisateurs authentifiés par tout fournisseur d'identité pris en charge. Les identités non authentifiées appartiennent généralement aux utilisateurs invités qui ne s'authentifient pas avec un fournisseur d'identité. Amazon Cognito fournit un identifiant unique etAWSLes informations d'identification pour prendre en charge les identités non authentifiées.

Pour les opérations suivantes, AWS IoT Core utilise le traitement AWS IoT Core attachées aux identités Amazon Cognito (via le`AttachPolicy`API) afin de définir les autorisations attachées au groupe d'identités Amazon Cognito avec les identités authentifiées.

- `iot:Connect`
- `iot:Publish`
- `iot:Subscribe`
- `iot:Receive`
- `iot:GetThingShadow`
- `iot:UpdateThingShadow`
- `iot:DeleteThingShadow`

Cela signifie qu'une identité Amazon Cognito nécessite des autorisations à partir de la stratégie de rôle IAM attachée au groupe et de l'objet AWS IoT Core attachée à Amazon Cognito Identity via l' AWS IoT Core `AttachPolicy`API.

Les utilisateurs authentifiés et non authentifiés sont différents types d'identité. Si vous ne joignez pas une stratégieAWS IoTà l'identité Amazon Cognito, un utilisateur authentifié échoue à l'autorisation dansAWS IoTet n'a pas accès àAWS IoTressources et actions.

> **Note**
>
> Pour d'autres AWS IoT Core opérations ou pour des identités non authentifiées, AWS IoT Core ne définit pas les autorisations attachées au rôle de groupe d'identités Amazon Cognito. Pour les identités authentifiées et non authentifiées, c'est la stratégie la plus permissive que nous vous recommandons d'attacher au rôle de groupe Amazon Cognito.

HTTP

Pour autoriser les identités Amazon Cognito non authentifiées à publier des messages sur HTTP dans une rubrique spécifique à l'identité Amazon Cognito, attachez la stratégie IAM suivante au rôle de groupe d'identités Amazon Cognito :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-
identity.amazonaws.com:sub}"]
        }
    ]
}
```

Pour autoriser des utilisateurs authentifiés, attachez la stratégie précédente au rôle de groupe d'identités Amazon Cognito et à l'identité Amazon Cognito à l'aide de l'outil AWS IoT Core AttachPolicyAPI.

> **Note**
>
> Lorsque vous autorisez les identités Amazon Cognito, AWS IoT Core considère les deux stratégies et accorde le moins de privilèges spécifiés. Une action n'est autorisée que si les deux stratégies autorisent l'action demandée. Si l'une des politiques empêche une action, cette action n'est pas autorisée.

MQTT

Pour autoriser les identités Amazon Cognito non authentifiées à publier des messages MQTT sur WebSocket dans une rubrique spécifique à l'identité Amazon Cognito de votre compte, attachez la stratégie IAM suivante au rôle de groupe d'identités Amazon Cognito :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-
identity.amazonaws.com:sub}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/${cognito-
identity.amazonaws.com:sub}"]
        }
    ]
}
```

Pour autoriser des utilisateurs authentifiés, attachez la stratégie précédente au rôle de groupe d'identités Amazon Cognito et à l'identité Amazon Cognito à l'aide de l'outil AWS IoT Core AttachPolicyAPI.

> **Note**
>
> Lorsque vous autorisez les identités Amazon Cognito, AWS IoT Core considère les deux stratégies et accorde le moins de privilèges spécifiés. Une action n'est autorisée que si les deux stratégies autorisent l'action demandée. Si l'une des politiques empêche une action, cette action n'est pas autorisée.

Exemples de stratégies de réception

Registered devices (11)

Pour les appareils enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide d'un ID client qui correspond au nom d'objet, et de s'abonner à la rubrique `my/topic` et de recevoir des messages de celle-ci :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
            ]
        }
    ]
}
```

Unregistered devices (11)

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide de l'ID client `client1`, et de s'abonner à une seule rubrique et de recevoir des messages de celle-ci :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/client1"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
            ]
        }
    ]
}
```

Exemples de stratégies de connexion et de publication

Pour les appareils enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core avec un ID client qui correspond au nom de l'objet et impose à l'appareil de publier uniquement sur une rubrique MQTT spécifique à l'ID client ou au nom de l'objet. Pour qu'une connexion soit établie, le nom de l'objet doit être inscrit dans le registre AWS IoT Core et être authentifié via une identité ou un principal attaché(e) à l'objet :

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action":["iot:Publish"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.ThingName}"]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
      }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés en tant qu'objets dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core avec l'ID client `client1` et impose à l'appareil de publier uniquement sur une rubrique MQTT spécifique à cet ID client.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action":["iot:Publish"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:ClientId}"]
      },
      {
        "Effect": "Allow",
        "Action": ["iot:Connect"],
        "Resource": ["arn:aws:iot:us-east-1:123456789012:client/client1"]
      }
    ]
}
```

Exemples de stratégies de certificat

Pour les appareils enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide d'un ID client qui correspond à un nom d'objet, et de publier dans une rubrique dont le nom équivaut au `certificateId` du certificat que l'appareil a utilisé pour s'authentifier :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:CertificateId}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide des ID client `client1`, `client2` et `client3`, et de publier dans une rubrique dont le nom équivaut au `certificateId` du certificat que l'appareil a utilisé pour s'authentifier :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${iot:CertificateId}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        }
    ]
}
```

Pour les appareils enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide d'un ID client qui correspond au nom d'objet, et de publier dans une rubrique dont le nom équivaut au champ `CommonName` de l'objet du certificat que l'appareil a utilisé pour s'authentifier :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Certificate.Subject.CommonName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        }
    ]
}
```

Note

Dans cet exemple, le nom commun de l'objet du certificat est utilisé comme identifiant de rubrique, en supposant que le nom commun de l'objet est unique pour chaque certificat enregistré. Si les certificats sont partagés entre plusieurs appareils, le nom commun de l'objet est le même pour tous les appareils qui partagent ce certificat, ce qui autorise la publication dans la même rubrique à partir de plusieurs appareils (non recommandé).

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide des ID client `client1`, `client2` et `client3`, et de publier dans une rubrique dont le nom équivaut au champ `CommonName` de l'objet du certificat que l'appareil a utilisé pour s'authentifier :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Certificate.Subject.CommonName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        }
    ]
}
```

> **Note**
>
> Dans cet exemple, le nom commun de l'objet du certificat est utilisé comme identifiant de rubrique, en supposant que le nom commun de l'objet est unique pour chaque certificat enregistré. Si les certificats sont partagés entre plusieurs appareils, le nom commun de l'objet est le même pour tous les appareils qui partagent ce certificat, ce qui autorise la publication dans la même rubrique à partir de plusieurs appareils (non recommandé).

Pour les appareils enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide d'un ID client qui correspond au nom d'objet, et de publier dans une rubrique dont le nom contient le préfixe `admin/` lorsque le champ `Subject.CommonName.2` du certificat utilisé pour authentifier l'appareil est défini sur `Administrator` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/*"],
            "Condition": {
                "StringEquals": {
                    "iot:Certificate.Subject.CommonName.2": "Administrator"
                }
            }
        }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide des ID client `client1`, `client2` et `client3`, et de publier dans une rubrique dont le nom contient le préfixe `admin/` lorsque le champ `Subject.CommonName.2` du certificat utilisé pour authentifier l'appareil est défini sur `Administrator` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
```

```
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/*"],
            "Condition": {
                "StringEquals": {
                    "iot:Certificate.Subject.CommonName.2": "Administrator"
                }
            }
        }
    ]
}
```

Pour les appareils enregistrés dans le registre AWS IoT Core , la stratégie suivante autorise un appareil à utiliser son nom d'objet pour publier dans une rubrique spécifique incluant `admin/` suivi du `ThingName` lorsque l'un des champs `Subject.CommonName` du certificat utilisé pour authentifier l'appareil est défini sur `Administrator` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/
${iot:Connection.Thing.ThingName}"],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:Certificate.Subject.CommonName.List": "Administrator"
                }
            }
        }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde l'autorisation de se connecter à AWS IoT Core à l'aide des ID client `client1`, `client2` et `client3`, et de publier dans la rubrique `admin` lorsque l'un des champs `Subject.CommonName` du certificat utilisé pour authentifier l'appareil est défini sur `Administrator` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin"],
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:Certificate.Subject.CommonName.List": "Administrator"
                }
            }
        }
    ]
}
```

Exemples de stratégies d'objet

La stratégie suivante permet à un appareil de se connecter si le certificat utilisé pour s'authentifier auprès d' AWS IoT Core est attaché à l'objet pour lequel la stratégie est évaluée :

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Effect":"Allow",
            "Action":["iot:Connect"],
            "Resource":[ "*" ],
            "Condition": {
                "Bool": {
                    "iot:Connection.Thing.IsAttached": ["true"]
                }
            }
        }
    ]
}
```

Autorisation avec les identités Amazon Cognito

Il existe deux types d'identités Amazon Cognito : authentifiées et non authentifiées. Lorsque votre application prend en charge les identités Amazon Cognito non authentifiées, aucune authentification n'est effectuée, si bien que vous ne savez pas qui est l'utilisateur. Pour les utilisateurs non authentifiés, vous accordez une autorisation en attachant un rôle IAM à un groupe d'identités non authentifiées. Vous devez accorder l'accès uniquement aux ressources que vous souhaitez mettre à la disposition des utilisateurs inconnus.

Lorsque votre application prend en charge les identités Amazon Cognito authentifiées, vous spécifiez une stratégie à deux endroits. Vous associez une stratégie IAM au groupe d'identités Amazon Cognito authentifiées et vous associez un élément AWS IoT Core à Amazon Cognito Identity Identity Identity Identity Identity Vous associez une stratégie IAM à un pool Amazon Cognito Identity à l'aide de la console Amazon Cognito Identity lorsque vous créez un pool Amazon Cognito Identity. Pour attacher un AWS IoT Core À une identité Amazon Cognito, vous devez définir une fonction Lambda qui appelle`AttachPolicy`.

> Note
>
> L'objet contextuel de votre fonction Lambda contient une valeur pour`context.cognito_identity_id`Lorsque vous appelez la fonction avecAWSque vous obtenez via les pools d'identités Amazon Cognito. Pour plus d'informations, consultez les rubriques suivantes.
>
> - AWS Lambda Objet de contexte dans Node.js

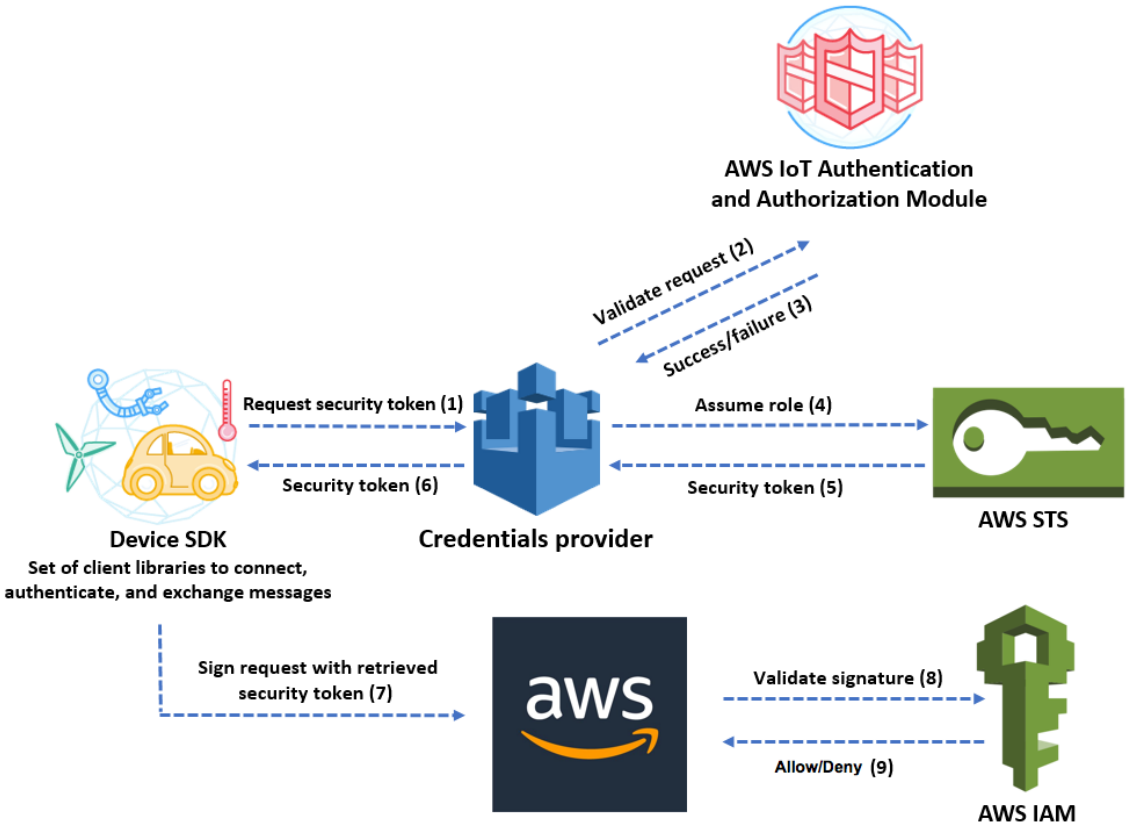
- AWS Lambda Objet de contexte dans Python
- AWS Lambda Objet de contexte en Ruby
- AWS Lambda Objet de contexte en Java
- AWS Lambda Objet de contexte dans Go
- AWS Lambda Objet de contexte en C#
- AWS Lambda Objet de contexte dans PowerShell

Autorisation des appels directs àAWSservices

Les appareils peuvent utiliser des certificats X.509 pour se connecter à AWS IoT Core en utilisant les protocoles d'authentification mutuelle TLS. AutreAWSLes services ne prennent pas en charge l'authentification par certificat, mais ils peuvent être appelés avecAWSInformations d'identification dansAWSSignature Version 4. La .Algorithme Signature Version 4L'appelant doit normalement disposer d'un ID de clé d'accès et d'une clé d'accès secrète. AWS IoT Core dispose d'un fournisseur d'informations d'identification qui vous permet d'utiliser leun certificat X.509comme identité de périphérique unique à authentifierAWSDemandes. Ainsi, vous n'avez plus besoin de stocker un ID de clé d'accès et une clé d'accès secrète sur votre appareil.

Le fournisseur d'informations d'identification authentifie un mandataire en utilisant un certificat X.509 et émet un jeton de sécurité temporaire à privilèges limités. Le jeton peut servir à signer et authentifier n'importe quelAWSde la demande. Cette façon d'authentifier votreAWSnécessite que vous créez et configurez unAWS Identity and Access Management(IAM)et attachez les stratégies IAM appropriées au rôle afin que le fournisseur d'informations d'identification puisse endosser le rôle en votre nom. Pour plus d'informations sur AWS IoT Core et IAM, voirGestion des identités et des accès pour AWS IoT (p. 315).

Le schéma suivant illustre le flux de travail du fournisseur d'informations d'identification.



1. L'appareil AWS IoT Core adresse une demande HTTPS au fournisseur d'informations d'identification pour demander un jeton de sécurité. La demande inclut le certificat X.509 de l'appareil pour l'authentification.
2. Le fournisseur d'informations d'identification transmet la demande au module d'authentification et d'autorisation AWS IoT Core pour valider le certificat et vérifier que l'appareil est autorisé à demander le jeton de sécurité.
3. Si le certificat est valide et qu'il est autorisé à demander un jeton de sécurité, le module d'authentification et d'autorisation AWS IoT Core renvoie un message de réussite. Dans le cas contraire, il envoie une exception à l'appareil.
4. Une fois que le fournisseur d'informations d'identification a validé le certificat, il appelle AWS Security Token Service (AWS STS) pour endosser le rôle IAM que vous avez créé à son intention.
5. AWS STS renvoie un jeton de sécurité temporaire à privilèges limités au fournisseur d'informations d'identification.
6. Le fournisseur d'informations d'identification renvoie le jeton de sécurité à l'appareil.
7. Le dispositif utilise le jeton de sécurité pour signer unAWSde la demande avecAWSSignature Version 4.
8. Le service demandé invoque IAM à valider la signature et à autoriser la demande par rapport aux stratégies d'accès attachées au rôle IAM que vous avez créé pour le fournisseur d'informations d'identification.
9. Si IAM valide la signature avec succès et autorise la demande, celle-ci aboutit. Sinon, IAM envoie une exception.

La section suivante explique comment utiliser un certificat pour obtenir un jeton de sécurité. Elle est rédigée en partant du principe que vous avez déjà enregistré un appareil et créé, puis activé votre propre certificat pour celui-ci.

Comment utiliser un certificat pour obtenir un jeton de sécurité

1. Configurez le rôle IAM que le fournisseur d'informations d'identification endosse au nom de votre appareil. Attachez la stratégie d'approbation suivante au rôle.

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

   Pour chaqueAWSque vous souhaitez appeler, attachez une stratégie d'accès au rôle. Le fournisseur d'informations d'identification prend en charge les variables de stratégie suivantes :

   - `credentials-iot:ThingName`
   - `credentials-iot:ThingTypeName`
   - `credentials-iot:AwsCertificateId`

   Lorsque l'appareil fournit le nom d'objet dans sa demande adressée à unAWS, le fournisseur d'informations d'identification ajoute`credentials-iot:ThingName`and`credentials-iot:ThingTypeName`en tant que variables de contexte au jeton de sécurité. Le fournisseur d'informations d'identification fournit `credentials-iot:AwsCertificateId` en tant que variable de contexte, même si l'appareil ne fournit pas le nom d'objet dans la demande. Vous transmettez le nom d'objet comme valeur de l'en-tête de la demande HTTP `x-amzn-iot-thingname`.

Ces trois variables opèrent uniquement pour les stratégies IAM, et non pour les stratégies AWS IoT Core .

2. Vérifiez que l'utilisateur qui effectue l'étape suivante (création d'un alias de rôle) est autorisé à transmettre le rôle nouvellement créé à AWS IoT Core . La politique suivante donne à la fois`iam:GetRole`and`iam:PassRole`vers unAWSUtilisateur. L'autorisation `iam:GetRole` autorise l'utilisateur à obtenir des informations sur le rôle que vous venez de créer. La .`iam:PassRole`L'autorisation permet à l'utilisateur de transmettre le rôle à un autreAWSService.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Allow",
        "Action": [
            "iam:GetRole",
            "iam:PassRole"
        ],
        "Resource": "arn:aws:iam::your
    Compte AWS
    id:role/your role name"
    }
}
```

3. Créez un alias de rôle AWS IoT Core . L'appareil qui va passer des appels directs àAWSLes services doivent savoir quel ARN de rôle utiliser au moment de se connecter à AWS IoT Core . Coder en dur l'ARN de rôle n'est pas une bonne solution, car cela vous contraint de mettre à jour l'appareil chaque fois que l'ARN de rôle est modifié. Il vaut mieux utiliser l'API `CreateRoleAlias` pour créer un alias de rôle qui pointe vers l'ARN de rôle. Si l'ARN de rôle est modifié, il vous suffit de mettre à jour l'alias de rôle. Aucune modification n'est nécessaire sur l'appareil. Cette API accepte les paramètres suivants :

   `roleAlias`

   Obligatoire. Chaîne arbitraire qui identifie l'alias de rôle. Elle fait office de clé primaire dans le modèle de données d'alias de rôle. Elle contient entre 1 et 128 caractères et doit se composer uniquement de caractères alphanumériques et de symboles =, @ et -. Les caractères alphabétiques majuscules et minuscules sont autorisés.

   `roleArn`

   Obligatoire. ARN du rôle auquel l'alias de rôle fait référence.

   `credentialDurationInSeconds`

   Facultatif. Durée de validité (en secondes) des informations d'identification. La valeur minimale est de 900 secondes (15 minutes). La valeur maximale est de 3 600 secondes (60 minutes). La valeur par défaut est de 3 600 secondes.

   Note

   Bien que la durée de vie des informations d'identification spécifiée dans le rôle IAM puisse être plus longue, lorsque AWS IoT Core Le fournisseur d'informations d'identification émet les informations d'identification, sa durée de vie maximale est de 3 600 secondes (60 minutes).

   Pour de plus amples informations sur cette API, veuillez consulter CreateRoleAlias.

4. Attachez une stratégie au certificat de l'appareil. La stratégie attachée au certificat de l'appareil doit accorder à l'appareil l'autorisation d'assumer le rôle. Pour ce faire, vous devez accorder une autorisation à l'alias de rôle pour l'action `iot:AssumeRoleWithCertificate`, comme dans l'exemple suivant.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"iot:AssumeRoleWithCertificate",
         "Resource":"arn:aws:iot:your region:your_aws_account_id:rolealias/your role
 alias"
      }
   ]
}
```

5. Adressez une demande HTTPS au fournisseur d'informations d'identification pour obtenir un jeton de sécurité. Fournissez les informations suivantes :

   - Certificat : S'agissant d'une requête HTTP avec l'authentification mutuelle TLS, vous devez fournir le certificat et la clé privée à votre client lorsque vous faites la demande. Servez-vous du certificat et de la clé privée que vous avez utilisés lorsque vous avez enregistré votre certificat auprès d' AWS IoT Core .

     Pour vérifier que votre appareil communique avec AWS IoT Core (et non avec un service usurpant son identité), consultez Authentification du serveur, suivez les liens pour télécharger les certificats CA appropriés, puis copiez-les sur votre appareil.
   - RoleAlias : Nom de l'alias de rôle que vous avez créée pour le fournisseur d'informations d'identification.
   - ThingName : Le nom d'objet que vous avez créé au moment d'enregistrer votre AWS IoT Core Un objet. Celui-ci est transmis comme valeur de l'en-tête HTTP `x-amzn-iot-thingname`. Cette valeur n'est obligatoire que si vous utilisez des attributs d'objet en tant que variables de stratégie dans des stratégies AWS IoT Core ou IAM.

     Note

     La .ThingNameque vous fournissez dans`x-amzn-iot-thingname`doit correspondre au nom de la stratégieAWS IoTRessource de chose assignée à un certificat. Si elle ne correspond pas, une erreur 403 est renvoyée.

   Exécutez la commande suivante dans l'AWS CLIPour obtenir le point de terminaison du fournisseur d'informations d'identification pour votre Compte AWS . Pour en savoir plus sur cette API, consultez DescribeEndpoint.

   aws iot describe-endpoint --endpoint-type iot:CredentialProvider

   L'objet JSON ci-dessous est un exemple de sortie de la commande describe-endpoint. Il contient le paramètre `endpointAddress` que vous utilisez pour demander un jeton de sécurité.

```
{
    "endpointAddress": "your_aws_account_specific_prefix.credentials.iot.your
 region.amazonaws.com"
}
```

   Utilisez le point de terminaison pour adresser une demande HTTPS au fournisseur d'informations d'identification pour qu'il renvoie un jeton de sécurité. L'exemple de commande suivant utilise `curl`, mais vous pouvez utiliser n'importe quel client HTTP.

   curl --cert ***your certificate*** --key ***your device certificate key pair*** -H "x-amzn-iot-thingname: ***your thing name***" --cacert AmazonRootCA1.pem https://***your endpoint*** /role-aliases/***your role alias***/credentials

Cette commande renvoie un objet de jeton de sécurité qui contient les éléments `accessKeyId`, `secretAccessKey`, `sessionToken`, ainsi qu'un délai d'expiration. L'objet JSON ci-dessous est un exemple de sortie de la commande `curl`.

```
   {"credentials":{"accessKeyId":"access key","secretAccessKey":"secret access
key","sessionToken":"session token","expiration":"2018-01-18T09:18:06Z"}}
```

Vous pouvez ensuite utiliser la stratégie`accessKeyId`,`secretAccessKey`, et`sessionToken`pour signer des demandes àAWSServices . Pour une démonstration de bout en bout en bout, consultezComment faire pour éliminer le besoin de codage en durAWSInformations d'identification dans les périphériques à l'aide de laAWS IoTFournisseur d'informations d'identificationArticle de blog surAWSBlog Sécurité.

Accès entre comptes avec IAM

AWS IoT Core permet d'autoriser un utilisateur habilité à publier ou à s'abonner à une rubrique définie dans une identité Compte AWS Ne sont pas détenues par le mandataire. Vous configurez l'accès entre comptes en créant une stratégie IAM et un rôle IAM, puis en attachant la stratégie au rôle.

Tout d'abord, créez une stratégie IAM gérée par le client comme décrit dansCrée des stratégies IAM, comme vous le feriez pour d'autres utilisateurs et certificats dans votre Compte AWS .

Pour les appareils enregistrés dans AWS IoT Core , la stratégie suivante accorde à des appareils qui se connectent à AWS IoT Core à l'aide d'un ID client correspondant au nom de chose de l'appareil et à publier dans le répertoire`my/topic/`*`thing-name`* où*`nom-objet`*est le nom de chose de l'appareil :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/my/topic/
${iot:Connection.Thing.ThingName}"],
        }
    ]
}
```

Pour les appareils qui ne sont pas enregistrés dans le registre AWS IoT Core , la stratégie suivante accorde à un appareil l'autorisation d'utiliser le nom d'objet `client1` enregistré dans le registre AWS IoT Core de votre compte (123456789012) pour se connecter à AWS IoT Core , et de publier dans une rubrique spécifique à l'ID client dont le nom contient le préfixe `my/topic/` :

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

```
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
            ]
        }
    ]
}
```

Suivez ensuite les étapes décrites dans Création d'un rôle pour déléguer des autorisations à un utilisateur IAM. Entrez l'ID de compte du Compte AWS avec lequel vous souhaitez partager l'accès. Puis, dans la dernière étape, attachez la stratégie que vous venez de créer au rôle. Si, ultérieurement, vous devez modifier l'objetAWSL'ID de compte auquel vous accordez l'accès, vous pouvez utiliser le format de stratégie d'approbation suivant à cet effet :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam:us-east-1:567890123456:user:MyUser"
            },
            "Action": "sts:AssumeRole",
        }
    ]
}
```

Protection des données dans AWS IoT Core

La .AWS Modèle de responsabilité partagées'applique à la protection des données dans AWS IoT Core . Comme décrit dans ce modèle, AWS est responsable de la protection de l'infrastructure globale sur laquelle l'ensemble du cloud AWS s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure est de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité des services AWS que vous utilisez. Pour de plus amples informations sur la confidentialité des données, veuillez consulter FAQ sur la confidentialité des données. Pour de plus amples informations sur la protection des données en Europe, veuillez consulter le billet de blog Modèle de responsabilité partagée AWS et RGPD sur le Blog de sécurité AWS.

À des fins de protection des données, nous vous recommandons de protéger les autorisations du compte AWS et de configurer les comptes d'utilisateur individuels avec AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multi-facteurs (MFA) avec chaque compte.
- Utilisez SSL/TLS pour communiquer avec des ressources AWS. Nous recommandons TLS 1.2 ou version ultérieure.

- Configurez l'API et la consignation des activités utilisateur avec AWS CloudTrail.
- Utilisez des solutions de chiffrement AWS, ainsi que tous les contrôles de sécurité par défaut au sein des services AWS.
- Utilisez des services de sécurité gérés avancés tels que Amazon Macie, qui contribuent à la découverte et à la sécurisation des données personnelles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS 140-2 lorsque vous accédez à AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour de plus amples informations sur les points de terminaison FIPS disponibles, consultez Federal Information Processing Standard (FIPS) 140-2.

Nous vous recommandons vivement de ne jamais placer d'informations identifiables sensibles, telles que des adresses e-mail de vos clients, dans des balises ou des champs de formulaire comme uneNomfield. Cela s'applique aussi lorsque vous utilisez AWS IoT ou d'autres services AWS à l'aide de la console, de l'API, de l'interface de ligne de commande (AWS CLI) ou des kits AWS SDK. Toutes les données que vous entrez dans les balises ou les champs de forme libre utilisés pour les noms peuvent être utilisées pour les journaux de facturation ou de diagnostic. Lorsque vous fournissez une URL à un serveur externe, n'incluez pas les informations d'identification non chiffrées dans l'URL pour valider votre demande adressée au serveur.

Pour en savoir plus sur la protection des données, veuillez consulter le billet de blog Modèle de responsabilité partagée AWS et RGPD sur le Blog AWS Security.

Les appareils AWS IoT collectent des données, effectuent une manipulation sur ces données, puis les envoient à un autre service Web. Vous pouvez choisir de stocker certaines données sur votre appareil pendant une courte durée. Il vous incombe d'assurer la protection de ces données au repos. Lorsque votre appareil envoie des données àAWS IoT, il le fait via une connexion TLS, comme indiqué plus loin dans cette section.AWS IoTpeuvent envoyer des données à n'importe quelAWSService. Pour de plus amples informations sur la sécurité des données de chaque service, veuillez consulter la documentation de ce service.AWS IoTpeut être configuré pour écrire des fichiers journaux dans CloudWatch Logs et journalAWS IoTAppels d'API àAWS CloudTrail. Pour plus d'informations sur la sécurité des données pour ces services, consultez.Authentification et contrôle d'accès pour Amazon CloudWatchandChiffrez les fichiers journaux CloudTrail avecAWSClés gérées par KMS.

Chiffrement des données dans AWS IoT

Par défaut, tous lesAWS IoTLes données en transit et au repos sont chiffrées. Les données en transit sont chiffrées à l'aide du protocole TLS (p. 313), et les données au repos sont chiffrées à l'aide deAWSclés possédées.AWS IoTne prend pas actuellement en charge les clés principales client gérées par le client (clés CMK) depuisAWSService de gestion des clés (AWS KMS) ; cependant, Device Advisor etAWS IoTUtilisation sans fil uniquement d'un KMSAWSClé maître client (AOCMK) possédée pour chiffrer les données client.

Sécurité du transport dans AWS IoT

La .AWS IoTL'agent de messages et le service Device Shadow chiffrent toutes les communications pendant le transit à l'aide deTLS version 1.2. TLS est utilisé afin de garantir la confidentialité des protocoles d'application (MQTT, HTTP et WebSocket) pris en charge parAWS IoT. La prise en charge TLS est disponible dans un certain nombre de langages de programmation et de systèmes d'exploitation. Données dansAWSest crypté par leAWSService. Pour plus d'informations sur le chiffrement des données dans d'autres services AWS, consultez la documentation relative à la sécurité du service concerné.

Pour MQTT, TLS chiffre la connexion entre l'appareil et l'agent. L'authentification de client TLS est utilisée par AWS IoT pour identifier les appareils. Pour HTTP, TLS chiffre la connexion entre l'appareil et l'agent. L'authentification est déléguée àAWSSignature Version 4.

AWS IoT exige que les appareils envoient l'extension SNI (Server Name Indication) au protocole TLS (Transport Layer Security) et fournissent l'adresse complète du point de terminaison dans le champ `host_name`. Le champ `host_name` doit contenir le point de terminaison que vous appelez, et il doit être :

- L'adresse `endpointAddress` renvoyée par aws iot describe-endpoint --endpoint-type iot:Data-ATS

  ou
- L'adresse `domainName` renvoyée par aws iot describe-domain-configuration –-domain-configuration-name "***`domain_configuration_name`***"

Les connexions tentées par des appareils sans l'`host_name`sera refusée et enregistrée dans CloudWatch.

AWS IoTne prend pas en charge l'extension SessionTicket TLS.

Sécurité du transport pour les appareils sans fil LoRaWan

Les appareils LoRaWan suivent les pratiques de sécurité décrites dansSÉCURITÉ LoRaWan™ : Livre blanc préparé pour la LoRa Alliance™ par Gemalto, Actility et Semtech.

Pour plus d'informations sur la sécurité du transport avec les appareils LoRaWan, consultez.Sécurité des données avec AWS IoT Core pour LoRaWAN (p. 1123).

Prise en charge des suites de chiffrement TLS

AWS IoT prend en charge les suites de chiffrement suivantes :

- ECDHE-ECDSA-AES128-GCM-SHA256 (recommandée)
- ECDHE-RSA-AES128-GCM-SHA256 (recommandée)
- ECDHE-ECDSA-AES128-SHA256
- ECDHE-RSA-AES128-SHA256
- ECDHE-ECDSA-AES128-SHA
- ECDHE-RSA-AES128-SHA
- ECDHE-ECDSA-AES256-GCM-SHA384
- ECDHE-RSA-AES256-GCM-SHA384
- ECDHE-ECDSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA384
- ECDHE-RSA-AES256-SHA
- ECDHE-ECDSA-AES256-SHA
- AES128-GCM-SHA256
- AES128-SHA256
- AES128-SHA
- AES256-GCM-SHA384
- AES256-SHA256
- AES256-SHA

Chiffrement des données dans AWS IoT

La protection des données fait référence au fait de protéger les données lorsqu'elles sont en transit (lorsqu'elles sont transmises à AWS IoT ou à partir de celui-ci) et au repos (lorsqu'elles sont stockées sur des appareils ou par d'autres services AWS). Toutes les données envoyées àAWS IoTest envoyé via une connexion TLS à l'aide des protocoles MQTT, HTTPS et WebSocket, ce qui le rend sécurisé par défaut

pendant le transit.AWS IoTcollectent des données, puis les envoient à d'autresAWSservices de traitement ultérieur. Pour plus d'informations sur le chiffrement des données dans d'autres services AWS, consultez la documentation relative à la sécurité du service concerné.

FreeRTOS fournit une bibliothèque PKCS #11 qui isole le stockage des clés, en accédant aux objets cryptographiques et en gérant les sessions. Il vous incombe d'utiliser cette bibliothèque pour chiffrer les données au repos sur vos appareils. Pour de plus amples informations, veuillez consulterBibliothèque FreeRTOS Public Key Cryptography Standard (PKCS) #11.

Device Advisor

Chiffrement en transit

Les données envoyées vers et depuis Device Advisor sont cryptées en transit. Toutes les données envoyées vers et depuis le service lors de l'utilisation des API Device Advisor sont chiffrées à l'aide de Signature Version 4. Pour plus d'informations sur la façon dontAWSLes demandes d'API sont signées, consultez.SignatureAWSDemandes d'API. Toutes les données envoyées depuis vos périphériques de test vers votre terminal de test Device Advisor sont envoyées via une connexion TLS afin qu'elles soient sécurisées par défaut pendant le transit.

Gestion des clés dans AWS IoT

Toutes les connexions à AWS IoT sont établies à l'aide de TLS, par conséquent, aucune clé de chiffrement côté client n'est nécessaire pour la connexion TLS initiale.

Les appareils doivent s'authentifier à l'aide d'un certificat X.509 ou d'une identité Amazon Cognito. Vous pouvez demander à AWS IoT de générer un certificat pour vous, auquel cas il générera une paire de clés publique/privée. Si vous utilisez la console AWS IoT, vous serez invité à télécharger le certificat et les clés. Si vous utilisez la stratégie`create-keys-and-certificate`CLI, le certificat et les clés sont renvoyés par la commande CLI. Il vous incombe de copier le certificat et la clé privée sur votre appareil et de veiller à leur sécurité.

AWS IoTne prend pas actuellement en charge les clés principales client gérées par le client (clés CMK) depuisAWS Key Management Service(AWS KMS) ; cependant, Device Advisor etAWS IoTWireless utilise uniquement unAWS KMSClé maître client (AOCMK) possédée pour chiffrer les données client.

Device Advisor

Toutes les données envoyées à Device Advisor lors de l'utilisation duAWSLes API sont chiffrées au repos. Device Advisor chiffre toutes vos données au repos à l'aide deAWS KMSLes clés principales client (CMK) stockées et gérées dansAWS Key Management Service. Device Advisor chiffre vos données avecAWS-appartenant aux clés principales client. Pour plus d'informations surAWSLes clés principales client, consultez.AWSKits CMK détenues par.

Gestion des identités et des accès pour AWS IoT

AWS Identity and Access Management(IAM) est unAWSqui aide un administrateur à contrôler en toute sécurité l'accès àAWSAWS. Les administrateurs IAM contrôlent qui peut êtreAuthentifié(connecté) etAuthorized(avoir des autorisations) pour utiliserAWS IoTAWS. IAM est un service AWS que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Audience

Utilisation deAWS Identity and Access Management(IAM) diffère selon le travail que vous effectuez dansAWS IoT.

Utilisateur du service— Si vous utilisez l'AWS IoTPour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Plus vous utiliserez de fonctionnalités AWS IoT pour effectuer votre travail, plus vous pourrez avoir besoin d'autorisations supplémentaires. Comprendre la gestion des accès peut vous aider à demander à votre administrateur les autorisations appropriées. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS IoT, consultez Résolution des problèmes liés à Identity and Access AWS IoT (p. 341).

Administrateur de service— Si vous êtes en charge deAWS IoTLes ressources de votre entreprise, vous bénéficiez probablement d'un accès total àAWS IoT. C'est à vous de déterminer les fonctionnalités et les ressources AWS IoT auxquelles vos employés pourront accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser AWS IoT, veuillez consulter Fonctionnement d'AWS IoT avec IAM (p. 320).

Administrateur IAM— Si vous êtes un administrateur IAM, vous souhaiterez peut-être obtenir des détails sur la façon dont vous pouvez écrire des stratégies pour gérer l'accès àAWS IoT. Pour voir des exemples de stratégies AWS IoT basées sur l'identité que vous pouvez utiliser dans IAM, veuillez consulter Exemples de stratégies basées sur l'identité AWS IoT (p. 338).

Authentification avec des identités IAM

DansAWS IoTLes identités peuvent être des certificats d'appareil (X.509), des identités Amazon Cognito ou des utilisateurs ou des groupes IAM. Cette rubrique traite uniquement des identités IAM. Pour plus d'informations sur les autres identités prises en charge par AWS IoT, consultez Authentification client (p. 236).

L'authentification correspond au processus par lequel vous vous connectez à AWS via vos informations d'identification. Pour de plus amples informations sur la connexion avec l'AWS Management Console, veuillez consulter Connexion à l'AWS Management Console en tant qu'utilisateur IAM ou utilisateur racine dans le Guide de l’utilisateur IAM.

Vous devez vous authentifier (être connecté à AWS) en tant qu'utilisateur racine du compte AWS, utilisateur IAM ou en endossant un rôle IAM. Vous pouvez également utiliser l'authentification de connexion unique de votre entreprise ou vous connecter par le biais de Google ou de Facebook. Dans ces cas, votre administrateur aura précédemment configuré une fédération d’identités avec des rôles IAM. Lorsque vous accédez à AWS avec des informations d’identification d’une autre entreprise, vous assumez indirectement un rôle.

Pour vous connecter directement à l'AWS Management Console, utilisez votre mot de passe avec votre adresse e-mail d’utilisateur racine ou votre nom d'utilisateur IAM. Vous pouvez accéder à AWS par programmation avec vos clés d'accès utilisateur IAM ou racine. AWS fournit un kit SDK et des outils de ligne de commande pour signer de manière chiffrée votre demande avec vos informations d'identification. Si vous n'utilisez pas les outils AWS, vous devez signer la demande vous-même. Pour ce faire, utilisez Signature Version 4, un protocole permettant d’authentifier les demandes d'API entrantes. Pour de plus amples informations sur l'authentification des demandes, veuillez consulter Processus de signature Signature Version 4 dans les Références générales AWS.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être également fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser la Multi-Factor Authentication (MFA) pour améliorer la sécurité de votre compte. Pour en savoir plus, veuillez consulter Utilisation de la Multi-Factor Authentication (MFA) dans AWS dans le Guide de l'utilisateur IAM.

Utilisateur racine d'un compte AWS

Lorsque vous créez pour la première fois un compte AWS, vous commencez avec une identité de connexion unique qui bénéficie d'un accès complet à tous les services et ressources AWS du compte. Cette identité est appelée l'utilisateur racine du compte AWS. Elle est accessible en vous connectant à l'aide de l'adresse e-mail et du mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes, y compris pour les tâches administratives. Respectez plutôt la bonne pratique qui consiste à avoir recours à l'utilisateur racine uniquement pour créer le premier utilisateur IAM. Ensuite, mettez en sécurité les informations d'identification de l'utilisateur racine et utilisez-les uniquement pour effectuer certaines tâches de gestion des comptes et des services.

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité dans votre compte AWS qui dispose d'autorisations spécifiques pour une seule personne ou application. Un utilisateur IAM peut disposer d'informations d'identification à long terme, comme un nom d'utilisateur et un mot de passe ou un ensemble de clés d'accès. Pour découvrir comment générer des clés d'accès, veuillez consulter Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM. Lorsque vous générez des clés d'accès pour un utilisateur IAM, veillez à afficher et enregistrer la paire de clés de manière sécurisée. Vous ne pourrez plus récupérer la clé d'accès secrète à l'avenir. Au lieu de cela, vous devrez générer une nouvelle paire de clés d'accès.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une entité au sein de votre compte AWS qui dispose d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez temporairement endosser un rôle IAM dans l'AWS Management Console grâce au changement de rôle. Vous pouvez obtenir un rôle en appelant une opération d'API AWS CLI ou AWS à l'aide d'une URL personnalisée. Pour de plus amples informations sur les méthodes d'utilisation des rôles, veuillez consulter Utilisation des rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Autorisations utilisateur IAM temporaires : Un utilisateur IAM peut endosser un rôle IAM pour accepter différentes autorisations temporaires concernant une tâche spécifique.
- Accès par des utilisateurs fédérés – Au lieu de créer un utilisateur IAM, vous pouvez utiliser des identités existantes provenant d'AWS Directory Service, de votre répertoire d'utilisateurs d'entreprise ou d'un fournisseur d'identité web. On parle alors d'utilisateurs fédérés. AWS attribue un rôle à un utilisateur fédéré lorsque l'accès est demandé via un fournisseur d'identité. Pour de plus amples informations sur les utilisateurs fédérés, veuillez consulter Utilisateurs fédérés et rôles dans le Guide de l'utilisateur IAM.

- Accès comptes multiples : Vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (mandataire de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès entre plusieurs comptes. Cependant, certains services AWS vous permettent d'attacher une stratégie directement à une ressource (au lieu d'utiliser un rôle en tant que proxy). Pour en savoir plus sur la différence entre les rôles et les stratégies basées sur les ressources pour l'accès comptes multiples, veuillez consulter Différence entre les rôles IAM et les stratégies basées sur les ressources dans le Guide de l'utilisateur IAM.
- Accès services multiples – Certains services AWS utilisent des fonctionnalités dans d'autres services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant pour ce service d'exécuter des applications dans Amazon EC2 ou de stocker des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du mandataire, un rôle de service ou un rôle lié au service.
  - Autorisations du mandataire – Lorsque vous utilisez un utilisateur ou un rôle IAM afin d'effectuer des actions dans AWS, vous êtes considéré comme mandataire. Les stratégies accordent des autorisations au mandataire. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui déclenche une autre action dans un autre service. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour savoir si une action nécessite d'autres actions supplémentaires dans une stratégie, consultezActions, ressources et clés de condition pourAWS IoTdans leRéférence de l'autorisation de service.
  - Rôle de service : Il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Les rôles de service fournissent un accès uniquement au sein de votre compte et ne peuvent pas être utilisés pour accorder l'accès à des services dans d'autres comptes. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour de plus amples informations, veuillez consulter Création d'un rôle pour la délégation d'autorisations à un service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié au service – Un rôle lié au service est un type de rôle de service lié à un service AWS. Le service peut assumer le rôle afin d'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.
- Applications qui s'exécutent sur Amazon EC2— Vous pouvez utiliser un rôle IAM pour gérer des informations d'identification temporaires pour les applications s'exécutant sur une instance EC2 et effectuer desAWS CLIouAWSDemandes d'API. Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un rôle AWS à une instance EC2 et le rendre disponible à toutes les applications associées, vous pouvez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour de plus amples informations, veuillez consulter Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, veuillez consulter Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

Gestion de l'accès à l'aide de stratégies

Vous contrôler les accès dans AWS en créant des stratégies et en les attachant à des identités AWS ou à des ressources . Une stratégie est un objet dans AWS qui, lorsqu'elle est associée à une identité ou à une ressource, définit leurs autorisations. Vous pouvez vous connecter en tant qu'utilisateur racine ou IAM ou vous pouvez endosser un rôle IAM. Lorsque vous effectuez ensuite une demande, AWS évalue les stratégies relatives basées sur l'identité ou les ressources. Les autorisations dans les stratégies déterminent si la demande est autorisée ou refusée. La plupart des stratégies sont stockées dans AWS en tant que documents JSON. Pour de plus amples informations sur la structure et le contenu des documents de stratégie JSON, veuillez consulter Présentation des stratégies JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les stratégies JSON AWS pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

Chaque entité IAM (utilisateur ou rôle) démarre sans autorisation. En d'autres termes, par défaut, les utilisateurs ne peuvent rien faire, pas même changer leurs propres mots de passe. Pour autoriser un utilisateur à effectuer une opération, un administrateur doit associer une stratégie d'autorisations à ce dernier. Il peut également ajouter l'utilisateur à un groupe disposant des autorisations prévues. Lorsqu'un administrateur accorde des autorisations à un groupe, tous les utilisateurs de ce groupe se voient octroyer ces autorisations.

Les stratégies IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une stratégie qui autorise l'action `iam:GetRole`. Un utilisateur avec cette stratégie peut obtenir des informations utilisateur à partir de l'AWS Management Console, de l'AWS CLI ou de l'API AWS.

Stratégies basées sur l'identité

Les stratégies basées sur l'identité sont des documents de stratégie d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces stratégies contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une stratégie basée sur l'identité, veuillez consulter Création de stratégies IAM dans le Guide de l'utilisateur IAM.

Les stratégies basées sur l'identité peuvent être classées comme étant des stratégies en ligne ou des stratégies gérées. Les stratégies en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les stratégies gérées sont des stratégies autonomes que vous pouvez lier à plusieurs utilisateurs, groupes et rôles de votre compte AWS. Les stratégies gérées incluent les stratégies gérées par AWS et les stratégies gérées par le client. Pour découvrir comment choisir entre une stratégie gérée et une stratégie en ligne, veuillez consulter Choix entre les stratégies gérées et les stratégies en ligne dans le Guide de l'utilisateur IAM.

Stratégies basées sur les ressources

Les stratégies basées sur les ressources sont des documents de stratégie JSON que vous attachez à une ressource. Des stratégies basées sur les ressources sont par exemple, les stratégies de confiance de rôle IAM et des stratégies de compartiment Amazon S3. Dans les services qui sont compatibles avec les stratégies basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la stratégie, cette dernière définit quel type d'actions un mandataire spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un mandataire dans une stratégie basée sur les ressources. Les mandataires peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou des services AWS.

Les stratégies basées sur les ressources sont des stratégies en ligne situées dans ce service. Vous ne pouvez pas utiliser les stratégies gérées AWS depuis IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels mandataires (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux stratégies basées sur les ressources, bien qu'elles n'utilisent pas le format de document de stratégie JSON.

Amazon S3, AWS WAF et Amazon VPC sont des exemples de services prenant en charge les listes ACL. Pour en savoir plus sur les listes de contrôle d'accès, veuillez consulter Présentation des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de stratégie

AWS prend en charge d'autres types de stratégies moins courantes. Ces types de stratégies peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de stratégies plus courants.

- Limite d'autorisations : Une limite d'autorisations est une fonction avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une stratégie basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations obtenues représentent la combinaison des stratégies basées sur l'identité de l'entité et de ses limites d'autorisations. Les stratégies basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour de plus amples informations sur les limites d'autorisations, veuillez consulter Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) – Les SCP sont des stratégies JSON qui spécifient le nombre maximal d'autorisations pour une organisation ou une unité d'organisation (OU) dans AWS Organizations. AWS Organizations est un service qui vous permet de regrouper et de gérer de façon centralisée plusieurs comptes AWS détenus par votre entreprise. Si vous activez toutes les fonctions d'une organisation, vous pouvez appliquer les stratégies de contrôle de service (SCP) à l'un ou à l'ensemble de vos comptes. Les politiques de contrôle des services (SCP) limitent les autorisations pour les entités dans les comptes membres, y compris chaque utilisateur racine de compte AWS. Pour de plus amples informations sur les organisations et les SCP, veuillez consulter Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations.
- Stratégies de session : Les stratégies de session sont des stratégies avancées que vous passez en tant que paramètre lorsque vous programmez afin de créer une session temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la session obtenue sont une combinaison des stratégies basées sur l'identité de l'utilisateur ou du rôle et des stratégies de session. Les autorisations peuvent également provenir d'une stratégie basée sur les ressources. Un refus explicite dans l'une de ces stratégies remplace l'autorisation. Pour de plus amples informations, veuillez consulter Stratégies de session dans le Guide de l'utilisateur IAM.

Plusieurs types de stratégie

Lorsque plusieurs types de stratégies s'appliquent à la requête, les autorisations obtenues sont plus compliquées à comprendre. Pour découvrir la façon dont AWS détermine s'il convient d'autoriser une demande en présence de plusieurs types de stratégies, consultez Logique d'évaluation de stratégies dans le Guide de l'utilisateur IAM.

Fonctionnement d'AWS IoT avec IAM

Avant d'utiliser IAM pour gérer l'accès àAWS IoT, vous devez comprendre quelles sont les fonctions IAM disponibles à utiliser avecAWS IoT. Pour obtenir une vue globale de la façon dontAWS IoTet autresAWSfonctionnent avec IAM, consultezAWSServices qui fonctionnent avec IAMdans leGuide de l'utilisateur IAM.

Rubriques

- Stratégies IAM (p. 320)
- Stratégies basées sur l'identité AWS IoT (p. 322)
- Stratégies basées sur les ressources AWS IoT (p. 337)
- Autorisation basée sur les balises AWS IoT (p. 337)
- AWS IoTRôles IAM (p. 338)

Stratégies IAM

AWS IoTest compatible avecAWS IoTet stratégies IAM. Cette rubrique traite uniquement des stratégies IAM. Pour plus d'informations, consultez Stratégies AWS IoT Core (p. 270). AWS Identity and Access Management définit une action de stratégie pour chaque opération définie par AWS IoT, y compris les API de plan de contrôle et de plan de données.

Stratégies gérées IAM

AWS IoTpropose un ensemble de stratégies gérées par IAM que vous pouvez utiliser telles quelles ou comme point de départ pour la création de stratégies IAM personnalisées. Ces stratégies permettent d'accéder à des opérations de configuration et sur les données. Les opérations de configuration vous permettent de créer des objets, des certificats, des stratégies et des règles. Les opérations sur les données envoient des données via les protocoles MQTT ou HTTP. Le tableau suivant décrit ces modèles.

Modèle de stratégie	Description
AWSIoTConfigAccess	Autorise l'identité associée à accéder à toutes les opérations de configuration AWS IoT. Cette stratégie peut affecter le traitement et le stockage des données.
AWSIoTConfigReadOnlyAccess	Autorise l'identité associée à accéder aux opérations de configuration en lecture seule.
AWSIoTDataAccess	Autorise à l'identité associée un accès complet à toutes les opérations sur les données AWS IoT. Les opérations sur les données envoient des données via les protocoles MQTT ou HTTP.
AWSIoTEventsFullAccess	Autorise l'identité associée à avoir un accès complet aux événements AWS IoT.
AWSIoTEventsReadOnlyAccess	Autorise l'identité associée à accéder aux événements AWS IoT en lecture seule.
AWSIoTFullAccess	Autorise les identités associées à avoir un accès complet à toutes les opérations de configuration et de messagerie AWS IoT.
AWSIoTLogging	Autorise l'identité associée à créer des groupes Amazon CloudWatch Logs et à diffuser des journaux vers les groupes. Cette stratégie est attachée à votre rôle de journalisation CloudWatch.
AWSIoTOTAUpdate	Permet à l'identité associée de créerAWS IoTTâches,AWS IoTtâches de signature de code, et pour décrireAWStâches de signataire de code.
AWSIoTRuleActions	Autorise l'identité associée à accéder à tousAWSServices pris en charge dansAWS IoTActions de règle.
AWSIoTThingsRegistration	Autorise l'identité associée à enregistrer des objets en bloc à l'aide de l'API StartThingRegistrationTask. Cette stratégie peut affecter le traitement et le stockage des données.
AWSIOTWirelessDataAccess	Autorise l'identité associée à envoyer des données àAWS IoTpériphériques sans fil.
awsiotWirelessFullAccess	Autorise l'identité associée à avoir un accès complet àAWS IoTSans fil.
awsiotWirelessFullPublishAccess	SubventionsAWS IoTAccès limité sans fil pour publier surAWS IoTRègles en votre nom.

Modèle de stratégie	Description
AWSIOTWirelessLogging	Autorise l'identité associée à créer des groupes de journaux Amazon CloudWatch et à diffuser des journaux vers les groupes. Cette stratégie est attachée à votre rôle de journalisation CloudWatch.
AWSIOTWirelessReadOnlyAccess	Autorise l'identité associée à accéder en lecture seule àAWS IoTSans fil.
AWSIOTWirelessGatewayCertManager	Autorise l'accès d'identité associée à créer, répertorier et décrireAWS IoTCertificats.

Stratégies basées sur l'identité AWS IoT

Avec les stratégies IAM basées sur l'identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées.AWS IoTprend en charge des actions, ressources et clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une stratégie JSON, veuillez consulter Références des éléments de stratégie JSON IAM dans le Guide de l'utilisateur IAM.

Actions

Les administrateurs peuvent utiliser les stratégies JSON AWS pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément `Action` d'une stratégie JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une stratégie. Les actions de stratégie possèdent généralement le même nom que l'opération d'API AWS associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une stratégie. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Le tableau suivant répertorie les actions IAM IoT, lesAWS IoTet la ressource manipulée par l'action.

Actions de stratégie	API AWS IoT	Ressources
iot:AcceptCertificateTransfer	AcceptCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id* Note La . Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.
iot:AddThingToThingGroup	AddThingToThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:AssociateTargetsWithJob	AssociateTargetsWithJob	Job
iot:AttachPolicy	AttachPolicy	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* ou

Actions de stratégie	API AWS IoT	Ressources
		arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:AttachPrincipalPolicy	AttachPrincipalPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:AttachSecurityProfile	AttachSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:AttachThingPrincipal	AttachThingPrincipal	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:CancelCertificateTransfer	CancelCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id* Note La . Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.
iot:CancelJob	CancelJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:CancelJobExecution	CancelJobExecution	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ClearDefaultAuthorizer	ClearDefaultAuthorizer	Aucune
iot:CreateAuthorizer	CreateAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:CreateCertificateFromCsr	CreateCertificateFromCsr	*
iot:CreateDimension	CreateDimension	arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:CreateJob	CreateJob	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name* arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
IOT : CreateJobTemplate	CreateJobTemplate	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
iot:CreateKeysAndCertificate	CreateKeysAndCertificate	*
iot:CreatePolicy	CreatePolicy	*
iot:CreatePolicyVersion	CreatePolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name* Note Il doit être unAWS IoT, pas une stratégie IAM.

Actions de stratégie	API AWS IoT	Ressources
iot:CreateRoleAlias	CreateRoleAlias	(paramètre : roleAlias) arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:CreateSecurityProfile	CreateSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:CreateThing	CreateThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:CreateThingGroup	CreateThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* pour le groupe en cours de création et pour le groupe parent, si utilisé
iot:CreateThingType	CreateThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:CreateTopicRule	CreateTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:DeleteAuthorizer	DeleteAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-name*
iot:DeleteCACertificate	DeleteCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:DeleteCertificate	DeleteCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DeleteDimension	DeleteDimension	arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:DeleteJob	DeleteJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
IOT : deleteJobTemplate	DeleteJobTemplate	arn:aws:iot :*region* :*id-compte*:job/*job-template-id*
iot:DeleteJobExecution	DeleteJobExecution	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DeletePolicy	DeletePolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:DeletePolicyVersion	DeletePolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:DeleteRegistrationCode	DeleteRegistrationCode	*
iot:DeleteRoleAlias	DeleteRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DeleteSecurityProfile	DeleteSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*

Actions de stratégie	API AWS IoT	Ressources
iot:DeleteThing	DeleteThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DeleteThingGroup	DeleteThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeleteThingType	DeleteThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DeleteTopicRule	DeleteTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeprecateThingType	DeprecateThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name* (paramètre : authorizerName) none
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:DescribeCertificate	DescribeCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Aucune
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfigurations	DescribeEventConfigurations	*
iot:DescribeIndex	DescribeIndex	arn:aws:iot:*region*:*account-id*:index/*index-name*
iot:DescribeJob	DescribeJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:DescribeJobExecution	DescribeJobExecution	Aucune
IOT : DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot :*region* :*id-compte*:job/*job-template-id*
iot:DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DescribeThing	DescribeThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DescribeThingGroup	DescribeThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DescribeThingRegistrationTask	DescribeThingRegistrationTask	Aucune
iot:DescribeThingType	DescribeThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*

Actions de stratégie	API AWS IoT	Ressources
iot:DetachPolicy	DetachPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id* ou arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DetachSecurityProfile	DetachSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DisableTopicRule	DisableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:EnableTopicRule	EnableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:GetIndexingConfiguration	GetIndexingConfiguration	Aucune
iot:GetJobDocument	GetJobDocument	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:GetLoggingOptions	GetLoggingOptions	*
iot:GetPolicy	GetPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:GetPolicyVersion	GetPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:GetRegistrationCode	GetRegistrationCode	*
iot:GetTopicRule	GetTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:ListAttachedPolicies	ListAttachedPolicies	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* ou arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListAuthorizers	ListAuthorizers	Aucune
iot:ListCACertificates	ListCACertificates	*
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Aucune
iot:ListJobExecutionsForJob	ListJobExecutionsForJob	Aucune
iot:ListJobExecutionsForThing	ListJobExecutionsForThing	Aucune

Actions de stratégie	API AWS IoT	Ressources
iot:ListJobs	ListJobs	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* si le paramètre thingGroupName est utilisé
IOT : ListJobTemplates	ListJobs	Aucune
iot:ListOutgoingCertificates	ListOutgoingCertificates	*
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListPolicyVersions	ListPolicyVersions	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListPrincipalThings	ListPrincipalThings	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListRoleAliases	ListRoleAliases	Aucune
iot:ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListThingGroups	ListThingGroups	Aucune
iot:ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingPrincipals	ListThingPrincipals	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingRegistrationTaskReports	ListThingRegistrationTaskReports	Aucune
iot:ListThingRegistrationTasks	ListThingRegistrationTasks	Aucune
iot:ListThingTypes	ListThingTypes	*
iot:ListThings	ListThings	*
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Aucune
iot:RegisterCACertificate	RegisterCACertificate	*
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Aucune
iot:RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*

Actions de stratégie	API AWS IoT	Ressources
iot:SearchIndex	SearchIndex	arn:aws:iot:*region*:*account-id*:index/*index-id*
iot:SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:SetLoggingOptions	SetLoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:SetV2LoggingLevel	SetV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:SetV2LoggingOptions	SetV2LoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:StartThingRegistrationTask	StartThingRegistrationTask	Aucune
iot:StopThingRegistrationTask	StopThingRegistrationTask	Aucune
iot:TestAuthorization	TestAuthorization	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Aucune
iot:TransferCertificate	TransferCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateAuthorizer	UpdateAuthorizer	arn:aws:iot:*region*:*account-id*:authorizerfunction/*authorizer-function-name*
iot:UpdateCACertificate	UpdateCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:UpdateCertificate	UpdateCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateDimension	UpdateDimension	arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:UpdateEventConfigurations	UpdateEventConfigurations	Aucune
iot:UpdateIndexingConfiguration	UpdateIndexingConfiguration	Aucune
iot:UpdateRoleAlias	UpdateRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:UpdateSecurityProfile	UpdateSecurityProfile	arn:aws:iot:*region*:*account-id*:securityprofile/*security-profile-name* arn:aws:iot:*region*:*account-id*:dimension/*dimension-name*
iot:UpdateThing	UpdateThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:UpdateThingGroup	UpdateThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:UpdateThingGroupsForThing	UpdateThingGroupsForThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*

Actions de stratégieAWS IoTUtilisez le préfixe suivant avant l'action : `iot:`. Par exemple, pour accorder à quelqu'un l'autorisation de répertorier tous les objets IoT enregistrés dans sa Compte AWS avec le`ListThings`, vous incluez l'API`iot:ListThings`dans leur politique. Les déclarations de stratégie

doivent inclure un élément `Action` ou `NotAction`. AWS IoT définit son propre ensemble d'actions qui décrivent les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

```
"Action": [
      "ec2:action1",
      "ec2:action2"
```

Vous pouvez aussi spécifier 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": "iot:Describe*"
```

Pour afficher une liste desAWS IoTactions, voirActions définies parAWS IoTdans leGuide de l'utilisateur IAM.

Actions Device Advisor

Le tableau suivant répertorie les actions IAM IoT, les actions associées àAWS IoTL'API Device Advisor et la ressource manipulée par l'action.

Actions de stratégie	API AWS IoT	Ressources
IotDeviceAdvisor:CreateSuiteDefinition	CreateSuiteDefinition	Aucune
IotDeviceAdvisor:DeleteSuiteDefinition	DeleteSuiteDefinition	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*
IotDeviceAdvisor:GetSuiteDefinition	GetSuiteDefinition	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*
IotDeviceAdvisor:GetSuiteRun	GetSuiteRun	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-run-id*
IotDeviceAdvisor:GetSuiteRunReport	GetSuiteRunReport	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/*suite-definition-id*/*suite-run-id*
IotDeviceAdvisor:ListSuiteDefinitions	ListSuiteDefinitions	Aucune
IotDeviceAdvisor:ListSuiteRuns	ListSuiteRuns	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*
IotDeviceAdvisor:ListTagsForResource	ListTagsForResource	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id* arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/suite-definition-id/*suite-run-id*
IotDeviceAdvisor:StartSuiteRun	StartSuiteRun	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*
IotDeviceAdvisor:TagResource	TagResource	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id* arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/suite-definition-id/*suite-run-id*

Actions de stratégie	API AWS IoT	Ressources
IotDeviceAdvisor:Untag*Resource*	UntagResource	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id* arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/suite-definition-id/*suite-run-id*
IotDeviceAdvisor:UpdateSuite	UpdateSuite Definition	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*
IotDeviceAdvisor:StopSuiteRun	StopSuiteRun	arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/suite-definition-id/*suite-run-id*

Actions de stratégieAWS IoTDevice Advisor utilise le préfixe suivant avant l'action : `iotdeviceadvisor:`. Par exemple, pour accorder à quelqu'un l'autorisation de répertorier toutes les définitions de suite enregistrées dans sa Compte AWS avec l'API ListSuiteDefinitions, vous incluez le`iotdeviceadvisor:ListSuiteDefinitions`dans leur politique.

Resources

Les administrateurs peuvent utiliser les stratégies JSON AWS pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément de stratégie JSON `Resource` indique le ou les objets pour lesquels l'action s'applique. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de définir une ressource à l'aide de son Amazon Resource Name (ARN). Vous pouvez le faire pour les actions qui sont compatibles avec un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

AWS IoTRessources

Actions de stratégie	API AWS IoT	Ressources
iot:AcceptCertificateTransfer	AcceptCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id* **Note** La . Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.
iot:AddThingToThingGroup	AddThingToThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:AssociateTargetsWithJob	AssociateTargetsWithJob	Aucune
iot:AttachPolicy	AttachPolicy	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*

Actions de stratégie	API AWS IoT	Ressources
		ou arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:AttachPrincipalPolicy	AttachPrincipalPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:AttachThingPrincipal	AttachThingPrincipal	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:CancelCertificateTransfer	CancelCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id* **Note** La . Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.
iot:CancelJob	CancelJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:CancelJobExecution	CancelJobExecution	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ClearDefaultAuthorizer	ClearDefaultAuthorizer	Aucune
iot:CreateAuthorizer	CreateAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:CreateCertificateFromCsr	CreateCertificateFromCsr	*
iot:CreateJob	CreateJob	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name* arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
IOT : CreateJobTemplate	CreateJobTemplate	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
iot:CreateKeysAndCertificate	CreateKeysAndCertificate	*
iot:CreatePolicy	CreatePolicy	*
CreatePolicyVersion	iot:CreatePolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name* **Note** Il doit être unAWS IoT, pas une stratégie IAM.
iot:CreateRoleAlias	CreateRoleAlias	(paramètre : roleAlias) arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:CreateThing	CreateThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*

Actions de stratégie	API AWS IoT	Ressources
iot:CreateThingGroup	CreateThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* pour le groupe en cours de création et pour le groupe parent, si utilisé
iot:CreateThingType	CreateThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:CreateTopicRule	CreateTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:DeleteAuthorizer	DeleteAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-name*
iot:DeleteCACertificate	DeleteCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:DeleteCertificate	DeleteCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DeleteJob	DeleteJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:DeleteJobExecution	DeleteJobExecution	arn:aws:iot:*region*:*account-id*:job/*job-id* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
IOT : deleteJobTemplate	DeleteJobTemplate	arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
iot:DeletePolicy	DeletePolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:DeletePolicyVersion	DeletePolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:DeleteRegistrationCode	DeleteRegistrationCode	*
iot:DeleteRoleAlias	DeleteRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DeleteThing	DeleteThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DeleteThingGroup	DeleteThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeleteThingType	DeleteThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DeleteTopicRule	DeleteTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:DeleteV2LoggingLevel	DeleteV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DeprecateThingType	DeprecateThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name* (paramètre : authorizerName) none

Actions de stratégie	API AWS IoT	Ressources
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:DescribeCertificate	DescribeCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Aucune
iot:DescribeEndpoint	DescribeEndpoint	*
iot:DescribeEventConfigurations	DescribeEventConfigurations	Aucune
iot:DescribeIndex	DescribeIndex	arn:aws:iot:*region*:*account-id*:index/*index-name*
iot:DescribeJob	DescribeJob	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:DescribeJobExecution	DescribeJobExecution	Aucune
IOT : DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot :*region* :*id-compte*:jobtemplate/*job-template-id*
iot:DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:DescribeThing	DescribeThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:DescribeThingGroup	DescribeThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DescribeThingRegistrationTask	DescribeThingRegistrationTask	Aucune
iot:DescribeThingType	DescribeThingType	arn:aws:iot:*region*:*account-id*:thingtype/*thing-type-name*
iot:DetachPolicy	DetachPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id* ou arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:DisableTopicRule	DisableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:EnableTopicRule	EnableTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:GetIndexingConfiguration	GetIndexingConfiguration	Aucune
iot:GetJobDocument	GetJobDocument	arn:aws:iot:*region*:*account-id*:job/*job-id*
iot:GetLoggingOptions	GetLoggingOptions	*
iot:GetPolicy	GetPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:GetPolicyVersion	GetPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*

Actions de stratégie	API AWS IoT	Ressources
iot:GetRegistrationCode	GetRegistrationCode	*
iot:GetTopicRule	GetTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:ListAttachedPolicies	ListAttachedPolicies	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* ou arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListAuthorizers	ListAuthorizers	Aucune
iot:ListCACertificates	ListCACertificates	*
iot:ListCertificates	ListCertificates	*
iot:ListCertificatesByCA	ListCertificatesByCA	*
iot:ListIndices	ListIndices	Aucune
iot:ListJobExecutionsForJob	ListJobExecutionsForJob	Aucune
iot:ListJobExecutionsForThing	ListJobExecutionsForThing	Aucune
iot:ListJobs	ListJobs	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* si le paramètre thingGroupName est utilisé
IOT : ListJobTemplates	ListJobTemplates	Aucune
iot:ListOutgoingCertificates	ListOutgoingCertificates	*
iot:ListPolicies	ListPolicies	*
iot:ListPolicyPrincipals	ListPolicyPrincipals	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListPolicyVersions	ListPolicyVersions	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListPrincipalThings	ListPrincipalThings	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:ListRoleAliases	ListRoleAliases	Aucune
iot:ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:ListThingGroups	ListThingGroups	Aucune
iot:ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingPrincipals	ListThingPrincipals	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ListThingRegistrationTaskReports	ListThingRegistrationTaskReports	Aucune
iot:ListThingRegistrationTasks	ListThingRegistrationTasks	Aucune
iot:ListThingTypes	ListThingTypes	*

Actions de stratégie	API AWS IoT	Ressources
iot:ListThings	ListThings	*
iot:ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:ListTopicRules	ListTopicRules	*
iot:ListV2LoggingLevels	ListV2LoggingLevels	Aucune
iot:RegisterCACertificate	RegisterCACertificate	*
iot:RegisterCertificate	RegisterCertificate	*
iot:RegisterThing	RegisterThing	Aucune
iot:RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name* arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot:*region*:*account-id*:rule/*rule-name*
iot:SearchIndex	SearchIndex	arn:aws:iot:*region*:*account-id*:index/*index-id*
iot:SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot:*region*:*account-id*:authorizer/*authorizer-function-name*
iot:SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot:*region*:*account-id*:policy/*policy-name*
iot:SetLoggingOptions	SetLoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:SetV2LoggingLevel	SetV2LoggingLevel	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:SetV2LoggingOptions	SetV2LoggingOptions	arn:aws:iot:*region*:*account-id*:role/*role-name*
iot:StartThingRegistrationTask	StartThingRegistrationTask	Aucune
iot:StopThingRegistrationTask	StopThingRegistrationTask	Aucune
iot:TestAuthorization	TestAuthorization	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:TestInvokeAuthorizer	TestInvokeAuthorizer	Aucune
iot:TransferCertificate	TransferCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateAuthorizer	UpdateAuthorizer	arn:aws:iot:*region*:*account-id*:authorizerfunction/*authorizer-function-name*
iot:UpdateCACertificate	UpdateCACertificate	arn:aws:iot:*region*:*account-id*:cacert/*cert-id*
iot:UpdateCertificate	UpdateCertificate	arn:aws:iot:*region*:*account-id*:cert/*cert-id*
iot:UpdateEventConfigurations	UpdateEventConfigurations	Aucune
iot:UpdateIndexingConfiguration	UpdateIndexingConfiguration	Aucune

Actions de stratégie	API AWS IoT	Ressources
iot:UpdateRoleAlias	UpdateRoleAlias	arn:aws:iot:*region*:*account-id*:rolealias/*role-alias-name*
iot:UpdateThing	UpdateThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*
iot:UpdateThingGroup	UpdateThingGroup	arn:aws:iot:*region*:*account-id*:thinggroup/*thing-group-name*
iot:UpdateThingGroupsForThing	UpdateThingGroupsForThing	arn:aws:iot:*region*:*account-id*:thing/*thing-name*

Pour de plus amples informations sur le format des ARN, veuillez consulter Noms ARN (Amazon Resource Name) et espaces de noms du service AWS.

Certaines actions AWS IoT, telles que la création de ressources, ne peuvent pas être exécutées sur une ressource précise. Dans ce cas, vous devez utiliser le caractère générique (*).

```
"Resource": "*"
```

Pour afficher une liste desAWS IoTLes types de ressources et leurs ARN, consultezRessources définies parAWS IoTdans leGuide de l'utilisateur IAM. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez Actions définies par AWS IoT.

Ressources Device Advisor

Pour définir des restrictions au niveau des ressources pourAWS IoTStratégies IAM Device Advisor, utilisez les formats ARN de ressource suivants pour les définitions de suite et les exécutions de suite.

Format ARN des ressources de définition de suite

arn:aws:iotdeviceadvisor :*region* :*id-compte*:suitedefinition/*suite-definition-id*

Format ARN des ressources d'exécution suite

arn:aws:iotdeviceadvisor :*region* :*id-compte*:suiterun/*suite-definition-id*/*suite-run-id*

Clés de condition

Les administrateurs peuvent utiliser les stratégies JSON AWS pour spécifier qui a accès à quoi. Cela signifie : quel mandataire peut effectuer des actions sur quel type de ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition tels que les signes égal ou inférieur à, pour faire correspondre la condition de la 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 IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour de plus amples d'informations, veuillez consulter Éléments d'une stratégie IAM : variables et balises dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques à un service. Pour afficher toutes les clés de condition globales AWS, consultez Clés de contexte de condition globale AWS dans le Guide de l'utilisateur IAM.

AWS IoT définit son propre ensemble de clés de condition et prend également en charge l'utilisation des clés de condition globales. Pour afficher toutes les clés de condition globales AWS, consultez Clés de contexte de condition globale AWS dans le Guide de l'utilisateur IAM.

Clés de condition AWS IoT

Clés de condition AWS IoT	Description	Type
`aws:RequestTag/${tag-key}`	Clé de balise présente dans la demande envoyée par l'utilisateur à AWS IoT.	Chaîne
`aws:ResourceTag/${tag-key}`	Composant de clé de balise d'une balise attachée à une ressource AWS IoT.	Chaîne
`aws:TagKeys`	Liste de tous les noms de clés de balise associés à la ressource de la demande.	Chaîne

Pour afficher une liste desAWS IoTClé de condition, consultezClés de condition pourAWS IoTdans leGuide de l'utilisateur IAM. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez Actions définies par AWS IoT.

Examples

Pour voir des exemples de stratégies AWS IoT basées sur l'identité, consultez Exemples de stratégies basées sur l'identité AWS IoT (p. 338).

Stratégies basées sur les ressources AWS IoT

Les stratégies basées sur les ressources sont des documents de stratégie JSON précisant les actions qu'un mandataire indiqué peut effectuer sur la ressource AWS IoT et dans quelles conditions.

AWS IoTne prend pas en charge les stratégies basées sur les ressource IAM. Il prend toutefois en chargeAWS IoTStratégies basées sur les ressources.

Autorisation basée sur les balises AWS IoT

Vous pouvez attacher des balises aux ressources de AWS IoT, ou transmettre des balises dans une demande à AWS IoT. Pour contrôler l'accès basé sur des balises, vous devez fournir les informations des balises dans l'élément de condition d'une stratégie en utilisant les clés de condition `iot:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Pour plus d'informations, consultez Utilisation des balises avec des stratégies IAM (p. 227). Pour plus d'informations sur le balisage des ressources AWS IoT, consultez Balisage de vos ressources AWS IoT (p. 226).

Pour afficher un exemple de stratégie basée sur l'identité permettant de limiter l'accès à une ressource basée sur les balises de cette ressource, veuillez consulter Affichage des ressources en fonction des balises AWS IoT (p. 340).

AWS IoTRôles IAM

UnRôle IAMest une entité au sein de votre Compte AWS qui dispose d'autorisations spécifiques.

Utilisation des informations d'identification temporaires avec AWS IoT

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 entre comptes. Vous obtenez des informations d'identification de sécurité temporaires en appelant des opérations d'API AWS STS comme AssumeRole ou GetFederationToken.

AWS IoT prend en charge l'utilisation des informations d'identification temporaires.

Rôles liés à un service

Les rôles liés à un service permettent aux services AWS d'accéder à des ressources dans 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 ne prend pas en charge les rôles liés à un service.

Rôles de service

Cette fonctionnalité permet à un service d'endosser un rôle de 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.

Exemples de stratégies basées sur l'identité AWS IoT

Par défaut, les utilisateurs et rôles IAM ne sont pas autorisés à créer ou modifierAWS IoTAWS. Ils ne peuvent pas non plus exécuter des tâches à l'aide de AWS Management Console, AWS CLI ou de l'API AWS. Un administrateur IAM doit créer des stratégies 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 IAM ayant besoin de ces autorisations.

Pour savoir comment créer une stratégie IAM basée sur l'identité à l'aide de ces exemples de documents de stratégie JSON, veuillez consulter Création de stratégies dans l'onglet JSON dans le Guide de l'utilisateur IAM.

Rubriques

Bonnes pratiques en matière de stratégies

Les stratégies basées sur l'identité sont très puissantes. Elles déterminent si une personne peut créer, consulter ou supprimer des ressources AWS IoT dans votre compte. Ces actions peuvent entraîner des

frais pour votre compte AWS. Lorsque vous créez ou modifiez des stratégies basées sur l'identité, suivez ces instructions et recommandations :

- Commencez à utiliserAWSStratégies gérées— Pour commencer à utiliserAWS IoTRapidement, utilisezAWSLes stratégies gérées pour accorder à vos employés les autorisations dont ils ont besoin. Ces stratégies sont déjà disponibles dans votre compte et sont gérées et mises à jour par AWS. Pour de plus amples informations, veuillez consulter Démarrez avec les autorisations à l'aide des stratégies gérées AWS dans le Guide de l'utilisateur IAM.
- Accorder le privilège le plus faible : Lorsque vous créez des stratégies personnalisées, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Commencez avec un minimum d'autorisations 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 de plus amples informations, veuillez consulter Accorder les privilèges les plus faibles possible dans le Guide de l'utilisateur IAM.
- Activer la MFA pour les opérations confidentielles : Pour plus de sécurité, demandez aux utilisateurs IAM d'utiliser la Multi-Factor Authentication (MFA) pour accéder à des ressources ou à des opérations d'API confidentielles. Pour de plus amples informations, veuillez consulter Utilisation de la Multi-Factor Authentication (MFA) dans AWS dans le Guide de l'utilisateur IAM.
- Utiliser des conditions de stratégie pour davantage de sécurité : Définissez les conditions dans lesquelles vos stratégies basées sur l'identité autorisent l'accès à une ressource, dans la mesure où cela reste pratique. Par exemple, vous pouvez rédiger les conditions pour spécifier une plage d'adresses IP autorisées d'où peut provenir une demande. Vous pouvez également écrire des conditions pour autoriser les requêtes uniquement à une date ou dans une plage de temps spécifiée, ou pour imposer l'utilisation de SSL ou de MFA. Pour de plus amples informations, veuillez consulterÉléments de stratégie JSON IAM : Conditiondans leGuide de l'utilisateur IAM.

Utilisation de la console AWS IoT

Pour accéder à la console AWS IoT, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et consulter les informations relatives à l'objetAWS IoTdans votre Compte AWS . Si vous créez une stratégie basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs et rôles IAM) tributaires de cette stratégie.

Pour garantir que ces entités pourront continuer à utiliser la console AWS IoT, attachez également la stratégie gérée AWS suivante aux entités : `AWSIoTFullAccess`. Pour de plus amples informations, veuillez consulterAjout d'autorisations à un utilisateurdans leGuide de l'utilisateur IAM.

Vous n'avez pas besoin d'accorder les autorisations minimales de console pour les utilisateurs qui effectuent des appels uniquement à l'interface AWS CLI ou API AWS. 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 IAM d'afficher les stratégies en ligne et gérées attachées à leur identité d'utilisateur. Cette stratégie inclut les autorisations nécessaires pour réaliser cette action sur la console ou par programmation à l'aide de l'AWS CLI ou de l'API AWS.

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

```
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Affichage des ressources en fonction des balises AWS IoT

Vous pouvez utiliser des conditions dans votre stratégie basée sur l'identité pour contrôler l'accès aux ressources AWS IoT en fonction des balises. Cet exemple montre comment créer une stratégie qui autorise l'affichage d'un objet. Toutefois, l'autorisation est accordée uniquement si la balise `Owner` de l'objet a pour valeur le nom d'utilisateur de cet utilisateur. Cette stratégie accorde également les autorisations nécessaires pour réaliser cette action sur la console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ListBillingGroupsInConsole",
            "Effect": "Allow",
            "Action": "iot:ListBillingGroups",
            "Resource": "*"
        },
        {
            "Sid": "ViewBillingGroupsIfOwner",
            "Effect": "Allow",
           "Action": "iot:DescribeBillingGroup",
            "Resource": "arn:aws:iot:*:*:billinggroup/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Vous pouvez rattacher cette stratégie aux utilisateurs IAM de votre compte. Si un utilisateur nommé `richard-roe` tente d'afficher un groupe de facturation AWS IoT, le groupe de facturation doit être balisé avec `Owner=richard-roe` ou `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 de plus amples informations, veuillez consulterÉléments de stratégie JSON IAM : Conditiondans leGuide de l'utilisateur IAM.

Affichage d'unAWS IoTRessources Device Advisor en fonction des balises

Vous pouvez utiliser des conditions dans votre stratégie basée sur l'identité pour contrôler l'accès àAWS IoTRessources Device Advisor basées sur des balises. L'exemple suivant montre comment créer une stratégie qui permet d'afficher une définition de suite particulière. Toutefois, l'autorisation est accordée uniquement si la balise de définition de suite a`SuiteType`définie sur la valeur de`MQTT`. Cette stratégie accorde également les autorisations nécessaires pour réaliser cette action sur la console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewSuiteDefinition",
            "Effect": "Allow",
            "Action": "iotdeviceadvisor:GetSuiteDefinition",
            "Resource": "arn:aws:iotdeviceadvisor:*:*:suitedefinition/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/SuiteType": "MQTT"}
            }
        }
    ]
}
```

Résolution des problèmes liés à Identity and Access AWS IoT

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avecAWS IoTet IAM.

Rubriques

Je ne suis pas autorisé à effectuer une action dans AWS IoT

Si AWS Management Console indique que vous n'êtes pas autorisé à exécuter 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 le paramètre`mateojackson`L'utilisateur IAM tente d'utiliser la console pour afficher des informations détaillées sur un objet mais ne dispose pas`iot:`*`DescribeThing`*Autorisations.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iot:DescribeThing
            on resource: MyIoTThing
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à la ressource *`MyIoTThing`* à l'aide de l'action `iot:`*`DescribeThing`*.

UtiliserAWS IoTDevice Advisor

Si vous utilisezAWS IoTDevice Advisor, l'exemple d'erreur suivant se produit lorsque`mateojackson`L'utilisateur IAM tente d'utiliser la console pour afficher des informations détaillées concernant une définition de suite, mais ne dispose pas de`iotdeviceadvisor:GetSuiteDefinition`Autorisations.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotdeviceadvisor:GetSuiteDefinition
            on resource: MySuiteDefinition
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'accéder à la*MySuiteDefinition*à l'aide de l'`iotdeviceadvisor`:*GetSuiteDefinition*action.

Je ne suis pas autorisé à exécuter : iam:PassRole

Si vous recevez un message d'erreur selon lequel vous n'êtes pas autorisé à exécuter l'action `iam:PassRole`, 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. Demandez à cette personne de mettre à jour vos stratégies pour vous permettre de transmettre un rôle à AWS IoT.

Certains services AWS vous permettent de transmettre un rôle existant à ce service, au lieu de créer un nouveau rôle de service ou rôle lié à un service. Pour ce faire, un utilisateur doit disposer des autorisations nécessaires pour transmettre le rôle au service.

L'exemple d'erreur suivant se produit lorsqu'un utilisateur IAM nommé `marymajor` essaie d'utiliser la console pour exécuter une action dans AWS IoT. Toutefois, l'action nécessite que le service ait des autorisations accordées par un rôle de service. Mary ne dispose pas des autorisations nécessaires pour transférer le rôle au service.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform: iam:PassRole
```

Dans ce cas, Mary demande à son administrateur de mettre à jour ses stratégies pour lui permettre d'exécuter l'action `iam:PassRole`.

Je veux afficher mes clés d'accès

Une fois les clés d'accès utilisateur IAM créées, vous pouvez afficher votre ID de clé d'accès à tout moment. Toutefois, vous ne pouvez pas revoir votre clé d'accès secrète. Si vous perdez votre clé d'accès secrète, vous devez créer une nouvelle paire de clés.

Les clés d'accès se composent de deux parties : un ID de clé d'accès (par exemple, `AKIAIOSFODNN7EXAMPLE`) et une clé d'accès secrète (par exemple, `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`). À l'instar d'un nom d'utilisateur et un mot de passe, vous devez utiliser à la fois l'ID de clé d'accès et la clé d'accès secrète pour authentifier vos demandes. Gérez vos clés d'accès de manière aussi sécurisée que votre nom d'utilisateur et votre mot de passe.

Important

Ne communiquez pas vos clés d'accès à un tiers, même pour qu'il vous aide à trouver votre ID utilisateur canonique. En effet, vous lui accorderiez ainsi un accès permanent à votre compte.

Lorsque vous créez une paire de clé d'accès, enregistrez l'ID de clé d'accès et la clé d'accès secrète dans un emplacement sécurisé. La clé d'accès secrète est accessible uniquement au moment de sa création. Si vous perdez votre clé d'accès secrète, vous devez ajouter de nouvelles clés d'accès pour votre utilisateur IAM. Vous pouvez avoir un maximum de deux clés d'accès. Si vous en avez déjà deux, vous

devez supprimer une paire de clés avant d'en créer une nouvelle. Pour afficher les instructions, veuillez consulter Gestion des clés d'accès dans le Guide de l'utilisateur IAM.

Je suis un administrateur et je veux autoriser d'autres utilisateurs à accéder à AWS IoT

Pour autoriser d'autres personnes à accéderAWS IoT, vous devez créer une entité IAM (utilisateur ou rôle) pour la personne ou l'application nécessitant un accès. Ils utiliseront les informations d'identification de cette entité pour accéder à AWS. Vous devez ensuite associer une stratégie à l'entité qui leur accorde les autorisations appropriées dans AWS IoT.

Pour démarrer immédiatement, veuillez consulter Création de votre premier groupe et utilisateur délégué IAM dans le Guide de l'utilisateur IAM.

Je veux permettre à des personnes extérieures à mon Compte AWS pour accéder à monAWS IoTresources

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation peuvent utiliser pour accéder à vos ressources. Vous pouvez spécifier la personne à qui vous souhaitez confier le rôle. Pour les services qui prennent en charge les stratégies basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces stratégies pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si AWS IoT prend en charge ces fonctionnalités, consultez Fonctionnement d'AWS IoT avec IAM (p. 320).
- Pour savoir comment fournir un accès à vos ressources sur les comptes AWS que vous détenez, consultez Octroi de l'accès à un utilisateur IAM dans un autre compte AWS vous appartenant dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des comptes AWS tiers, veuillez consulter Octroi de l'accès à des comptes AWS appartenant à des tiers dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, veuillez consulter 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 stratégies basées sur les ressources pour l'accès comptes multiples, veuillez consulter Différence entre les rôles IAM et les stratégies basées sur les ressources dans le Guide de l'utilisateur IAM.

Journalisation et surveillance

La surveillance est un enjeu important pour assurer la fiabilité, la disponibilité et les performances d'AWS IoT et de vos solutions AWS. Vous devez collecter des données de surveillance à partir de toutes les parties de votreAWSSolution de telle sorte que vous puissiez déboguer plus facilement une éventuelle défaillance à plusieurs points. Pour de plus amples informations sur les procédures de journalisation et de surveillance, veuillez consulter Surveillance de AWS IoT (p. 352)

Outils de supervision

AWS fournit des outils que vous pouvez utiliser pour surveiller AWS IoT. 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 pour surveiller AWS IoT et signaler en cas de problème :

- Alarmes Amazon CloudWatch : surveillez une seule métrique sur une durée définie et exécutez une ou plusieurs actions en fonction de la valeur de la métrique par rapport à un seuil donné sur un certain nombre de périodes. L'action est une notification envoyée à une rubrique Amazon Simple Notification Service (Amazon SNS) ou une politique Amazon EC2 Auto Scaling. Les alarmes CloudWatch n'appellent pas une action uniquement parce qu'elles se trouvent dans un état particulier. L'état doit avoir changé et avoir été maintenu pendant un nombre de périodes spécifié. Pour plus d'informations, consultez ContrôleAWS IoTalarmes et mesures à l'aide d'Amazon CloudWatch (p. 359).
- Amazon CloudWatch Logs – Surveillez, stockez et accédez à vos fichiers journaux depuis AWS CloudTrail ou d'autres sources. Amazon CloudWatch Logs vous permet également de voir les étapes critiquesAWS IoTLes cas de test Device Advisor prennent, générés des événements et des messages MQTT envoyés à partir de vos appareils ou AWS IoT Core pendant l'exécution du test. Ces journaux permettent de déboguer et de prendre des mesures correctives sur vos appareils. Pour de plus amples informations, veuillez consulterContrôleAWS IoTUtilisation CloudWatch Logs (p. 373)Pour plus d'informations sur l'utilisation d'Amazon CloudWatch, consultez.Surveillance des fichiers journauxdans leGuide de l'utilisateur Amazon CloudWatch..
- Amazon CloudWatch Events : mettez en correspondance les événements et transmettez-les à un ou plusieurs flux ou fonctions cibles pour apporter des modifications, capturer les informations d'état et prendre des mesures correctives. Pour de plus amples informations, veuillez consulterQu'est-ce qu'Amazon CloudWatch Eventsdans leGuide de l'utilisateur Amazon CloudWatch..
- Surveillance de journaux AWS CloudTrail – Partagez les fichiers journaux entre comptes, surveillez les fichiers journaux CloudTrail en temps réel en les envoyant à CloudWatch Logs, écrivez des applications de traitement de journaux en langage Java et assurez-vous que vos fichiers journaux n'ont pas changé après leur livraison par CloudTrail. Pour de plus amples informations, veuillez consulterJournalisez les appels d'API AWS IoT via AWS CloudTrail. (p. 391)et aussiUtilisation de fichiers journaux CloudTraildans leAWS CloudTrailGuide de l'utilisateur.

Outils de surveillance manuelle

Un autre élément important de la surveillanceAWS IoTimplique de surveiller manuellement les éléments que les alarmes CloudWatch ne couvrent pas. La .AWS IoT, CloudWatch et d'autresAWSLes tableaux de bord des consoles fournissent un aperçu de l'état de votreAWSEnvironment. Nous recommandons de consulter également les fichiers journaux sur AWS IoT.

- Le tableau de bord AWS IoT affiche :
  - Certificats CA
  - Certificates
  - Stratégies
  - Rules
  - Objets
- La page d'accueil CloudWatch présente :
  - Alarmes et statuts en cours.
  - Graphiques des alarmes et des ressources.
  - Statut d'intégrité du service.

  Vous pouvez utiliser CloudWatch pour effectuer les opérations suivantes :
  - Créer des tableaux de bord personnalisés pour surveiller les services de votre choix
  - Données de métriques de graphiques pour résoudre les problèmes et découvrir les tendances.

- Rechercher et parcourir toutes les métriques des ressources AWS
- Créer et modifier des alarmes pour être informé des problèmes.

Validation de la conformité pourAWS IoTCore

Les auditeurs tiers évaluent la sécurité et la conformité des services AWS dans le cadre de plusieurs programmes de conformité AWS, tels que SOC, PCI, FedRAMP, et HIPAA.

Pour savoir siAWS IoTou autreAWSLes services entrent dans le champ d'application de certains programmes de conformité, consultez.AWSServices visés par le programme de conformité. Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour de plus amples informations, veuillez consulter Téléchargement de rapports dans AWS Artifact.

Votre responsabilité en matière de conformité lorsque vous utilisez des services AWS est déterminée par la sensibilité de vos données, des objectifs de conformité de votre entreprise, ainsi que de la législation et de la réglementation en vigueur. AWS fournit les ressources suivantes pour faciliter le respect de la conformité :

- Guides Quick Start de la sécurité et de la conformité : ces guides de déploiement proposent des considérations architecturales et fournissent des étapes pour déployer des environnements de référence centrés sur la sécurité et la conformité sur AWS.
- Livre blanc sur l'architecture pour la sécurité et la conformité HIPAA : le livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications conformes à HIPAA.

  Note

  Tous les services ne sont pas conformes à HIPAA.
- AWS Ressources de conformité : cet ensemble de manuels et de guides peut s'appliquer à votre secteur et à votre emplacement.
- Évaluation des ressources à l'aide de règles dans leAWS Config Guide du développeur : le service AWS Config évalue dans quelle mesure vos configurations de ressources sont conformes aux pratiques internes, aux directives sectorielles et aux réglementations.
- AWS Security Hub : ce service AWS fournit une vue complète de votre état de sécurité au sein d'AWS qui vous permet de vérifier votre conformité aux normes du secteur et aux bonnes pratiques de sécurité.
- AWS Audit Manager : ce service AWS vous aide à auditer en continu votre utilisation d'AWS pour simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dansAWSIoT Core

La .AWSL'infrastructure mondiale repose sur Région AWS s et zones de disponibilité. Région AWS fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone de disponibilité à l'autre sans interruption. Les zones de disponibilité sont plus hautement disponibles, tolérantes aux pannes et évolutives que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur Région AWS s et les zones de disponibilité, consultez.AWSInfrastructure globale.

AWS IoT Core stocke des informations sur vos appareils dans le registre des appareils. Il stocke également les certificats d'autorité de certification, les certificats d'appareil et les données de shadow d'appareil. En

cas de défaillance du matériel ou du réseau, ces données sont automatiquement répliquées entre les zones de disponibilité, mais pas entre les régions.

AWS IoT Core publie des événements MQTT lorsque le registre des appareils est mis à jour. Vous pouvez utiliser ces messages pour sauvegarder vos données de registre et les enregistrer quelque part, par exemple dans une table DynamoDB. Vous êtes responsable de l'enregistrement des certificats qu' AWS IoT Core crée pour vous ou que vous créez vous-même. Le shadow d'appareil stocke les données d'état relatives à vos appareils et peut être renvoyé lorsqu'un appareil revient en ligne.AWS IoT Device Advisor stocke des informations sur la configuration de votre suite de tests. Ces données sont automatiquement répliquées en cas de défaillance du matériel ou du réseau.

AWS IoT Core Les ressources sont propres à la région et ne sont pas répliquées entre Régions AWS sauf si vous le faites expressément.

Pour plus d'informations sur les bonnes pratiques de sécurité, consultezBonnes pratiques de sécurité dans AWS IoT Core (p. 347).

Sécurité de l'infrastructure dans AWS IoT

En tant que collection de services gérés,AWS IoTest protégé par leAWSLes procédures de sécurité du réseau mondial qui sont décrites dans leAmazon Web Services : Présentation des processus de sécuritéLivre blanc.

Vous utilisez les appels d'API publiés par AWS pour accéder à AWS IoT via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE) La plupart des systèmes modernes telles que Java 7 et versions ultérieures prennent en charge ces modes. Pour plus d'informations, consultez Sécurité du transport dans AWS IoT (p. 313).

Les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un mandataire IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires afin de signer les demandes.

Analyse et gestion des vulnérabilités dansAWS IoTCore

Les parcs IoT sont composés d'un grand nombre d'appareils disposant de capacités diverses, d'une durée de vie longue et qui sont répartis géographiquement. Ces caractéristiques peuvent rendre la configuration des parcs complexe et source d'erreurs. Les appareils étant souvent limités en puissance de calcul, de mémoire et de capacités de stockage, l'utilisation du chiffrement et d'autres formes de sécurité sur les appareils eux-mêmes s'en trouve limitée. En outre, les appareils utilisent souvent des logiciels aux vulnérabilités connues. Ces facteurs font des parcs IoT une cible attractive pour les pirates informatiques et rend difficile la sécurisation de votre parc sur une base permanente.

AWS IoT Device Defender résout ces problèmes en fournissant des outils pour identifier les problèmes de sécurité, ainsi que les écarts par rapport aux bonnes pratiques. Vous pouvez utiliserAWS IoTDevice Defender pour analyser, auditer et surveiller les appareils connectés afin de détecter les comportements anormaux et d'atténuer les risques de sécurité.AWS IoT Device Defender peut auditer les parcs d'appareils afin de s'assurer qu'ils respectent les bonnes pratiques en matière de sécurité et de détecter les comportements anormaux sur les appareils. Cela vous permet d'appliquer des stratégies de sécurité cohérentes dans l'ensemble de votre parc d'appareils AWS IoT et de réagir rapidement lorsque des appareils sont menacés. Pour plus d'informations, consultez AWS IoT Device Defender (p. 790).

AWS IoTDevice Advisor envoie les mises à jour et les correctifs selon les besoins.AWS IoT Device Advisor met automatiquement à jour les cas de test Les cas de test que vous sélectionnez sont toujours avec la dernière version.

Bonnes pratiques de sécurité dans AWS IoT Core

Cette section présente des informations sur les bonnes pratiques en matière de sécurité pour AWS IoT Core . Pour plus d'informations, consultez l'article de blog relatif aux dix règles d'or de la sécurité pour les solutions IoT.

Protection des connexions MQTT dans AWS IoT

AWS IoT Core est un service basé sur le cloud géré qui permet aux appareils connectés d'interagir avec d'autres appareils et des applications cloud facilement et en toute sécurité. AWS IoT Core prend en charge HTTP, WebSocket et MQTT. Ce dernier est un protocole de communication léger spécialement conçu pour gérer les connexions intermittentes. Si vous vous connectez àAWS IoTÀ l'aide de MQTT, chaque connexion doit être associé à un identifiant connu comme une ID client. Les ID de client MQTT identifient uniquement des connexions MQTT. Si une nouvelle connexion établie à l'aide d'un ID client qui est déjà déclarée pour une autre connexion, l'agent de message AWS IoT supprime l'ancienne connexion pour autoriser la nouvelle. Les ID client doivent être uniques dans chaque Compte AWS et chaque Région AWS . Cela signifie que vous n'avez pas besoin d'appliquer l'unicité globale des identifiants clients en dehors de votre Compte AWS ou à travers les régions au sein de votre Compte AWS .

L'impact et la gravité de l'abandon des connexions MQTT sur votre parc d'appareils dépend de nombreux facteurs. Il s'agit des licences suivantes :

- Votre cas d'utilisation (par exemple, les données que vos appareils envoient à AWS IoT, la quantité de données et la fréquence d'envoi des données).
- La configuration de votre client MQTT (par exemple, les paramètres de reconnexion automatique, les temporisations de back-off associées et l'utilisation de sessions persistantes MQTT (p. 83)).
- Contraintes liées aux ressources de l'appareil.
- La cause première des déconnexions, leur agressivité et leur persistance.

Pour éviter les conflits d'ID client et leurs impacts négatifs potentiels, assurez-vous que chaque appareil ou application mobile est doté d'unAWS IoTou une stratégie IAM qui limite les ID client pouvant être utilisés pour les connexions MQTT auAWS IoTAgent de messages. Par exemple, vous pouvez utiliser une stratégie IAM pour empêcher un appareil de fermer involontairement la connexion d'un autre appareil à l'aide d'un ID client déjà utilisé. Pour plus d'informations, consultez Authorization (p. 268).

Tous les appareils de votre parc doivent avoir des identifiants avec des privilèges qui autorisent uniquement les actions prévues, y compris, mais sans s'y limiter, les actions MQTT AWS IoT telles que la publication de messages ou l'abonnement à des rubriques ayant une portée et un contexte spécifiques. Les stratégies d'autorisation spécifiques peuvent varier selon vos cas d'utilisation. Identifiez les stratégies d'autorisation qui répondent le mieux aux exigences de votre entreprise et de sécurité.

Pour simplifier la création et la gestion des stratégies d'autorisation, vous pouvez utiliserVariables de stratégie AWS IoT Core (p. 273)andVariables de stratégie IAM. Les variables de la stratégie peuvent être placées dans une stratégie et lorsque celle-ci est évaluée, les variables sont remplacées par les valeurs provenant de la demande de l'appareil. Avec des variables de stratégie, vous pouvez créer une stratégie unique pour l'octroi d'autorisations à plusieurs appareils. Vous pouvez identifier les variables de stratégie pertinentes pour votre cas d'utilisation en fonction de la configuration de votre compte AWS IoT, du mécanisme d'authentification et du protocole réseau utilisé dans la connexion à l'agent de message AWS IoT. Toutefois, pour écrire les meilleures stratégies d'autorisation, vous devez prendre en compte les spécificités de votre cas d'utilisation et votre modèle de menaces.

Par exemple, si vous avez enregistré vos appareils dans leAWS IoT, vous pouvez utiliserVariables de stratégie (p. 274)inAWS IoTLes stratégies pour accorder ou refuser des autorisations en fonction de propriétés d'objet telles que les noms d'objet, les types d'objet et les valeurs d'attribut d'objet. Le nom d'objet est obtenu à partir de l'ID client dans le message MQTT Connect envoyé lorsqu'un objet se connecte à AWS IoT. Les variables de stratégie chose sont remplacées lorsqu'une chose se connecte àAWS IoTvia MQTT à l'aide de l'authentification mutuelle TLS ou de MQTT via le protocole WebSocket à l'aide d'authentifiésIdentity Amazon Cognito. Vous pouvez utiliser la stratégieAttachThingPrincipalAPI pour attacher des certificats et des identités Amazon Cognito authentifiées à un objet.`iot:Connection.Thing.ThingName`est une variable de stratégie utile pour appliquer des restrictions à l'ID client. L'exemple de stratégie AWS IoT suivant exige que le nom d'un objet enregistré soit utilisé comme ID client pour les connexions MQTT vers l'agent de message AWS IoT :

```
{
    "Version":"2012-10-17",
    "Statement":[
    {
        "Effect":"Allow",
        "Action":"iot:Connect",
        "Resource":[
            "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ]
    }
    ]
}
```

Si vous voulez identifier les conflits d'ID client en cours, vous pouvez activer et utiliserCloudWatch LogsAWS IoT (p. 373). Pour chaque connexion MQTT que l'agent de message AWS IoT déconnecte en raison de conflits d'ID client, un enregistrement de journal similaire au suivant est généré :

```
{
    "timestamp": "2019-04-28 22:05:30.105",
    "logLevel": "ERROR",
    "traceId": "02a04a93-0b3a-b608-a27c-1ae8ebdb032a",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "Disconnect",
    "protocol": "MQTT",
    "clientId": "clientId01",
    "principalId": "1670fcf6de55adc1930169142405c4a2493d9eb5487127cd0091ca0193a3d3f6",
    "sourceIp": "203.0.113.1",
    "sourcePort": 21335,
    "reason": "DUPLICATE_CLIENT_ID",
    "details": "A new connection was established with the same client ID"
}
```

Vous pouvez utiliser une stratégieCloudWatch Logstels que`{$.reason= "DUPLICATE_CLIENT_ID" }`pour rechercher des instances de conflits d'ID client ou pour configurerFilCloudWatch métriqueset les alarmes CloudWatch correspondantes pour la surveillance et la création de rapports continus.

Vous pouvez utiliserAWS IoTDevice Defenderidentifier trop permissiveAWS IoTet stratégies IAM.AWS IoT Device Defender fournit également une vérification d'audit qui vous avertit si plusieurs appareils de votre parc se connectent à l'AWS IoTCourtier de messages utilisant le même ID client.

Vous pouvez utiliserAWS IoTDevice Advisor pour vérifier que vos appareils peuvent se connecter de manière fiable à AWS IoT Core et suivez les bonnes pratiques de sécurité de.

Voir aussi

- AWS IoT Core
- AWS IoTFonctions de sécurité (p. 232)
- Variables de stratégie AWS IoT Core (p. 273)
- Variables de stratégie IAM
- Amazon Cognito Identity
- AWS IoT Device Defender
- CloudWatch Logs AWS IoT (p. 373)

Veiller à la synchronisation de l'horloge de votre appareil

Il est important que l'heure soit exacte sur votre appareil. Les certificats X.509 ont une date et une heure d'expiration. L'horloge de votre appareil est utilisée pour vérifier qu'un certificat de serveur est toujours valide. Si vous construisez des appareils IoT commerciaux, n'oubliez pas que vos produits peuvent être stockés pendant de longues périodes avant d'être vendus. Les horloges en temps réel peuvent dériver pendant cette période et les batteries peuvent se décharger, par conséquent, il ne suffit pas de régler l'heure en usine.

Pour la plupart des systèmes, cela signifie que le logiciel de l'appareil doit inclure un client NTP (Network Time Protocol). L'appareil doit attendre qu'il se synchronise avec un serveur NTP avant d'essayer de se connecter à AWS IoT Core . Si ce n'est pas possible, le système doit fournir un moyen aux utilisateurs de définir l'heure de l'appareil afin que les connexions suivantes réussissent.

Une fois l'appareil synchronisé avec un serveur NTP, il peut établir une connexion avec AWS IoT Core . L'ampleur du décalage d'horloge autorisé dépend de ce que vous essayez de faire avec la connexion.

Valider le certificat de serveur

La première chose qu'un appareil fait pour interagir avec AWS IoT est d'établir une connexion sécurisée. Lorsque vous connectez votre appareil à AWS IoT, assurez-vous que vous êtes bien en contact avec AWS IoT, et non un autre serveur qui se fait passer pour AWS IoT. Chacun des serveurs AWS IoT est mis en service avec un certificat émis pour le domaine `iot.amazonaws.com`. Ce certificat a été délivré à AWS IoT par une autorité de certification de confiance qui a vérifié notre identité et la propriété du domaine.

L'une des premières choses que fait AWS IoT Core lorsqu'un appareil se connecte est d'envoyer à celui-ci un certificat de serveur. Les appareils peuvent vérifier qu'ils sont censés se connecter à `iot.amazonaws.com` et que le serveur à l'autre extrémité de cette connexion possède un certificat provenant d'une autorité de confiance pour ce domaine.

Les certificats TLS sont au format X.509 et comprennent diverses informations, telles que le nom de l'organisation, l'emplacement, le nom de domaine et une période de validité. La période de validité est spécifiée sous la forme d'une paire de valeurs temporelles nommées `notBefore` et `notAfter`. Les services comme AWS IoT Core utilisent des périodes de validité limitées (par exemple, un an) pour leurs certificats de serveur et commencent à en fournir de nouveaux avant l'expiration des anciens.

Utiliser une identité unique par appareil

Utilisez une identité unique par client. Les appareils utilisent généralement des certificats client X.509. Les applications Web et mobiles utilisent Amazon Cognito Identity. Cela vous permet d'appliquer des autorisations précises à vos appareils.

Par exemple, si une application se compose d'un téléphone portable qui reçoit des mises à jour d'état de deux objets connectés différents : une ampoule et un thermostat. L'ampoule envoie l'état de son niveau de batterie, et un thermostat envoie des messages indiquant la température.

AWS IoT authentifie les appareils individuellement et traite chaque connexion individuellement. Vous pouvez appliquer des contrôles d'accès précis grâce à des stratégies d'autorisation. Vous pouvez définir une stratégie pour le thermostat, qui l'autorise à publier dans un espace de rubrique. Vous pouvez définir une stratégie distincte pour l'ampoule, qui l'autorise à publier dans un autre espace de rubrique. Enfin, vous pouvez définir une stratégie pour l'application mobile, qui l'autorise uniquement à se connecter et à s'abonner aux rubriques du thermostat et de l'ampoule, afin de recevoir des messages de ces appareils.

Appliquez le principe du moins de privilèges et définissez les autorisations par appareil autant que possible. Tous les appareils ou utilisateurs doivent avoir une stratégie AWS IoT dans AWS IoT qui les autorisent uniquement à se connecter avec un ID client connu, à publier et à s'abonner à un ensemble de rubriques fixe et identifié.

Utilisez une seconde Région AWS comme sauvegarde

Envisagez de stocker une copie de vos données en une seconde Région AWS en tant que sauvegarde. Pour de plus amples informations, veuillez consulterReprise après sinistre pourAWS IoT.

Utiliser la mise en service juste à temps

La création et la mise en service manuelles de chaque appareil peuvent prendre du temps. AWS IoT fournit un moyen de définir un modèle pour mettre en service les appareils lorsqu'ils se connectent pour la première fois à AWS IoT. Pour plus d'informations, consultez Mise en service juste-à-temps (p. 737).

Autorisations à exécuterAWS IoTTests Device Advisor

Le modèle de stratégie ci-dessous indique les autorisations minimales et l'entité IAM requises pour exécuterAWS IoTCas de test Device Advisor. Vous devrez remplacer*votre-appareil*avec l'ARN du rôle de périphérique que vous créez sous l'ongletConditions préalables.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "your-device-role-arn",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "iotdeviceadvisor.amazonaws.com"
                }
            }
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "iot:Connect",
                "logs:DescribeLogStreams",
                "iot:DescribeThing",
                "iot:DescribeCertificate",
                "logs:CreateLogGroup",
                "logs:PutLogEvents",
                "iot:DescribeEndpoint",
                "execute-api:Invoke*",
                "logs:CreateLogStream",
                "iot:ListPrincipalPolicies",
                "iot:ListThingPrincipals",
                "iot:ListThings",
```

```
                "iot:Publish",
                "iot:CreateJob",
                "iot:DescribeJob",
                "iot:ListCertificates",
                "iot:ListAttachedPolicies",
                "iot:UpdateThingShadow",
                "iot:GetPolicy"
            ],
            "Resource": "*"
        },
        {
            "Sid": "VisualEditor2",
            "Effect": "Allow",
            "Action": "iotdeviceadvisor:*",
            "Resource": "*"
        }
    ]
}
```

Formation et certification AWS

Suivez le cours suivant pour en savoir plus sur les concepts clés relatifs àAWS IoTSécurité :AWS IoT Apprêt de sécurité.

Surveillance de AWS IoT

La surveillance est un enjeu important pour assurer la fiabilité, la disponibilité et les performances d'AWS IoT et de vos solutions AWS.

Nous vous encourageons fortement à collecter des données de surveillance à partir de toutes les parties de votre solution AWS afin de faciliter le débogage d'une défaillance multi-points, le cas échéant. Commencez par créer un plan de surveillance qui répond aux questions suivantes. Si vous ne savez pas comment y répondre, vous pouvez continuer à activer la journalisation (p. 353) et établir vos lignes de base de performances.

- Quels sont les objectifs de la supervision ?
- Quelles sont les ressources à surveiller ?
- À quelle fréquence les ressources doivent-elles être surveillées ?
- Quels outils de surveillance utiliser ?
- Qui exécute les tâches de supervision ?
- Qui doit être informé en cas de problème ?

L'étape suivante consiste à activer la journalisation (p. 353) et à établir une référence de performances AWS IoT normales dans votre environnement, en mesurant la performance à différents moments et dans différentes conditions de charge. Pendant que vous surveillez AWS IoT, conservez les données de surveillance historiques afin de pouvoir les comparer avec les données de performances actuelles. Cela vous aide à identifier les modèles de performances normaux et les anomalies de performances, et à élaborer des méthodes pour résoudre ces problèmes.

Pour établir vos performances de base pour AWS IoT, vous devez surveiller ces métriques pour démarrer. Vous pouvez toujours surveiller plus de métriques ultérieurement.

- `PublishIn.Success` (p. 365)
- `PublishOut.Success` (p. 365)
- `Subscribe.Success` (p. 365)
- `Ping.Success` (p. 365)
- `Connect.Success` (p. 365)
- `GetThingShadow.Accepted` (p. 368)
- `UpdateThingShadow.Accepted` (p. 368)
- `DeleteThingShadow.Accepted` (p. 368)
- `RulesExecuted` (p. 364)

Les rubriques de cette section peuvent vous aider à démarrer la journalisation et la surveillance de AWS IoT.

Rubriques

- Configurer la journalisation AWS IoT (p. 353)
- ContrôleAWS IoTalarmes et mesures à l'aide d'Amazon CloudWatch (p. 359)
- ContrôleAWS IoTUtilisation CloudWatch Logs (p. 373)

- Journalisez les appels d'API AWS IoT via AWS CloudTrail. (p. 391)

Configurer la journalisation AWS IoT

Vous devez activer la journalisation à l'aide de la console AWS IoT, de l'interface de ligne de commande ou de l'API avant de pouvoir surveiller et consigner l'activité AWS IoT.

Vous pouvez activer la journalisation pour tous les AWS IoT ou uniquement des groupes de choses spécifiques. Vous pouvez configurer la journalisation de AWS IoT à l'aide de la console AWS IoT, de l'interface de ligne de commande ou de l'API ; toutefois, vous devez utiliser l'interface de ligne de commande ou l'API pour configurer la journalisation pour des groupes de choses spécifiques.

Lorsque vous envisagez comment configurer votre journalisation de AWS IoT, la configuration de journalisation par défaut détermine comment l'activité AWS IoT sera consignée, sauf indication contraire. À partir de là, vous pouvez obtenir des journaux détaillés avec un niveau de journal (p. 358) par défaut de `INFO` ou `DEBUG`. Après avoir examiné les journaux initiaux, vous pouvez modifier le niveau de journal par défaut à un niveau moins détaillé tel que `WARN` ou `ERROR`, et définir un niveau de journal plus détaillé spécifique à la ressource sur les ressources qui pourraient nécessiter plus d'attention. Les niveaux de journal peuvent être modifiés quand vous le souhaitez.

Configurer le rôle et la stratégie de journalisation

Avant de pouvoir activer la journalisationAWS IoT, vous devez créer un rôle IAM et une stratégie qui accorde àAWSAutorisation de surveillerAWS IoTen votre nom.

> Note
>
> Avant d'activerAWS IoT, assurez-vous de comprendre les autorisations d'accès à CloudWatch Logs. Les utilisateurs détenant un accès à CloudWatch Logs peuvent consulter les informations de débogage à partir de vos appareils. Pour plus d'informations, consultez Authentification et contrôle d'accès pour Amazon CloudWatch Logs.
> Si vous vous attendez à des modèles de trafic élevés dans AWS IoT Core en raison des tests de charge, pensez à désactiver la journalisation de l'IoT pour empêcher la limitation. Si un trafic élevé est détecté, notre service peut désactiver la connexion à votre compte.

Voici montre comment créer un rôle de journalisation et une stratégie pour AWS IoT Core AWS. Pour plus d'informations sur la création d'un rôle et d'une stratégie de journalisation IAM pour AWS IoT Core pour LoRaWAN, consultezCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103).

Création d'un rôle de journalisation

Pour créer un rôle de journalisation, ouvrez la fenêtreHub de rôles de la console IAMet choisissezCréation d'un rôle.

1. UnderSélectionner le type d'entité de confiance, choisissezAWSService,IoT.
2. UnderSélectionnez votre cas d'utilisation, choisissezIoT, puisSuivant: Permissions (Autorisations).
3. Sur la page qui affiche les stratégies qui sont automatiquement attachées au rôle de service, choisissezSuivant: Tags (Balises), puisSuivant: Review (Examiner).
4. Entrez un nom de rôle et une description de rôle pour le rôle, puis choisissez Créer un rôle.
5. Dans la liste des élémentsRôles, recherchez le rôle que vous avez créé, ouvrez-le et copiez la boîte de dialogueARN de rôle(*`role-arn`*) à utiliser lorsque vousConfigurez la journalisation par défaut dans AWS IoT (console) (p. 354).

Stratégie de rôle de journalisation

Les documents de stratégie suivants fournissent la stratégie de rôle et la stratégie d'approbation qui permettentAWS IoTpour envoyer des entrées de journal à CloudWatch en votre nom. Si vous avez également autorisé AWS IoT Core pour que LoRaWan soumette des entrées de journal, vous verrez un document de stratégie créé pour vous qui consigne les deux activités. Pour plus d'informations sur la création d'un rôle et d'une stratégie de journalisation IAM pour AWS IoT Core pour LoRaWAN, consultezCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103).

> **Note**
>
> Ces documents ont été créés pour vous lorsque vous avez créé le rôle de journalisation.

Stratégie de rôle :

```
{
        "Version": "2012-10-17",
        "Statement": [
            {
                "Effect": "Allow",
                "Action": [
                    "logs:CreateLogGroup",
                    "logs:CreateLogStream",
                    "logs:PutLogEvents",
                    "logs:PutMetricFilter",
                    "logs:PutRetentionPolicy"
                 ],
                "Resource": [
                    "*"
                ]
            }
        ]
    }
```

Stratégie d'approbation pour journaliser uniquement AWS IoT Core activité :

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

Configurez la journalisation par défaut dans AWS IoT (console)

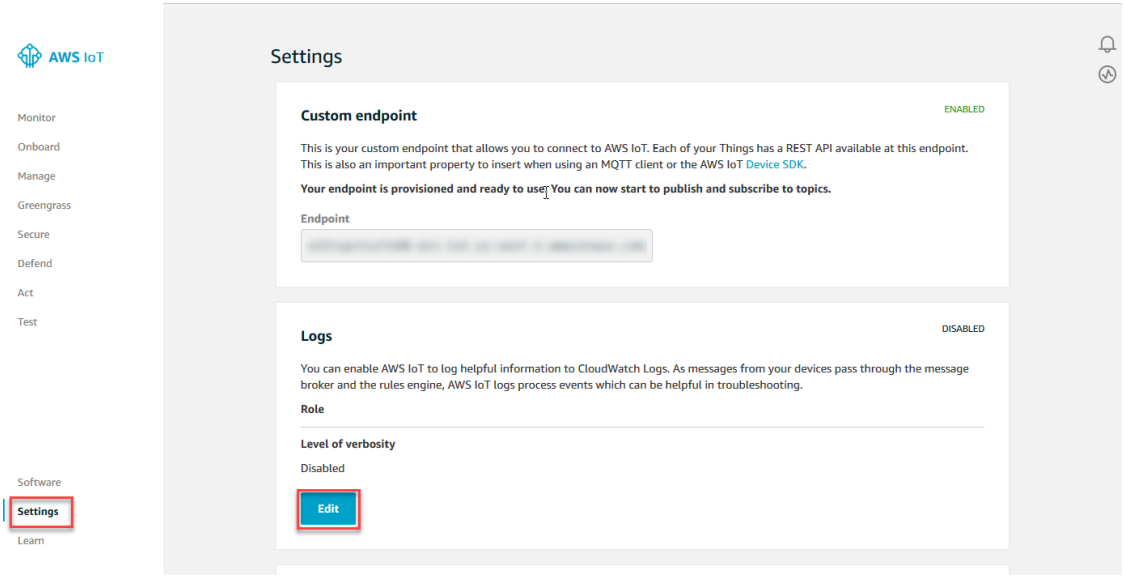
Cette section décrit comment utiliser la console AWS IoT pour configurer la journalisation pour tous les AWS IoT. Pour configurer la journalisation uniquement pour des groupes de choses spécifiques, vous devez utiliser l'interface de ligne de commande ou l'API. Pour de plus amples informations sur la

configuration de la journalisation pour des groupes de choses spécifiques, veuillez consulter Configurer la connexion spécifique aux ressources AWS IoT (interface de ligne de commande) (p. 357).

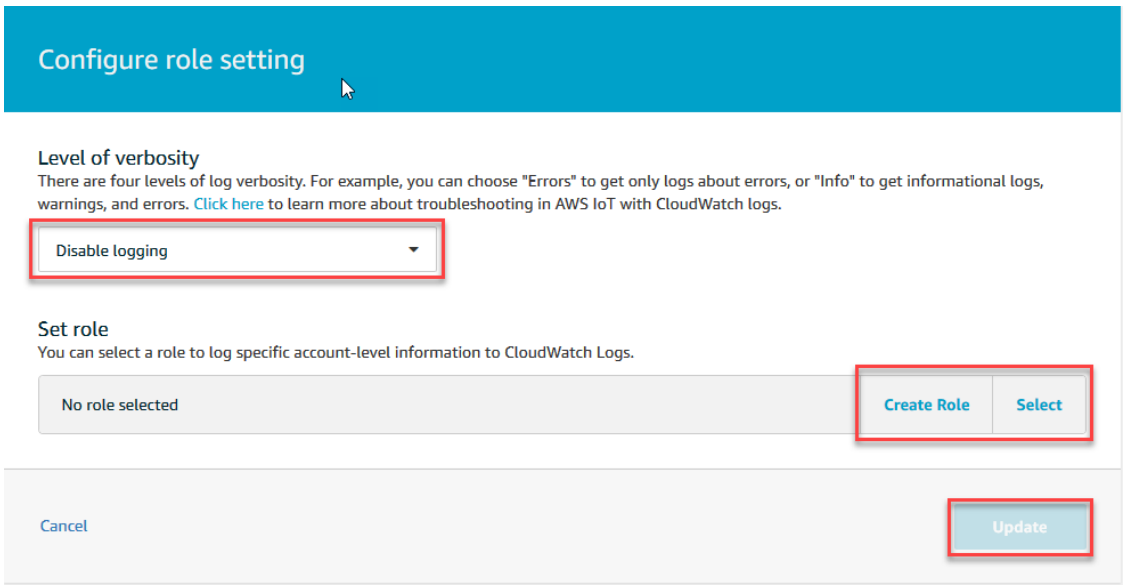
Pour utiliser la console AWS IoT pour configurer la journalisation par défaut pour tous les AWS IoT

1. Connectez-vous à la console AWS IoT. Pour plus d'informations, consultez Ouverture d'AWS IoTconsole (p. 21).
2. Dans le panneau de navigation de gauche, choisissez Paramètres. Dans la section Journaux de la page Paramètres, choisissez Modifier.

   La section Journaux affiche le rôle de journalisation et le niveau de verbosité utilisés par tous les AWS IoT.



3. Dans la pageConfiguration du rôle, choisissez la pageNiveau de détailqui décrit leNiveau de détail (p. 358)des entrées de journal que vous souhaitez afficher dans les journaux CloudWatch.



4. Choisissez Sélectionner pour spécifier un rôle que vous avez créé dans Création d'un rôle de journalisation (p. 353), ou Créer un rôle pour créer un rôle à utiliser pour la journalisation.
5. Choisissez Mettre à jour pour enregistrer vos modifications.

Une fois que vous avez activé la journalisation, visitez Affichage d'unAWS IoTLogs dans la console CloudWatch (p. 373) pour en savoir plus sur l'affichage des entrées du journal.

Configurer la connexion par défaut dans AWS IoT (interface de ligne de commande)

Cette section décrit comment configurer la journalisation globale pour AWS IoT à l'aide de l'interface de ligne de commande.

> Note
>
> Vous avez besoin du nom de ressource Amazon (ARN) du rôle que vous souhaitez utiliser. Si vous devez créer un rôle à utiliser pour la journalisation, veuillez consulter Création d'un rôle de journalisation (p. 353) avant de continuer.
> Le principal utilisé pour appeler l'API doit avoir Transmettre les autorisations de rôle (p. 396) pour votre rôle de journalisation.

Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

Pour utiliser l'interface de ligne de commande pour configurer la journalisation par défaut pour AWS IoT

1. Utilisez la commande set-v2-logging-options pour définir les options de journalisation de votre compte.

```
aws iot set-v2-logging-options \
    --role-arn logging-role-arn \
    --default-log-level log-level
```

   où :

   --role-arn

   L'ARN du rôle qui accordeAWS IoTVous pouvez écrire dans vos journaux dans CloudWatch Logs.

   --default-log-level

   Le niveau de journalisation (p. 358) à utiliser. Les valeurs valables sont `ERROR`, `WARN`, `INFO` `DEBUG` ou `DISABLED`

   --no-disable-all-logs

   Paramètre facultatif qui active toute la journalisation de AWS IoT. Utilisez ce paramètre pour activer la journalisation lorsqu'elle est désactivée.

   --disable-all-logs

   Paramètre facultatif qui désactive toute la journalisation de AWS IoT. Utilisez ce paramètre pour désactiver la journalisation lorsqu'elle est activée.

2. Utilisez la commande get-v2-logging-options pour obtenir vos options de journalisation actuelles.

```
aws iot get-v2-logging-options
```

Une fois que vous avez activé la journalisation, visitez Affichage d'unAWS IoTLogs dans la console CloudWatch (p. 373) pour en savoir plus sur l'affichage des entrées du journal.

Note

AWS IoT continue à prendre en charge les anciennes commandes pour définir et obtenir la journalisation globale sur votre compte : set-logging-options et get-logging-options. Lorsque ces commandes sont utilisées, les journaux obtenus contiennent du texte brut et non des charges de travail JSON et la latence de journalisation est généralement plus élevée. Aucune amélioration supplémentaire ne sera apportée à l'implémentation de ces anciennes commandes. Nous vous recommandons d'utiliser les versions « v2 » pour configurer vos options de journalisation et, si possible, de modifier toutes les applications existantes qui utilisent les anciennes versions.

Configurer la connexion spécifique aux ressources AWS IoT (interface de ligne de commande)

Cette section décrit comment configurer la journalisation spécifique à la ressource pour AWS IoT à l'aide de l'interface de ligne de commande. La journalisation spécifique à la ressource vous permet de spécifier un niveau de journalisation pour un groupe d'objets (p. 210)spécifique.

Les groupes d'objets peuvent contenir d'autres groupes d'objets pour créer une relation hiérarchique. Cette procédure décrit comment configurer la journalisation d'un seul groupe d'objets. Vous pouvez appliquer cette procédure au groupe d'objets parent dans une hiérarchie pour configurer la journalisation de tous les groupes d'objets de la hiérarchie. Vous pouvez également appliquer cette procédure à un groupe d'objets enfant pour remplacer la configuration de journalisation de son parent.

Note

Vous avez besoin du nom de ressource Amazon (ARN) du rôle que vous souhaitez utiliser. Si vous devez créer un rôle à utiliser pour la journalisation, veuillez consulter Création d'un rôle de journalisation (p. 353) avant de continuer.
Le principal utilisé pour appeler l'API doit avoir Transmettre les autorisations de rôle (p. 396) pour votre rôle de journalisation.

Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

Pour utiliser l'interface de ligne de commande pour configurer la journalisation spécifique aux ressources pour AWS IoT

1. Utilisez la commande set-v2-logging-options pour définir les options de journalisation de votre compte.

```
aws iot set-v2-logging-options \
    --role-arn logging-role-arn \
    --default-log-level log-level
```

où :

--role-arn

L'ARN du rôle qui accordeAWS IoTVous pouvez écrire dans vos journaux dans CloudWatch Logs.

--default-log-level

Le niveau de journalisation (p. 358) à utiliser. Les valeurs valables sont `ERROR`, `WARN`, `INFO` `DEBUG` ou `DISABLED`

--no-disable-all-logs

Paramètre facultatif qui active toute la journalisation de AWS IoT. Utilisez ce paramètre pour activer la journalisation lorsqu'elle est désactivée.

--disable-all-logs

Paramètre facultatif qui désactive toute la journalisation de AWS IoT. Utilisez ce paramètre pour désactiver la journalisation lorsqu'elle est activée.

2. Utilisez la commande set-v2-logging-level pour configurer la journalisation spécifique à une ressource pour un groupe de choses.

```
aws iot set-v2-logging-level \
              --log-target targetType=THING_GROUP,targetName=thing_group_name \
              --log-level log_level
```

--log-target

Type et nom de la ressource pour laquelle vous configurez la journalisation. La valeur `target_type` doit être `THING_GROUP`. La valeur du paramètre log-target peut être du texte, comme indiqué dans l'exemple de commande précédent, ou une chaîne JSON, telle que l'exemple suivant.

```
aws iot set-v2-logging-level \
              --log-target '{"targetType": "THING_GROUP","targetName":
 "thing_group_name"}' \
              --log-level log_level
```

--log-level

Le niveau de journalisation utilisé lors de la génération de journaux pour la ressource spécifiée. Les valeurs valides sont : DEBUG, INFO, ERROR, WARN et DISABLED

3. Utilisez la commande list-v2-logging-levels pour répertorier les niveaux de journalisation actuellement configurés.

```
aws iot list-v2-logging-levels
```

4. Utilisez la commande delete-v2-logging-level pour supprimer un niveau de journalisation spécifique à la ressource.

```
aws iot delete-v2-logging-level \
              --targetType "THING_GROUP" \
              --targetName "thing_group_name"
```

--targetType

La valeur `target_type` doit être `THING_GROUP`.

--targetName

Nom du groupe d'objets pour lequel le niveau de journalisation doit être supprimé.

Une fois que vous avez activé la journalisation, visitez Affichage d'unAWS IoTLogs dans la console CloudWatch (p. 373) pour en savoir plus sur l'affichage des entrées du journal.

Niveaux de journalisation

Ces niveaux de journal déterminent les événements qui sont consignés et s'appliquent aux niveaux de journal par défaut et spécifiques aux ressources.

ERROR

Toute erreur qui entraîne l'échec d'une opération.

Les journaux contiennent uniquement des informations ERROR.

WARN

Tout ce qui peut éventuellement entraîner des incohérences dans le système, mais qui n'entraîne pas nécessairement l'échec de l'opération.

Les journaux contiennent des informations ERROR et WARN.

INFO

Informations générales sur le flux des objets.

Les journaux contiennent des informations INFO, ERROR et WARN.

DEBUG

Informations qui peuvent être utiles lors du débogage d'un problème.

Les journaux contiennent des informations DEBUG, INFO, ERROR et WARN.

DISABLED

Toute la journalisation est désactivée.

ContrôleAWS IoTalarmes et mesures à l'aide d'Amazon CloudWatch

Vous pouvez surveiller AWS IoT à l'aide de CloudWatch, qui collecte et traite les données brutes d'AWS IoT en métriques lisibles et disponibles pratiquement en temps réel. Ces statistiques sont enregistrées pour une durée de deux semaines et, par conséquent, vous pouvez accéder aux informations historiques et acquérir un meilleur point de vue de la façon dont votre service ou application web s'exécute. Par défaut,AWS IoTLes données des métriques sont automatiquement envoyées à CloudWatch à toutes les minutes. Pour de plus amples informations, veuillez consulterQue sont Amazon CloudWatch, les Amazon CloudWatch Events et les journaux Amazon CloudWatch Logs ?dans leGuide de l'utilisateur Amazon CloudWatch..

Utilisation de métriques AWS IoT

Les métriques présentées par AWS IoT fournissent des informations qui permettent divers types d'analyses. Les cas d'utilisation suivants sont basés sur un scénario où vous avez dix objets qui se connectent à Internet une fois par jour. Chaque jour :

- Dix objets se connectent à AWS IoT en même temps.
- Chaque objet s'abonne à un filtre de rubrique, puis attend une heure avant de déconnecter. Au cours de cette période, les objets communiquent entre eux pour en savoir plus sur l'état du monde.
- Chaque objet publie sa perception, d'après les données qu'il vient de détecter avec `UpdateThingShadow`.
- Chaque objet se déconnecte de AWS IoT.

Pour vous aider à démarrer, ces rubriques explorent certaines des questions que vous pourriez avoir.

- Comment puis-je être informé si mes objets ne se connectent pas chaque jour ? (p. 360)

- Comment puis-je être informé si mes objets ne publient pas de données chaque jour ? (p. 361)
- Comment puis-je être informé si les mises à jour du shadow de mon objet sont rejetées chaque jour ? (p. 361)
- Comment créer une alarme CloudWatch pour les travaux ? (p. 362)

En savoir plus sur les alarmes et les métriques CloudWatch

- Création d'alarmes CloudWatch pour la surveillance d'AWS IoT (p. 360)
- Métriques et dimensions d'AWS IoT (p. 363)

Création d'alarmes CloudWatch pour la surveillance d'AWS IoT

Créez une alarme CloudWatch qui envoie un message Amazon SNS lorsque l'alarme change de statut. Une alarme surveille une métrique sur la période que vous spécifiez. Lorsque la valeur de la métrique dépasse un seuil donné sur un certain nombre de périodes, une ou plusieurs actions sont effectuées. L'action est une notification envoyée à une rubrique Amazon SNS ou à une stratégie Auto Scaling. Les alarmes déclenchent les actions pour les changements d'état soutenus uniquement. Les alarmes CloudWatch ne déclenchent pas d'actions simplement parce qu'elles sont dans un état particulier : l'état doit avoir changé et été maintenu pendant un certain nombre de périodes.

Les rubriques suivantes décrivent quelques exemples d'utilisation d'alarmes CloudWatch.

- Comment puis-je être informé si mes objets ne se connectent pas chaque jour ? (p. 360)
- Comment puis-je être informé si mes objets ne publient pas de données chaque jour ? (p. 361)
- Comment puis-je être informé si les mises à jour du shadow de mon objet sont rejetées chaque jour ? (p. 361)
- Comment créer une alarme CloudWatch pour les travaux ? (p. 362)

Vous pouvez voir toutes les métriques sur lesquelles les alarmes CloudWatch peuvent surveiller.Métriques et dimensions d'AWS IoT (p. 363).

Comment puis-je être informé si mes objets ne se connectent pas chaque jour ?

1. Créez une rubrique Amazon SNS nommée`things-not-connecting-successfully`et enregistrez son ARN (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que *`sns-topic-arn`*.

   Pour de plus amples informations sur la création d'une notification Amazon SNS, veuillez consulter.Démarrer avec Amazon SNS.
2. Créez l'alarme.

```
aws cloudwatch put-metric-alarm \
    --alarm-name ConnectSuccessAlarm \
    --alarm-description "Alarm when my Things don't connect successfully" \
    --namespace AWS/IoT \
    --metric-name Connect.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
```

```
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Testez l'alarme.

```
aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
 "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
 "initializing" --state-value ALARM
```

4. Vérifiez que l'alarme s'affiche dans votreConsole CloudWatch.

Comment puis-je être informé si mes objets ne publient pas de données chaque jour ?

1. Créez une rubrique Amazon SNS nomméethings-not-publishing-dataet enregistrez son ARN (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que *sns-topic-arn*.

   Pour de plus amples informations sur la création d'une notification Amazon SNS, veuillez consulter.Démarrer avec Amazon SNS.

2. Créez l'alarme.

```
aws cloudwatch put-metric-alarm \
    --alarm-name PublishInSuccessAlarm\
    --alarm-description "Alarm when my Things don't publish their data \
    --namespace AWS/IoT \
    --metric-name PublishIn.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Testez l'alarme.

```
aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
 "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
 "initializing" --state-value ALARM
```

4. Vérifiez que l'alarme s'affiche dans votreConsole CloudWatch.

Comment puis-je être informé si les mises à jour du shadow de mon objet sont rejetées chaque jour ?

1. Créez une rubrique Amazon SNS nomméethings-shadow-updates-rejectedet enregistrez son ARN (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que *sns-topic-arn*.

   Pour de plus amples informations sur la création d'une notification Amazon SNS, veuillez consulter.Démarrer avec Amazon SNS.

2. Créez l'alarme.

```
aws cloudwatch put-metric-alarm \
    --alarm-name UpdateThingShadowSuccessAlarm \
    --alarm-description "Alarm when my Things Shadow updates are getting rejected" \
    --namespace AWS/IoT \
    --metric-name UpdateThingShadow.Success \
    --dimensions Name=Protocol,Value=MQTT \
    --statistic Sum \
    --threshold 10 \
    --comparison-operator LessThanThreshold \
    --period 86400 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions sns-topic-arn
```

3. Testez l'alarme.

```
aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-
reason "initializing" --state-value OK
```

```
aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-
reason "initializing" --state-value ALARM
```

4. Vérifiez que l'alarme s'affiche dans votreConsole CloudWatch.

Comment créer une alarme CloudWatch pour les travaux ?

Le service Jobs fournit des métriques CloudWatch qui vous permettent de surveiller vos travaux. Vous pouvez créer des alarmes CloudWatch pour surveiller lesMétriques de tâches (p. 368).

La commande suivante crée une alarme CloudWatch pour surveiller le nombre total d'exécutions de travaux ayant échoué pour Job*SampleOtaJob*et vous avertit quand plus de 20 exécutions de travail ont échoué. L'alarme surveille la métrique Jobs `FailedJobExecutionTotalCount` en vérifiant la valeur signalée toutes les 300 secondes. Il est activé lorsqu'une seule valeur signalée est supérieure à 20, ce qui signifie qu'il y a eu plus de 20 exécutions de travail ayant échoué depuis le début de la tâche. Lorsque l'alarme s'éteint, elle envoie une notification à la rubrique Amazon SNS fournie.

```
aws cloudwatch put-metric-alarm \
    --alarm-name TotalFailedJobExecution-SampleOTAJob \
    --alarm-description "Alarm when total number of failed job execution exceeds the
 threshold for SampleOTAJob" \
    --namespace AWS/IoT \
    --metric-name FailedJobExecutionTotalCount \
    --dimensions Name=JobId,Value=SampleOTAJob \
    --statistic Sum \
    --threshold 20 \
    --comparison-operator GreaterThanThreshold \
    --period 300 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions arn:aws:sns:<AWS_REGION>:<AWS_ACCOUNT_ID>:SampleOTAJob-has-too-many-
failed-job-ececutions
```

La commande suivante crée une alarme CloudWatch pour surveiller le nombre d'exécutions de travaux ayant échoué pour Job*SampleOtaJob*Dans une période donnée. Vous êtes ensuite averti quand plus de cinq exécutions de travail ont échoué au cours de cette période. L'alarme surveille la métrique Jobs `FailedJobExecutionCount` en vérifiant la valeur signalée toutes les 3600 secondes. Il est activé

lorsqu'une seule valeur signalée est supérieure à 5, ce qui signifie qu'il y a eu plus de 5 exécutions de travail ayant échoué au cours de la dernière heure. Lorsque l'alarme s'éteint, elle envoie une notification à la rubrique Amazon SNS fournie.

```
aws cloudwatch put-metric-alarm \
    --alarm-name FailedJobExecution-SampleOTAJob \
    --alarm-description "Alarm when number of failed job execution per hour exceeds the
 threshold for SampleOTAJob" \
    --namespace AWS/IoT \
    --metric-name FailedJobExecutionCount \
    --dimensions Name=JobId,Value=SampleOTAJob \
    --statistic Sum \
    --threshold 5 \
    --comparison-operator GreaterThanThreshold \
    --period 3600 \
    --unit Count \
    --evaluation-periods 1 \
    --alarm-actions arn:aws:sns:<AWS_REGION>:<AWS_ACCOUNT_ID>:SampleOTAJob-has-too-many-
failed-job-ececutions-per-hour
```

Métriques et dimensions d'AWS IoT

Lorsque vous interagissez avecAWS IoT, le service envoie les métriques et les dimensions suivantes à CloudWatch toutes les minutes. Vous pouvez utiliser les procédures suivantes pour afficher les métriques d'AWS IoT.

Pour consulter les métriques (console CloudWatch)

Les métriques sont d'abord regroupées par espace de noms de service, puis par les différentes combinaisons de dimension au sein de chaque espace de noms.

1. Ouvrez la console CloudWatch à l'adresse https://console.aws.amazon.com/cloudwatch/.
2. Dans le volet de navigation, sélectionnez Metrics (Métriques).
3. Dans le volet Métriques CloudWatch par catégorie, sous la catégorie de métriques pour AWS IoT, sélectionnez une catégorie de métriques, puis, dans le volet supérieur, faites défiler l'écran pour afficher la liste complète des métriques.

Pour afficher les métriques (CLI)

- À partir d'une invite de commande, utilisez la commande suivante :

```
aws cloudwatch list-metrics --namespace "AWS/IoT"
```

CloudWatch affiche les groupes de métriques suivants pourAWS IoT :

- Métriques AWS IoT (p. 364)
- AWS IoT Core métriques du fournisseur d'informations d'identification (p. 364)
- Métriques de règle (p. 364)
- Métriques d'action de règle (p. 365)
- Métriques spécifiques à l'action HTTP (p. 365)
- Métriques d'agent de messages (p. 365)
- Métriques de shadow d'appareil (p. 368)

Métriques AWS IoT

Métrique	Description
`AddThingToDynamicThingGroupsFailed`	Nombre d'événements d'échec associés à l'ajout d'un objet à un groupe d'objets dynamiques. La dimension `DynamicThingGroupName` contient le nom des groupes dynamiques qui n'ont pas pu ajouter des objets.
`NumLogBatchesFailedToPublishThrottled`	Le lot singulier d'événements de journaux qui ne s'est pas publié en raison d'erreurs de limitation.
`NumLogEventsFailedToPublishThrottled`	Le nombre d'événements de journaux au sein du lot qui ne s'est pas publié en raison d'erreurs de limitation.

AWS IoT Core métriques du fournisseur d'informations d'identification

Métrique	Description
`CredentialExchangeSuccess`	Nombre de succès`AssumeRoleWithCertificate`Demandes de AWS IoT Core fournisseur d'informations d'identification.

Métriques de règle

Métrique	Description
`ParseError`	Nombre d'erreurs d'analyse JSON s'étant produites dans des messages publiés dans une rubrique dans laquelle une règle écoute. La dimension `RuleName` contient le nom de la règle.
`RuleMessageThrottled`	Le moteur de règles limite le nombre de messages en raison d'un comportement malveillant ou parce que le nombre de messages dépasse la limite du moteur de règles. La dimension `RuleName` contient le nom de la règle à déclencher.
`RuleNotFound`	La règle à déclencher est introuvable. La dimension `RuleName` contient le nom de la règle.
`RulesExecuted`	Nombre de règles AWS IoT exécutées.

Métrique	Description
`TopicMatch`	Nombre de messages entrants publiés dans une rubrique dans laquelle une règle écoute. La dimension `RuleName` contient le nom de la règle.

Métriques d'action de règle

Métrique	Description
`Failure`	Nombre d'appels d'action de règle en échec. La dimension `RuleName` contient le nom de la règle qui spécifie l'action. La dimension `ActionType` contient le type d'action ayant été appelé.
`Success`	Nombre d'appels d'action de règle réussis. La dimension `RuleName` contient le nom de la règle qui spécifie l'action. La dimension `ActionType` contient le type d'action ayant été appelé.

Métriques spécifiques à l'action HTTP

Métrique	Description
`HttpCode_Other`	Généré si le code de statut de la réponse du service web / de l'application en aval n'est pas 2xx, 4xx ou 5xx.
`HttpCode_4XX`	Généré si le code de statut de la réponse du service web / de l'application en aval est compris entre 400 et 499.
`HttpCode_5XX`	Généré si le code de statut de la réponse du service web / de l'application en aval est compris entre 500 et 599.
`HttpInvalidUrl`	Généré si une URL de point de terminaison, après remplacement des modèles de substitution, ne commence pas par `https://`.
`HttpRequestTimeout`	Généré si le service web / l'application en aval ne renvoie pas de réponse dans le délai d'expiration de la demande. Pour de plus amples informations, veuillez consulter Quotas de service.
`HttpUnknownHost`	Généré si l'URL est valide, mais que le service n'existe pas ou est inaccessible.

Métriques d'agent de messages

> **Note**
>
> Les métriques d'agent de message sont affichées dans la console CloudWatch sousmesures de protocole.

Métrique	Description
`Connect.AuthError`	Nombre de demandes de connexion n'ayant pas pu être autorisées par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Connect.ClientError`	Le nombre de demandes de connexion rejetées, car le message MQTT ne respectait pas les exigences définies dans Quotas AWS IoT (p. 1158). La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Connect.ClientIDThrottle`	Nombre de demandes de connexion limitées car le client a dépassé le taux de demandes de connexion autorisé pour un ID client spécifique. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Connect.ServerError`	Nombre de demandes de connexion ayant échoué à cause d'une erreur interne. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Connect.Success`	Nombre de connexions réussies à l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Connect.Throttle`	Nombre de demandes de connexion ayant été limitées car le compte dépassait le taux de demandes de connexion autorisé. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `CONNECT`.
`Ping.Success`	Nombre de messages ping reçus par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message ping.
`PublishIn.AuthError`	Nombre de demandes de publication que l'agent de messages n'a pas pu autoriser. La dimension `Protocol` contient le protocole utilisé pour publier le message.
`PublishIn.ClientError`	Le nombre de demandes de publication rejetées par l'agent de messages, car le message ne respectait pas les exigences définies dans les Quotas AWS IoT (p. 1158). La dimension `Protocol` contient le protocole utilisé pour publier le message.
`PublishIn.ServerError`	Nombre de demandes de publication que l'agent de messages n'a pas pu traiter à cause d'une erreur interne. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishIn.Success`	Nombre de demandes de publication traitées avec succès par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishIn.Throttle`	Nombre de demandes de publication ayant été limitées car le client dépassait le taux de messages entrants

Métrique	Description
	autorisé. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishOut.AuthError`	Nombre de demandes de publication effectuées par l'agent de messages n'ayant pas pu être autorisées par AWS IoT. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishOut.ClientError`	Le nombre de demandes de publication effectuées par l'agent de messages qui ont été rejetées, car le message ne respectait pas les exigences définies dans Quotas AWS IoT (p. 1158). La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishOut.Success`	Nombre de demandes de publication effectuées avec succès par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`PublishOut.Throttle`	Nombre de demandes de publication ayant été limitées car le client dépassait le taux de messages sortants autorisé. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `PUBLISH`.
`Subscribe.AuthError`	Nombre de demandes d'abonnement adressées par un client et n'ayant pas pu être autorisées. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `SUBSCRIBE`.
`Subscribe.ClientError`	Le nombre de demandes d'abonnement rejetées, car le message `SUBSCRIBE` ne respectait pas les exigences définies dans Quotas AWS IoT (p. 1158). La dimension `Protocol` contient le protocole utilisé pour envoyer le message `SUBSCRIBE`.
`Subscribe.ServerError`	Nombre de demandes d'abonnement ayant été rejetées à cause d'une erreur interne. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `SUBSCRIBE`.
`Subscribe.Success`	Nombre de demandes d'abonnement traitées avec succès par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `SUBSCRIBE`.
`Subscribe.Throttle`	Nombre de demandes d'abonnement ayant été limitées car le client dépassait le taux de demandes d'abonnement autorisé. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `SUBSCRIBE`.
`Unsubscribe.ClientError`	Le nombre de demandes de désabonnement rejetées, car le message `UNSUBSCRIBE` ne respectait pas les exigences définies dans Quotas AWS IoT (p. 1158). La dimension `Protocol` contient le protocole utilisé pour envoyer le message `UNSUBSCRIBE`.

Métrique	Description
`Unsubscribe.ServerError`	Nombre de demandes d'annulation d'abonnement ayant été rejetées à cause d'une erreur interne. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `UNSUBSCRIBE`.
`Unsubscribe.Success`	Nombre de demandes d'annulation d'abonnement traitées avec succès par l'agent de messages. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `UNSUBSCRIBE`.
`Unsubscribe.Throttle`	Nombre de demandes d'annulation d'abonnement ayant été rejetées car le client dépassait le taux de demandes d'annulation d'abonnement autorisé. La dimension `Protocol` contient le protocole utilisé pour envoyer le message `UNSUBSCRIBE`.

Métriques de shadow d'appareil

Note

Les métriques de shadow d'appareil sont affichées dans la console CloudWatch, sousmesures de protocole.

Métrique	Description
`DeleteThingShadow.Accepted`	Nombre de demandes `DeleteThingShadow` traitées avec succès. La dimension `Protocol` contient le protocole utilisé pour effectuer la demande.
`GetThingShadow.Accepted`	Nombre de demandes `GetThingShadow` traitées avec succès. La dimension `Protocol` contient le protocole utilisé pour effectuer la demande.
`ListThingShadow.Accepted`	Nombre de demandes `ListThingShadow` traitées avec succès. La dimension `Protocol` contient le protocole utilisé pour effectuer la demande.
`UpdateThingShadow.Accepted`	Nombre de demandes `UpdateThingShadow` traitées avec succès. La dimension `Protocol` contient le protocole utilisé pour effectuer la demande.

Métriques de tâches

Métrique	Description
`CanceledJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `CANCELED` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.

Métrique	Description
`CanceledJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `CANCELED` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`ClientErrorCount`	Nombre d'erreurs client générées pendant l'exécution de la tâche. La dimension `JobId` contient l'ID de la tâche.
`FailedJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `FAILED` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.
`FailedJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `FAILED` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`InProgressJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `IN_PROGRESS` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.
`InProgressJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `IN_PROGRESS` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`RejectedJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `REJECTED` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`RemovedJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `REMOVED` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`QueuedJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `QUEUED` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.
`QueuedJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `QUEUED` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.
`RejectedJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `REJECTED` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.
`RemovedJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `REMOVED` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.

Métrique	Description
`ServerErrorCount`	Nombre d'erreurs de serveur générées pendant l'exécution de la tâche. La dimension `JobId` contient l'ID de la tâche.
`SuccededJobExecutionCount`	Nombre d'exécutions de tâche dont le statut est passé à `SUCCESS` au cours d'une période définie par CloudWatch. (Pour plus d'informations sur les métriques CloudWatch, consultez Métriques Amazon CloudWatch.) La dimension `JobId` contient l'ID de la tâche.
`SuccededJobExecutionTotalCount`	Nombre total d'exécutions de tâche dont le statut est `SUCCESS` pour la tâche donnée. La dimension `JobId` contient l'ID de la tâche.

Métriques d'audit Device Defender

Métrique	Description
`NonCompliantResources`	Nombre de ressources détectées comme non conformes avec un contrôle. Le système renvoie le nombre de ressources non conformes pour chaque contrôle de chaque audit effectué.
`ResourcesEvaluated`	Nombre de ressources évaluées pour conformité. Le système renvoie le nombre de ressources évaluées pour chaque contrôle de chaque audit effectué.

Métriques de détection Device Defender

Métrique	Description
`Violations`	Nombre de nouvelles violations de comportement de profil de sécurité détectées depuis la dernière évaluation. Le système signale le nombre de nouvelles violations pour le compte, pour un profil de sécurité spécifique et pour un comportement spécifique d'un profil de sécurité spécifique.
`ViolationsCleared`	Nombre de violations de comportements de profil de sécurité résolues depuis la dernière évaluation. Le système signale le nombre de violations résolues pour le compte, pour un profil de sécurité spécifique et pour un comportement spécifique d'un profil de sécurité spécifique.
`ViolationsInvalidated`	Nombre de violations de comportement de profil de sécurité pour lesquelles les informations ne sont plus disponibles depuis la dernière évaluation (l'appareil de génération de rapports ayant arrêté de créer des rapports ou n'étant plus surveillé pour une raison quelconque). Le système signale le nombre de violations non validées pour la totalité du compte, pour un profil de sécurité

Métrique	Description
	spécifique et pour un comportement spécifique d'un profil de sécurité spécifique.

Métriques de mise en service d'appareils

AWS IoTmesures de mise en service d'une flotte

Métrique	Description
`ApproximateNumberOfThingsRegistered`	Nombre d'objets qui ont été enregistrés par Fleet Provisioning. Bien que le dénombrement soit généralement exact, l'architecture distribuée de AWS IoT Core rend difficile le maintien d'un dénombrement précis des objets enregistrés. La statistique à utiliser pour cette métrique est la suivante : • Taille maxpour indiquer le nombre total de choses qui ont été enregistrées. Pour obtenir un nombre d'éléments enregistrés au cours de la fenêtre d'agrégation CloudWatch, consultez la`RegisterThingFailed`Métriques. Dimensions : ClaimCertificateID (p. 372)
`CreateKeysAndCertificateFailed`	Nombre d'échecs survenus lors des appels vers le serveur`CreateKeysAndCertificate`MQTT API. La mesure est émise dans les cas Success (valeur = 0) et Echec (valeur = 1). Cette mesure permet de suivre le nombre de certificats créés et enregistrés pendant les fenêtres d'agrégation prises en charge par CloudWatch, telles que 5 minutes ou 1 heure. Les statistiques disponibles pour cette mesure sont les suivantes : • Sumpour signaler le nombre d'appels ayant échoué. • Exemple de comptagepour signaler le nombre total d'appels réussis et échoués.
`CreateCertificateFromCsrFailed`	Nombre d'échecs survenus lors des appels vers le serveur`CreateCertificateFromCsr`MQTT API. La mesure est émise dans les cas Success (valeur = 0) et Echec (valeur = 1). Cette mesure peut être utilisée pour suivre le nombre d'éléments enregistrés pendant les fenêtres d'agrégation prises en charge par CloudWatch, telles que 5 minutes ou 1 heure. Les statistiques disponibles pour cette mesure sont les suivantes :

Métrique	Description
	• Sumpour signaler le nombre d'appels ayant échoué. • Exemple de comptagepour signaler le nombre total d'appels réussis et échoués.
`RegisterThingFailed`	Nombre d'échecs survenus lors des appels vers le serveur`RegisterThing`MQTT API. La mesure est émise dans les cas Success (valeur = 0) et Echec (valeur = 1). Cette mesure peut être utilisée pour suivre le nombre d'éléments enregistrés pendant les fenêtres d'agrégation prises en charge par CloudWatch, telles que 5 minutes ou 1 heure. Pour le nombre total de choses enregistrées, consultez la`ApproximateNumberOfThingsRegistered`Métriques. Les statistiques disponibles pour cette mesure sont les suivantes : • Sumpour signaler le nombre d'appels ayant échoué. • Exemple de comptagepour signaler le nombre total d'appels réussis et échoués. Dimensions : TemplateName (p. 372)

Métriques de mise en service juste-à-temps

Métrique	Description
`ProvisionThing.ClientError`	Nombre de fois qu'un périphérique n'a pas pu provisionner en raison d'une erreur client. Par exemple, la stratégie spécifiée dans le modèle n'existait pas.
`ProvisionThing.ServerError`	Nombre de fois qu'un périphérique n'a pas pu provisionner en raison d'une erreur de serveur. Les clients peuvent réessayer de provisionner l'appareil après avoir attendu et ils peuvent contacterAWS IoTSi la question reste la même.
`ProvisionThing.Success`	Nombre de fois où un appareil a été mis en service avec succès.

Dimensions pour les métriques

Les métriques utilisent l'espace de noms et fournissent des métriques pour les dimensions suivantes.

Dimension	Description
`ActionType`	Le type d'action (p. 401) spécifié par la règle déclenchée par la demande.
`BehaviorName`	Nom du comportement du profil de sécurité de détection Device Defender qui est surveillé.

Dimension	Description
`ClaimCertificateId`	La `certificateId`de la revendication utilisée pour fournir les instruments.
`CheckName`	Nom du contrôle d'audit Device Defender dont les résultats sont surveillés.
`JobId`	ID de la tâche dont la progression ou la réussite/l'échec de connexion du message est surveillé(e).
`Protocol`	Protocole utilisé pour effectuer la demande. Les valeurs valides sont : MQTT ou HTTP
`RuleName`	Nom de la règle déclenchée par la demande.
`ScheduledAuditName`	Nom de l'audit Device Defender programmé dont les résultats du contrôle sont surveillés. La valeur `OnDemand` est utilisée si les résultats concernent un audit effectué à la demande.
`SecurityProfileName`	Nom du profil de sécurité de détection Device Defender dont les comportements sont surveillés.
`TemplateName`	Nom du modèle de mise en service.

ContrôleAWS IoTUtilisation CloudWatch Logs

Lorsque la journalisation AWS IoT est activée (p. 353), AWS IoT envoie des événements de progression sur chaque message au fur et à mesure qu'il passe de vos appareils via le courtier de messages et le moteur de règles. DansConsole CloudWatch, CloudWatch Logs apparaissent dans un groupe de journaux nomméAWSIotLogs.

Pour de plus amples informations sur CloudWatch Logs, veuillez consulter.CloudWatch Logs. Pour plus d'informations sur lesAWS IoTCloudWatch Logs, voir.CloudWatchAWS IoTentrées du journal (p. 374).

Affichage d'unAWS IoTLogs dans la console CloudWatch

> Note
>
> La `AWSIotLogsV2`n'est pas visible dans la console CloudWatch tant que :
>
> - Vous avez activé la connexionAWS IoT. Pour plus d'informations sur la façon d'activer la connexionAWS IoT, voirConfigurer la journalisation AWS IoT (p. 353)
> - Certaines entrées de journal ont été écrites parAWS IoT.

Pour afficher vos rapportsAWS IoTLogs dans la console CloudWatch

1. Parcourir jusqu'àhttps://console.aws.amazon.com/cloudwatch/. Dans le panneau de navigation, sélectionnez Groupes de journaux.
2. Dans la zone de texte Filtre, entrez **`AWSIotLogsV2`**, puis appuyez sur Entrée.
3. Double-cliquez sur le groupe de journaux `AWSIotLogsV2`.
4. ChoisissezRechercher tout. Une liste complète des journaux AWS IoT générés pour votre compte s'affiche.

5. Choisissez l'icône de développement pour afficher un flux individuel.

Vous pouvez également entrer une requête dans la zone de texte Filtrer les événements. Voici quelques demandes intéressantes à essayer :

- `{ $.logLevel = "INFO" }`

  Trouvez tous les journaux qui ont un niveau de journalisation `INFO`.
- `{ $.status = "Success" }`

  Trouvez tous les journaux qui ont un statut `Success`.
- `{ $.status = "Success" && $.eventType = "GetThingShadow" }`

  Trouvez tous les journaux qui ont un statut `Success` et un type d'événement `GetThingShadow`.

Pour de plus amples informations sur la création d'expressions de filtre, veuillez consulterCloudWatch Logs.

CloudWatchAWS IoTentrées du journal

Chaque composant d'AWS IoT génère ses propres entrées de journal. Chaque entrée de journal a un `eventType` qui spécifie l'opération ayant provoqué la génération de l'entrée de journal. Cette section décrit les entrées de journal générées par les journaux suivantsAWS IoTComposants. Pour obtenir des informations sur AWS IoT Core pour la surveillance LoRaWan, consultezAfficher CloudWatch AWS IoT Core pour les entrées LoRaWAN (p. 1115).

Rubriques

Entrées du journal du courtier de messages

Le courtier de messages AWS IoT génère les journaux pour les événements suivants :

Rubriques

Connexion de l'entrée de journal

Le courtier de messages AWS IoT génère une entrée de journal avec un `eventType` de `Connect` lorsqu'un client MQTT se connecte.

Exemple d'entrée de journal de connexion

```
{
    "timestamp": "2017-08-10 15:37:23.476",
    "logLevel": "INFO",
    "traceId": "20b23f3f-d7f1-feae-169f-82263394fbdb",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Connect",
    "protocol": "MQTT",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `Connect` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

sourceIp

L'adresse IP d'origine de la demande.

sourcePort

Le port d'origine de la demande.

Déconnecter l'entrée du journal

Le courtier de messages AWS IoT génère une entrée de journal avec un `eventType` de `Disconnect` lorsqu'un client MQTT se déconnecte.

Exemple d'entrée de journal de déconnexion

```
{
    "timestamp": "2017-08-10 15:37:23.476",
    "logLevel": "INFO",
    "traceId": "20b23f3f-d7f1-feae-169f-82263394fbdb",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Disconnect",
    "protocol": "MQTT",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `Disconnect` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

sourceIp

L'adresse IP d'origine de la demande.

sourcePort

Le port d'origine de la demande.

disconnectReason

La raison pour laquelle le client se déconnecte.

Entrée de journal Publish-In

Lorsque le courtier de messages AWS IoT reçoit un message MQTT, il génère une entrée de journal avec un `eventType` de `Publish-In`.

Exemple d'entrée de journal Publish-In

```
{
        "timestamp": "2017-08-10 15:39:30.961",
        "logLevel": "INFO",
        "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",
        "accountId": "123456789012",
        "status": "Success",
        "eventType": "Publish-In",
        "protocol": "MQTT",
        "topicName": "$aws/things/MyThing/shadow/get",
        "clientId": "abf27092886e49a8a5c1922749736453",
        "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
        "sourceIp": "205.251.233.181",
        "sourcePort": 13490
    }
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `Publish-In` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

sourceIp

L'adresse IP d'origine de la demande.

sourcePort

Le port d'origine de la demande.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée du journal Publish-Out

Lorsque le courtier de messages publie un message MQTT, il génère une entrée de journal avec un `eventType` de `Publish-Out`

Exemple d'entrée de journal Publish-Out

```
{
    "timestamp": "2017-08-10 15:39:30.961",
    "logLevel": "INFO",
    "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Publish-Out",
    "protocol": "MQTT",
    "topicName": "$aws/things/MyThing/shadow/get",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `Publish-Out` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

sourceIp

L'adresse IP d'origine de la demande.

sourcePort

Le port d'origine de la demande.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Souscrire l'entrée de journal

Le courtier de messages AWS IoT génère une entrée de journal avec un `eventType` de `Subscribe` lorsqu'un client MQTT s'abonne à une rubrique.

Exemple d'entrée de journal d'abonnement

```
{
    "timestamp": "2017-08-10 15:39:04.413",
    "logLevel": "INFO",
    "traceId": "7aa5c38d-1b49-3753-15dc-513ce4ab9fa6",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "Subscribe",
    "protocol": "MQTT",
    "topicName": "$aws/things/MyThing/shadow/#",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "sourceIp": "205.251.233.181",
    "sourcePort": 13490
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `Subscribe` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

sourceIp

L'adresse IP d'origine de la demande.

sourcePort

Le port d'origine de la demande.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrées de journal de shadow d'appareil

Le service Device Shadow AWS IoT génère des entrées de journal pour les événements suivants :

Rubriques

- Entrée de journal DeleteThingShadow (p. 378)
- Entrée de journal GetThingShadow (p. 379)
- Entrée de journal UpdateThingShadow (p. 379)

Entrée de journal DeleteThingShadow

Le service Device Shadow génère une entrée de journal avec un `eventType` de `DeleteThingShadow` en cas de réception d'une demande de suppression d'un shadow d'appareil.

Exemple d'entrée de journal DeleteThingShadow

```
{
    "timestamp": "2017-08-07 18:47:56.664",
```

```
    "logLevel": "INFO",
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "DeleteThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "Jack",
    "topicName": "$aws/things/Jack/shadow/delete"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `DeleteThingShadow` contiennent les attributs suivants :

deviceShadowName

Nom du shadow à mettre à jour.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

Le nom de la rubrique sur laquelle la demande a été publiée.

Entrée de journal GetThingShadow

Le service Device Shadow génère une entrée de journal avec un `eventType` de `GetThingShadow` en cas de réception d'une demande get pour un shadow.

Exemple d'entrée de journal GetThingShadow

```
{
    "timestamp": "2017-08-09 17:56:30.941",
    "logLevel": "INFO",
    "traceId": "b575f19a-97a2-cf72-0ed0-c64a783a2504",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "GetThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "MyThing",
    "topicName": "$aws/things/MyThing/shadow/get"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `GetThingShadow` contiennent les attributs suivants :

deviceShadowName

Le nom du shadow demandé.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

Le nom de la rubrique sur laquelle la demande a été publiée.

Entrée de journal UpdateThingShadow

Le service Device Shadow génère une entrée de journal avec un `eventType` de `UpdateThingShadow` en cas de réception d'une demande de mise à jour d'un shadow d'appareil.

Exemple d'entrée de journal UpdateThingShadow

```
{
    "timestamp": "2017-08-07 18:43:59.436",
    "logLevel": "INFO",
    "traceId": "d0074ba8-0c4b-a400-69df-76326d414c28",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "UpdateThingShadow",
    "protocol": "MQTT",
    "deviceShadowName": "Jack",
    "topicName": "$aws/things/Jack/shadow/update"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `UpdateThingShadow` contiennent les attributs suivants :

deviceShadowName

Nom du shadow à mettre à jour.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

Le nom de la rubrique sur laquelle la demande a été publiée.

Entrées de journal du moteur de règles

Le moteur de règles AWS IoT génère des journaux pour les événements suivants :

Rubriques

- Entrée de journal FunctionExecution (p. 380)
- Entrée de journal RuleExecution (p. 381)
- Entrée de journal RuleMatch (p. 382)
- Entrée de journal RuleMessageThrottled (p. 382)
- Entrée de journal RuleNotFound (p. 383)
- Entrée de journal StartingRuleExecution (p. 384)

Entrée de journal FunctionExecution

Le moteur de règles génère une entrée de journal avec un `eventType` de `FunctionExecution` lorsque la requête SQL d'une règle appelle une fonction externe. Une fonction externe est appelée lorsqu'une action de règle effectue une demande HTTP pour AWS IoT ou un autre service Web (par exemple, en appelant `get_thing_shadow` ou `machinelearning_predict`).

Exemple d'entrée de journal FunctionExecution

```
{
    "timestamp": "2017-07-13 18:33:51.903",
    "logLevel": "DEBUG",
    "traceId": "180532b7-0cc7-057b-687a-5ca1824838f5",
    "status": "Success",
    "eventType": "FunctionExecution",
    "clientId": "N/A",
```

```
    "topicName":"rules/test",
    "ruleName": "ruleTestPredict",
    "ruleAction": "MachinelearningPredict",
    "resources": {
        "ModelId": "predict-model"
    },
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `FunctionExecution` contiennent les attributs suivants :

clientId

`N/A` pour les journaux `FunctionExecution`.

principalId

L'ID du mandataire formulant la demande.

resources

Une collection des ressources utilisées par les actions de la règle.

ruleName

Le nom de la règle de correspondance.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée de journal RuleExecution

Lorsque le moteur de règles AWS IoT déclenche l'action d'une règle, il génère une entrée de journal `RuleExecution`.

Exemple d'entrée de journal RuleExecution

```
{
    "timestamp": "2017-08-10 16:32:46.070",
    "logLevel": "INFO",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "RuleExecution",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "ruleAction": "RepublishAction",
    "resources": {
        "RepublishTopic": "rules/republish"
    },
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `RuleExecution` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

resources

Une collection des ressources utilisées par les actions de la règle.

ruleAction

Le nom de l'action déclenchée.

ruleName

Le nom de la règle de correspondance.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée de journal RuleMatch

Le moteur de règles AWS IoT génère une entrée de journal avec un `eventType` de `RuleMatch` lorsque le courtier de messages reçoit un message correspondant à une règle.

Exemple d'entrée de journal RuleMatch

```
{
    "timestamp": "2017-08-10 16:32:46.002",
    "logLevel": "INFO",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "RuleMatch",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `RuleMatch` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

ruleName

Le nom de la règle de correspondance.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée de journal RuleMessageThrottled

Lorsqu'un message est régulé, le moteur de règles AWS IoT génère une entrée de journal avec un `eventType` de `RuleMessageThrottled`.

Exemple d'entrée de journal RuleMessageThrottled

```
{
    "timestamp": "2017-10-04 19:25:46.070",
    "logLevel": "ERROR",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleMessageThrottled",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "$aws/rules/example_rule",
    "ruleName": "example_rule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "reason": "RuleExecutionThrottled",
    "details": "Message for Rule example_rule throttled"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `RuleMessageThrottled` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

détails

Une brève explication de l'erreur.

principalId

L'ID du mandataire formulant la demande.

reason

La chaîne « RuleMessageThrottled ».

ruleName

Le nom de la règle à déclencher.

topicName

Le nom de la rubrique qui a été publiée.

Entrée de journal RuleNotFound

Lorsque le moteur de règles AWS IoT ne peut pas trouver de règle avec un nom donné, il génère une entrée de journal avec un `eventType` de `RuleNotFound`.

Exemple d'entrée de journal RuleNotFound

```
{
    "timestamp": "2017-10-04 19:25:46.070",
    "logLevel": "ERROR",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleNotFound",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "$aws/rules/example_rule",
    "ruleName": "example_rule",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
    "reason": "RuleNotFound",
    "details": "Rule example_rule not found"
```

```
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `RuleNotFound` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

détails

Une brève explication de l'erreur.

principalId

L'ID du mandataire formulant la demande.

reason

La chaîne « RuleNotFound ».

ruleName

Nom de la règle qui est introuvable.

topicName

Le nom de la rubrique qui a été publiée.

Entrée de journal StartingRuleExecution

Lorsque le moteur de règles AWS IoT commence à déclencher l'action d'une règle, il génère une entrée de journal avec un `eventType` de `StartingRuleExecution`.

Exemple d'entrée de journal StartingRuleExecution

```
{
    "timestamp": "2017-08-10 16:32:46.002",
    "logLevel": "DEBUG",
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "StartingRuleExecution",
    "clientId": "abf27092886e49a8a5c1922749736453",
    "topicName": "rules/test",
    "ruleName": "JSONLogsRule",
    "ruleAction": "RepublishAction",
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `rule-` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

principalId

L'ID du mandataire formulant la demande.

ruleAction

Le nom de l'action déclenchée.

ruleName

Le nom de la règle de correspondance.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrées du journal des tâches

Le service AWS IoT Job génère des journaux pour les événements suivants. Les entrées de journal sont générées lorsqu'une demande HTTP ou MQTT est reçue à partir de l'appareil.

Rubriques

Entrée de journal DescribeJobExecution

Le service AWS IoT Jobs génère une entrée de journal avec un `eventType` de `DescribeJobExecution` lorsque le service reçoit une demande pour décrire l'exécution d'une tâche.

Exemple d'entrée de journal DescribeJobExecution

```
{
    "timestamp": "2017-08-10 19:13:22.841",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "DescribeJobExecution",
    "protocol": "MQTT",
    "clientId": "thingOne",
    "jobId": "002",
    "topicName": "$aws/things/thingOne/jobs/002/get",
    "clientToken": "myToken",
    "details": "The request status is SUCCESS."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `GetJobExecution` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

clientToken

Identifiant unique, sensible à la casse, afin de garantir l'idempotence de la requête. Pour de plus amples informations, veuillez consulter la section Comment garantir l'idempotence.

détails

Informations supplémentaires issues du service Jobs.

jobId

L'ID de tâche pour l'exécution du travail.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

La rubrique utilisée pour effectuer la demande.

Entrée de journal GetPendingJobExecution

Le service AWS IoT Jobs génère une entrée de journal avec un `eventType` de `GetPendingJobExecution` lorsque le service reçoit une demande d'exécution de tâche.

Exemple d'entrée de journal GetPendingJobExecution

```
{
    "timestamp": "2018-06-13 17:45:17.197",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "GetPendingJobExecution",
    "protocol": "MQTT",
    "clientId": "299966ad-54de-40b4-99d3-4fc8b52da0c5",
    "topicName": "$aws/things/299966ad-54de-40b4-99d3-4fc8b52da0c5/jobs/get",
    "clientToken": "24b9a741-15a7-44fc-bd3c-1ff2e34e5e82",
    "details": "The request status is SUCCESS."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `GetPendingJobExecution` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

clientToken

Identifiant unique, sensible à la casse, afin de garantir l'idempotence de la demande. Pour de plus amples informations, veuillez consulter la section Comment garantir l'idempotence.

détails

Informations supplémentaires issues du service Jobs.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée de journal ReportFinalJobExecutionCount

La .AWS IoTLe service Jobs génère une entrée de journal avec un`entryType`de`ReportFinalJobExecutionCount`lorsqu'une tâche est terminée.

Exemple d'entrée de journal ReportFinalJobExecutionCount

```
{
    "timestamp": "2017-08-10 19:44:16.776",
    "logLevel": "INFO",
```

```
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "ReportFinalJobExecutionCount",
    "jobId": "002",
    "details": "Job 002 completed. QUEUED job execution count: 0 IN_PROGRESS job execution
 count: 0 FAILED job execution count: 0 SUCCEEDED job execution count: 1 CANCELED job
 execution count: 0 REJECTED job execution count: 0 REMOVED job execution count: 0"
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `ReportFinalJobExecutionCount` contiennent les attributs suivants :

détails

Informations supplémentaires issues du service Jobs.

jobId

L'ID de tâche pour l'exécution du travail.

Entrée de journal StartNextPendingJobExecution

Lorsqu'il reçoit une demande de démarrage de la prochaine exécution du travail en attente, le service AWS IoT Jobs génère une entrée de journal avec un `eventType` de `StartNextPendingJobExecution`.

Exemple d'entrée de journal StartNextPendingJobExecution

```
{
    "timestamp": "2018-06-13 17:49:51.036",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "StartNextPendingJobExecution",
    "protocol": "MQTT",
    "clientId": "95c47808-b1ca-4794-bc68-a588d6d9216c",
    "topicName": "$aws/things/95c47808-b1ca-4794-bc68-a588d6d9216c/jobs/start-next",
    "clientToken": "bd7447c4-3a05-49f4-8517-dd89b2c68d94",
    "details": "The request status is SUCCESS."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `StartNextPendingJobExecution` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

clientToken

Identifiant unique, sensible à la casse, afin de garantir l'idempotence de la demande. Pour de plus amples informations, veuillez consulter la section Comment garantir l'idempotence.

détails

Informations supplémentaires issues du service Jobs.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

La rubrique utilisée pour effectuer la demande.

Entrée de journal UpdateJobExecution

Le service AWS IoT Jobs génère une entrée de journal avec un `eventType` de `UpdateJobExecution` lorsque le service reçoit une requête de mise à jour de l'exécution d'une tâche.

Exemple d'entrée de journal UpdateJobExecution

```
{
    "timestamp": "2017-08-10 19:25:14.758",
    "logLevel": "DEBUG",
    "accountId": "123456789012",
    "status": "Success",
    "eventType": "UpdateJobExecution",
    "protocol": "MQTT",
    "clientId": "thingOne",
    "jobId": "002",
    "topicName": "$aws/things/thingOne/jobs/002/update",
    "clientToken": "myClientToken",
    "versionNumber": "1",
    "details": "The destination status is IN_PROGRESS. The request status is SUCCESS."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `UpdateJobExecution` contiennent les attributs suivants :

clientId

L'ID du client formulant la demande.

clientToken

Identifiant unique, sensible à la casse, afin de garantir l'idempotence de la demande. Pour de plus amples informations, veuillez consulter la section Comment garantir l'idempotence.

détails

Informations supplémentaires issues du service Jobs.

jobId

L'ID de tâche pour l'exécution du travail.

protocole ;

Le protocole utilisé lors de la création de la demande. Les valeurs valides sont `MQTT` ou `HTTP`.

topicName

La rubrique utilisée pour effectuer la demande.

versionNumber

Version de l'exécution de tâche.

Entrées de journal de provisionnement des appareils

Le service de mise en service d'appareils AWS IoT génère des journaux pour les événements suivants :

Rubriques

- Entrée de journal GetDeviceCredentials (p. 389)
- Entrée de journal ProvisionDevice (p. 389)

Entrée de journal GetDeviceCredentials

Le service de mise en service d'appareils AWS IoT génère une entrée de journal avec un `eventType` de `GetDeviceCredential` lorsqu'un client appelle `GetDeviceCredential`.

Exemple d'entrée de journal GetDeviceCredentials

```
{
  "timestamp" : "2019-02-20 20:31:22.932",
  "logLevel" : "INFO",
  "traceId" : "8d9c016f-6cc7-441e-8909-7ee3d5563405",
  "accountId" : "123456789101",
  "status" : "Success",
  "eventType" : "GetDeviceCredentials",
  "deviceCertificateId" :
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
  "details" : "Additional details about this log."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `GetDeviceCredentials` contiennent les attributs suivants :

détails

Une brève explication de l'erreur.

deviceCertificateId

ID du certificat d'appareil.

Entrée de journal ProvisionDevice

Le service de mise en service d'appareils AWS IoT génère une entrée de journal avec un `eventType` de `ProvisionDevice` lorsqu'un client appelle `ProvisionDevice`.

Exemple d'entrée de journal ProvisionDevice

```
{
  "timestamp" : "2019-02-20 20:31:22.932",
  "logLevel" : "INFO",
  "traceId" : "8d9c016f-6cc7-441e-8909-7ee3d5563405",
  "accountId" : "123456789101",
  "status" : "Success",
  "eventType" : "ProvisionDevice",
  "provisioningTemplateName" : "myTemplate",
  "deviceCertificateId" :
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
  "details" : "Additional details about this log."
 }
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `ProvisionDevice` contiennent les attributs suivants :

détails

Une brève explication de l'erreur.

deviceCertificateId

ID du certificat d'appareil.

provisioningTemplateName

Nom du modèle de mise en service.

Entrées dynamiques du journal des groupes d'objets

Les groupes d’objets dynamiquesAWS IoT génèrent des journaux pour l'événement suivant.

Rubriques

- Entrée de journal AddThingToDynamicThingGroupsFailed (p. 390)

Entrée de journal AddThingToDynamicThingGroupsFailed

Lorsque AWS IoT a pas été en mesure d'ajouter un objet aux groupes dynamiques spécifiés, il génère une entrée de journal avec un `eventType` de `AddThingToDynamicThingGroupsFailed`. Cela se produit lorsqu'un objet a répondu aux critères d'appartenance au groupe d’objets dynamiques. Toutefois, il n'a pas pu être ajouté au groupe dynamique ou il a été supprimé du groupe dynamique. Cela peut se produire pour les raisons suivantes :

- L’objet appartient déjà au nombre maximal de groupes.
- L'option --override-dynamic-groups a été utilisée pour ajouter l’objet à un groupe d’objets statiques. Il a été retiré d'un groupe d’objets dynamiques pour rendre cela possible.

Pour de plus amples informations, veuillez consulter Limitations et conflits de groupes d’objets dynamiques (p. 223).

Exemple d'entrée de journal AddThingToDynamicThingGroupsFailed

Cet exemple montre une entrée de journal correspondant à une erreur `AddThingToDynamicThingGroupsFailed`. Dans cet exemple, TestThing répondait aux critères d'appartenance aux groupes d’objets dynamiques répertoriés dans `dynamicThingGroupNames`, mais n'a pas pu être ajouté à ces groupes dynamiques, comme décrit dans `reason`.

```
{
 "timestamp": "2020-03-16 22:24:43.804",
 "logLevel": "ERROR",
 "traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",
 "accountId": "571032923833",
 "status": "Failure",
 "eventType": "AddThingToDynamicThingGroupsFailed",
 "thingName": "TestThing",
 "dynamicThingGroupNames": [
  "DynamicThingGroup11",
  "DynamicThingGroup12",
  "DynamicThingGroup13",
  "DynamicThingGroup14"
 ],
 "reason": "The thing failed to be added to the given dynamic thing group(s) because the
 thing already belongs to the maximum allowed number of groups."
}
```

En plus de Attributs CloudWatch Logs (p. 391), les entrées de journal `AddThingToDynamicThingGroupsFailed` contiennent les attributs suivants :

dynamicThingGroupNames

Tableau des groupes d’objets dynamiques auquel l’objet n'a pas pu être ajouté.

reason

Raison pour laquelle l'objet n'a pas pu être ajouté aux groupes d'objets dynamiques.

thingName

Nom de l'objet qui n'a pas pu être ajouté à un groupe d'objets dynamiques.

Attributs CloudWatch Logs

Toutes les entrées de journal CloudWatch Logs incluent les attributs suivants :

accountId

Votre compte Compte AWS ID.

eventType

Le type d'événement pour lequel le journal a été créé. La valeur du type d'événement dépend de l'événement qui a généré l'entrée de journal. Chaque description d'entrée de journal inclut la valeur de `eventType` pour cette entrée de journal.

logLevel

Le niveau de journalisation utilisé. Pour plus d'informations, consultez the section called "Niveaux de journalisation" (p. 358).

status

Le statut d'une demande.

timestamp

L'horodatage UNIX du moment où le client s'est connecté au courtier de messages AWS IoT.

traceId

Un identifiant généré de façon aléatoire qui peut être utilisé pour mettre en corrélation tous les journaux pour une demande spécifique.

Journalisez les appels d'API AWS IoT via AWS CloudTrail.

AWS IoT est intégré à AWS CloudTrail, un service qui enregistre les actions effectuées par un utilisateur, un rôle ou un service AWS dans AWS IoT. CloudTrail capture tous les appels d'API pourAWS IoTEn tant qu'événements, y compris les appels à partir deAWS IoTet à partir d'appels de code vers la consoleAWS IoTAPI. Si vous créez un journal de suivi, vous pouvez diffuser en continu les événements CloudTrail dans un compartiment Amazon S3, y compris les événements pourAWS IoT. Si vous ne configurez pas de journal de suivi, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Event history (Historique des événements). En utilisant les informations collectées par CloudTrail, vous pouvez déterminer quelle demande a été faite àAWS IoT, l'adresse IP à partir de laquelle la demande a été faite, qui a effectué la demande, quand elle a eu lieu et autres informations supplémentaires.

Pour en savoir plus sur CloudTrail, consultez le AWS CloudTrail Guide de l'utilisateur .

AWS IoTInformations dans CloudTrail

CloudTrail est activé sur votre Compte AWS lorsque vous créez le compte. Lorsque l'activité se produit dansAWS IoT, cette activité est enregistrée dans un événement CloudTrail avec d'autresAWSévénements

de service dansHistorique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS . Pour de plus amples informations, veuillez consulter Affichage des événements avec l'historique des événements CloudTrail.

Pour un registre permanent des événements dans votre Compte AWS , y compris des événements pourAWS IoT, créez un journal d'activité. Un journal de suivi permet à CloudTrail de livrer des fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les Région AWS Les Le sentier enregistre les événements de tous les Région AWS s dans la sectionAWSet livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. Vous pouvez configurer d'autresAWSPour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour plus d'informations, consultez :

- Présentation de la création d'un journal de suivi
- Intégrations et services pris en charge par CloudTrail
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers journaux CloudTrail de plusieurs régions et Réception de fichiers journaux CloudTrail de plusieurs comptes

> **Note**
>
> AWS IoTLes actions de plan de données (côté appareil) ne sont pas consignées par CloudTrail. Utilisez CloudWatch pour surveiller ces actions.

D'une manière générale,AWS IoTLes actions de plan de contrôle qui apportent des modifications sont consignées par CloudTrail. Appels tels queCreateThing,CreateKeysAndCertificate, etUpdateCertificatelaisser les entrées CloudTrail, tandis que les appels tels queListThingsandListTopicRulesne le font pas.

Chaque événement ou entrée du journal contient des informations sur la personne qui a généré la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les informations d'identification utilisateur racine ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, consultez l'élément userIdentity CloudTrail.

AWS IoTsont répertoriées dans le documentAWS IoTAPI Reference.AWS IoT Les actions sans fil sont répertoriées dans le documentAWS IoTRéférence d'API sans fil.

Présentation des entrées des fichiers journaux AWS IoT

Un journal de suivi est une configuration qui permet la livraison d'événements sous forme de fichiers journaux vers un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail peuvent contenir une ou plusieurs entrées de journal. Un événement représente une demande individuelle émise à partir d'une source quelconque et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. Les fichiers journaux CloudTrail ne constituent pas une trace de pile ordonnée d'appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'action `AttachPolicy`.

```
{
    "timestamp":"1460159496",
    "AdditionalEventData":"",
    "Annotation":"",
    "ApiVersion":"",
    "ErrorCode":"",
    "ErrorMessage":"",
    "EventID":"8bff4fed-c229-4d2d-8264-4ab28a487505",
    "EventName":"AttachPolicy",
    "EventTime":"2016-04-08T23:51:36Z",
    "EventType":"AwsApiCall",
    "ReadOnly":"",
    "RecipientAccountList":"",
    "RequestID":"d4875df2-fde4-11e5-b829-23bf9b56cbcd",
    "RequestParamters":{
        "principal":"arn:aws:iot:us-
east-1:123456789012:cert/528ce36e8047f6a75ee51ab7beddb4eb268ad41d2ea881a10b67e8e76924d894",
        "policyName":"ExamplePolicyForIoT"
    },
    "Resources":"",
    "ResponseElements":"",
    "SourceIpAddress":"52.90.213.26",
    "UserAgent":"aws-internal/3",
    "UserIdentity":{
        "type":"AssumedRole",
        "principalId":"AKIAI44QH8DHBEXAMPLE",
        "arn":"arn:aws:sts::12345678912:assumed-role/iotmonitor-us-east-1-beta-
InstanceRole-1C5T1YCYMHPYT/i-35d0a4b6",
        "accountId":"222222222222",
        "accessKeyId":"access-key-id",
        "sessionContext":{
            "attributes":{
                "mfaAuthenticated":"false",
                "creationDate":"Fri Apr 08 23:51:10 UTC 2016"
            },
            "sessionIssuer":{
                "type":"Role",
                "principalId":"AKIAI44QH8DHBEXAMPLE",
                "arn":"arn:aws:iam::123456789012:role/executionServiceEC2Role/iotmonitor-
us-east-1-beta-InstanceRole-1C5T1YCYMHPYT",
                "accountId":"222222222222",
                "userName":"iotmonitor-us-east-1-InstanceRole-1C5T1YCYMHPYT"
            }
        },
        "invokedBy":{
            "serviceAccountId":"111111111111"
        }
    },
    "VpcEndpointId":""
}
```

Règles pour AWS IoT

Les règles offrent à vos dispositifs la possibilité d'interagir avec les services AWS. Les règles sont analysées et les actions exécutées en fonction du flux de rubrique MQTT. Vous pouvez utiliser des règles pour prendre en charge des tâches telles que celles-ci :

- Augmenter ou filtrer les données reçues d'un appareil.
- Écrire des données provenant d'un appareil dans une base de données Amazon DynamoDB.
- Enregistrez un fichier dans Amazon S3.
- Envoyer une notification push à tous les utilisateurs à l'aide d'Amazon SNS.
- Publie des données dans une file d'attente Amazon SQS.
- Appeler une fonction Lambda pour extraire des données.
- Traiter les messages provenant d'un grand nombre d'appareils à l'aide de Amazon Kinesis.
- Envoyez les données au Service Amazon Elasticsearch Service.
- Capture une métrique CloudWatch.
- Modifier une alarme CloudWatch.
- Envoyer les données à partir d'un message MQTT à Amazon Machine Learning pour faire des prévisions basées sur un modèle Amazon ML.
- Envoyer un message à un flux d'entrée Salesforce IoT
- Envoyer des données de message à un canal AWS IoT Analytics.
- Lancer l'exécution d'une machine d'état Step Functions.
- Envoyer des données de message à une entrée AWS IoT Events.
- Envoyer des données de message à une propriété de ressources dans AWS IoT SiteWise.
- Envoyer des données de message à une application web ou à un service.

Vos règles peuvent utiliser des messages MQTT qui passent par le protocole de publication/d'abonnement pris en charge par lethe section called “Protocoles de communication de périphérique” (p. 79)ou, en utilisant leBasic Ingest (p. 480), vous pouvez envoyer en toute sécurité des données de périphérique à laAWSservices énumérés ci-dessus sans encourirCoûts de messagerie. (LeBasic Ingest (p. 480)optimise le flux de données en enlevant l'agent de messages de publication/abonnement du chemin d'intégration, ce qui le rend plus économique, tout en préservant les fonctions de sécurité et de traitement de données deAWS IoT.)

AvantAWS IoTpeut effectuer ces actions, vous devez lui accorder l'autorisation d'accéder à votreAWSRessources en votre nom. Lorsque les actions sont effectuées, vous payez les frais standards pour les services AWS que vous utilisez.

Table des matières

Attribuer à AWS IoT l'accès requis

Vous utilisez les rôles IAM pour contrôler leAWSà laquelle chaque règle a accès. Avant de créer une règle, vous devez créer un rôle IAM avec une stratégie qui autorise l'accès auxAWSAWS.AWS IoTassume ce rôle lors de l'exécution d'une règle.

Pour créer un rôle IAM (AWS CLI)

1. Enregistrez le document de stratégie d'approbation suivant, qui accorde à AWS IoT l'autorisation d'assumer le rôle, dans un fichier appelé `iot-role-trust.json` :

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

   Utilisez la commande create-role pour créer un rôle IAM spécifiant le fichier `iot-role-trust.json` :

```
aws iam create-role --role-name my-iot-role --assume-role-policy-document file://iot-role-trust.json
```

   La sortie de cette commande ressemble à ce qui suit :

```
{
  "Role": {
    "AssumeRolePolicyDocument": "url-encoded-json",
    "RoleId": "AKIAIOSFODNN7EXAMPLE",
    "CreateDate": "2015-09-30T18:43:32.821Z",
    "RoleName": "my-iot-role",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:role/my-iot-role"
  }
}
```

2. Enregistrez le code JSON suivant dans un fichier nommé `my-iot-policy.json`.

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

   Ce JSON est un exemple de document de stratégie qui accordeAWS IoTaccès administrateur à DynamoDB.

Utilisez la commande create-policy pour accorder à AWS IoT l'accès aux ressources AWS en assumant le rôle, en transmettant le fichier `my-iot-policy.json` :

```
aws iam create-policy --policy-name my-iot-policy --policy-document file://my-iot-policy.json
```

Pour plus d'informations sur la façon d'accorder l'accès aux services AWS dans les stratégies pour AWS IoT, consultez Création d'une règle AWS IoT (p. 397).

La sortie de la commande create-policy contient l'ARN de la stratégie. Vous devez attacher la stratégie à un rôle.

```
{
  "Policy": {
    "PolicyName": "my-iot-policy",
    "CreateDate": "2015-09-30T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/my-iot-policy",
    "UpdateDate": "2015-09-30T19:31:18.620Z"
  }
}
```

3. Utilisez la commande attach-role-policy pour attacher la stratégie à votre rôle :

```
aws iam attach-role-policy --role-name my-iot-role --policy-arn
 "arn:aws:iam::123456789012:policy/my-iot-policy"
```

Transmettre les autorisations de rôle

Une définition de règle comporte un rôle IAM qui accorde l'autorisation d'accéder aux ressources spécifiées dans l'action de la règle. Le moteur de règles endosse ce rôle lors du déclenchement de l'action de cette règle. Le rôle doit être défini dans le même Compte AWS En règle générale.

Lorsque vous créez ou remplacez une règle, vous transférez en fait un rôle au moteur de règles. L'utilisateur qui effectue cette opération nécessite l'autorisation `iam:PassRole`. Pour vous assurer que vous disposez de cette autorisation, créez une stratégie qui accorde à l'autorisation`iam:PassRole`et attachez à votre utilisateur IAM. La stratégie suivante montre comment accorder l'autorisation `iam:PassRole` pour un rôle.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::123456789012:role/myRole"
      ]
    }
  ]
```

```
}
```

Dans cet exemple de stratégie, l'autorisation `iam:PassRole` est accordée pour le rôle `myRole`. Le rôle a été spécifié à l'aide de son ARN. Vous devez attacher cette stratégie à votre utilisateur IAM ou au rôle auquel appartient votre utilisateur. Pour plus d'informations, consultez Utilisation des stratégies gérées.

> **Note**
>
> Les fonctions Lambda utilisent des stratégies basées sur les ressources, où la stratégie est attachée directement à la fonction Lambda même. Lorsque vous créez une règle qui appelle une fonction Lambda, vous ne transmettez pas de rôle, et l'utilisateur qui crée la règle n'a donc pas besoin de l'option`iam:PassRole`Autorisation. Pour plus d'informations sur l'autorisation de la fonction Lambda, consultezAttribution d'autorisations à l'aide d'une stratégie.

Création d'une règle AWS IoT

Vous configurez les règles d'acheminement des données provenant de vos objets connectés. Règles se composent de ce qui suit :

Nom de la règle

Nom de la règle.

> **Note**
>
> Il est déconseillé d'utiliser des informations personnelles identifiables dans le nom de vos règles.

Description facultative

Description textuelle de la règle.

> **Note**
>
> Il est déconseillé d'utiliser des informations personnelles identifiables dans les descriptions de vos règles.

Instruction SQL

Syntaxe SQL simplifiée pour filtrer les messages reçus dans une rubrique MQTT et pousser les données ailleurs. Pour plus d'informations, consultez Référence SQL AWS IoT (p. 482).

Version de SQL

Version du moteur de règles SQL à utiliser lors de l'évaluation de la règle. Même si cette propriété est facultative, nous vous recommandons vivement de préciser la version de SQL. Si cette propriété n'est pas définie, la valeur par défaut, `2015-10-08`, est utilisée. Pour plus d'informations, consultez Versions de SQL (p. 545).

Une ou plusieurs actions

Actions effectuées par AWS IoT lors de l'exécution de la règle. Par exemple, vous pouvez insérer des données dans une table DynamoDB, écrire des données dans un compartiment Amazon S3, publier dans une rubrique Amazon SNS ou appeler une fonction Lambda.

Une action d'erreur

L'action AWS IoT s'exécute lorsqu'elle n'est pas en mesure d'effectuer une action de règle.

Lorsque vous créez une règle, soyez conscient de la quantité de données que vous publiez dans des rubriques. Si vous créez des règles qui incluent un modèle de rubrique de caractère générique, elles

peuvent correspondre à un pourcentage élevé de vos messages, et vous devrez peut-être augmenter la capacité des ressources AWS utilisées par les actions cibles. En outre, si vous créez une règle de republication qui inclut un modèle de rubrique de caractère générique, vous pouvez vous retrouver avec une règle circulaire qui tourne en boucle à l'infini.

> **Note**
>
> La création et la mise à jour de règles sont des actions de niveau administrateur. Tout utilisateur détenant des autorisations de création ou de mise à jour de règles peut accéder aux données traitées par les règles.

Pour créer une règle (AWS CLI)

Utilisez la commande create-topic-rule pour créer une règle :

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

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à l'`iot/test`dans la table DynamoDB spécifiée. L'instruction SQL filtre les messages et le rôle accordé à l'ARNAWS IoTAutorisation d'écrire dans la table DynamoDB.

```
{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "dynamoDB": {
          "tableName": "my-dynamodb-table",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role",
          "hashKeyField": "topic",
          "hashKeyValue": "${topic(2)}",
          "rangeKeyField": "timestamp",
          "rangeKeyValue": "${timestamp()}"
      }
  }]
}
```

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à la rubrique `iot/test` dans le compartiment S3 spécifié. L'instruction SQL filtre les messages et le rôle ARN accorde à l'ARNAWS IoTAutorisation d'écrire dans le compartiment Amazon S3.

```
{
  "awsIotSqlVersion": "2016-03-23",
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "actions": [
    {
      "s3": {
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",
        "bucketName": "my-bucket",
        "key": "myS3Key"
      }
    }
  ]
}
```

Voici un exemple de fichier de charge utile avec une règle qui publie des données dans Amazon Elasticsearch Service :

```
{
```

```
  "sql":"SELECT *, timestamp() as timestamp FROM 'iot/test'",
  "ruleDisabled":false,
  "awsIotSqlVersion": "2016-03-23",
  "actions":[
    {
      "elasticsearch":{
        "roleArn":"arn:aws:iam::123456789012:role/aws_iot_es",
        "endpoint":"https://my-endpoint",
        "index":"my-index",
        "type":"my-type",
        "id":"${newuuid()}"
      }
    }
  ]
}
```

Voici un exemple de fichier de charge utile avec une règle qui appelle une fonction Lambda :

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "lambda": {
          "functionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-lambda-
function"
      }
  }]
}
```

Voici un exemple de fichier de charge utile avec une règle qui publie dans une rubrique Amazon SNS :

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "sns": {
          "targetArn": "arn:aws:sns:us-west-2:123456789012:my-sns-topic",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Voici un exemple de fichier de charge utile avec une règle qui permet de republier dans une autre rubrique MQTT :

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "republish": {
          "topic": "my-mqtt-topic",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Voici un exemple de fichier de charge utile avec une règle qui publie des données dans un flux Amazon Kinesis Data Firehose :

```
{
  "sql": "SELECT * FROM 'my-topic'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "firehose": {
          "roleArn": ""arn:aws:iam::123456789012:role/my-iot-role",
          "deliveryStreamName": "my-stream-name"
      }
  }]
}
```

Voici un exemple de fichier de charge utile avec une règle qui utilise l'Amazon Machine Learning`machinelearning_predict`Pour republier dans une rubrique, si les données dans la charge utile MQTT sont classées comme un 1.

```
{
  "sql": "SELECT * FROM 'iot/test' where machinelearning_predict('my-model',
 'arn:aws:iam::123456789012:role/my-iot-aml-role', *).predictedLabel=1",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
    "republish": {
        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role",
        "topic": "my-mqtt-topic"
    }
  }]
}
```

Voici un exemple de fichier de charge utile assorti d'une règle qui publie des messages dans un flux d'entrée Salesforce IoT Cloud.

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "salesforce": {
          "token": "ABCDEFGHI123456789abcdefghi123456789",
          "url": "https://ingestion-cluster-id.my-env.sfdcnow.com/streams/stream-id/
connection-id/my-event"
      }
  }]
}
```

L'exemple suivant illustre un fichier de charge utile avec une règle lançant l'exécution d'une machine d'état Step Functions.

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [{
      "stepFunctions": {
          "stateMachineName": "myCoolStateMachine",
          "executionNamePrefix": "coolRunning",
          "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
  }]
}
```

Affichage des règles

Utilisez la commande list-topic-rules pour répertorier vos règles :

```
aws iot list-topic-rules
```

Utilisez la commande get-topic-rule pour obtenir des informations sur une règle :

```
aws iot get-topic-rule --rule-name myrule
```

Suppression d'une règle

Lorsque vous avez terminé avec une règle, vous pouvez la supprimer.

Pour supprimer une règle (AWS CLI)

Utilisez la commande delete-topic-rule pour supprimer une règle :

```
aws iot delete-topic-rule --rule-name myrule
```

Actions de règle AWS IoT

AWS IoTLes actions de règle spécifient que faire lorsqu'une règle est déclenchée. Vous pouvez définir des actions pour envoyer des données à une base de données Amazon DynamoDB, envoyer des données à Amazon Kinesis Data Streams, appeler unAWS Lambda, et ainsi de suite.AWS IoTprend en charge les actions suivantes dans Région AWS s où le service de l'action est disponible.

Action de la règle	Description	Nom de l'API
Apache Kafka (p. 403)	Envoie un message à un cluster Apache Kafka.	`kafka`
Alarmes CloudWatch (p. 408)	Change l'état d'une alarme Amazon CloudWatch.	`cloudwatchAlarm`
CloudWatch Logs (p. 410)	Envoie un message à Amazon CloudWatch Logs.	`cloudwatchLogs`
Métriques CloudWatch (p. 411)	Envoie un message à une métrique CloudWatch.	`cloudwatchMetric`
DynamoDB (p. 412)	Envoie un message à une table DynamoDB.	`dynamoDB`
DynamoDBv2 (p. 415)	Envoie des données de message à plusieurs colonnes dans une table DynamoDB.	`dynamoDBv2`
Elasticsearch (p. 416)	Envoie un message à un point de terminaison Amazon Elasticsearch Service.	`elasticsearch`

Action de la règle	Description	Nom de l'API
HTTPS (p. 418)	Publie un message sur un point de terminaison HTTPS.	`http`
IoT Analytics (p. 420)	Envoie un message à unAWS IoT Analyticscanal	`iotAnalytics`
IoT Events (p. 422)	Envoie un message à unAWS IoT EventsEntrée	`iotEvents`
IoT SiteWise (p. 423)	Envoie les données de message àAWS IoT SiteWisepropriétés des actifs.	`iotSiteWise`
Kinesis Data Firehose (p. 427)	Envoie un message à un flux de diffusion Kinesis Data Firehose.	`firehose`
Kinesis Data Streams (p. 429)	Envoie un message à un flux de données Kinesis.	`kinesis`
Lambda (p. 431)	Invoque une fonction Lambda avec des données de message en entrée.	`lambda`
Republish (p. 433)	Republie un message dans une autre rubrique MQTT.	`republish`
S3 (p. 434)	Stocke un message dans un compartiment Amazon Simple Storage Service (Amazon S3).	`s3`
Salesforce IoT (p. 436)	Envoyer un message à un flux d'entrée Salesforce IoT	`salesforce`
SNS (p. 437)	Publication d'un message sous forme de notification push Amazon Simple Notification Service (Amazon SNS).	`sns`
SQS (p. 438)	Envoie un message à une file d'attente Amazon Simple Queue Service (Amazon SQS).	`sqs`
Step Functions (p. 440)	Commence parAWS Step FunctionsMachine d'état.	`stepFunctions`
the section called “Timestream” (p. 441)	Envoie un message à une table de base de données Amazon Timestream.	`timestream`
the section called “VPC” (p. 446)	Envoie des données à un Amazon Virtual Private Cloud (Amazon VPC).	`VPC`

Notes

- Vous devez définir la règle dans le même Région AWS comme ressource d'un autre service, afin que l'action de règle puisse interagir avec cette ressource.

- Le moteur de règles AWS IoT peut effectuer plusieurs tentatives pour exécuter une action en cas d'erreurs intermittentes. Si toutes les tentatives échouent, le message est ignoré et l'erreur est disponible dans vos journaux CloudWatch. Vous pouvez spécifier une action en cas d'erreur pour chaque règle appelée après une défaillance. Pour plus d'informations, consultez Gestion des erreurs (action d'erreur) (p. 452).
- Certaines actions de règle déclenchent des actions dans les services qui s'intègrent avec AWS Key Management Service (AWS KMS) pour prendre en charge le chiffrement des données au repos. Si vous utilisez unAWS KMSLa clé principale client (CMK) client pour chiffrer les données au repos, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Consultez les rubriques relatives au chiffrement des données dans le guide du service approprié pour savoir comment gérer les autorisations de votre CMK gérée par le client. Pour plus d'informations sur les clés CMK et les clés CMK gérées par le client, consultezAWS Key Management ServiceConceptsdans leAWS Key Management ServiceManuel du développeur.

Apache Kafka

L'action Apache Kafka (Kafka) envoie des messages directement à vos clusters Amazon Managed Streaming for Apache Kafka (Amazon MSK) ou Apache Kafka autogérés pour l'analyse et la visualisation des données.

> Note
>
> Ce sujet suppose que vous connaissiez la plateforme Apache Kafka et les concepts connexes. Pour plus d'informations sur Apache Kafka, consultezApache Kafka.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`ec2:CreateNetworkInterface`,`ec2:DescribeNetworkInterfaces`,`ec2:CreateNetworkInterfacePerm` et`ec2:DescribeVpcAttribute`. Ce rôle crée et gère des interfaces réseau élastiques vers votre Amazon Virtual Private Cloud pour atteindre votre courtier Kafka. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriser AWS IoT Core Pour effectuer cette action de règle.

  Pour plus d'informations sur les interfaces réseau, consultezInterfaces réseau ElasticDans le Guide de l'utilisateur Amazon EC2.

  La stratégie attachée au rôle que vous spécifiez doit se présenter comme suit.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:DescribeNetworkInterfaces",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DescribeVpcAttribute"
            ],
```

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

- Si vous utilisezAWS Secrets ManagerPour stocker les informations d'identification requises pour se connecter à votre agent Kafka, vous devez créer un rôle IAM qui AWS IoT Core peut supposer d'effectuer le`secretsmanager:GetSecretValue`and`secretsmanager:DescribeSecret`.

  La stratégie attachée au rôle que vous spécifiez doit se présenter comme suit.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue",
            "secretsmanager:DescribeSecret"
            ],
            "Resource": [

 "arn:aws:secretsmanager:region:123456789012:secret:kafka_client_truststore-*",
                "arn:aws:secretsmanager:region:123456789012:secret:kafka_keytab-*"
            ]
        }
    ]
}
```

- Vous devez créer une destination de cloud privé virtuel (VPC). (Vous pouvez exécuter vos clusters Apache Kafka dans Amazon Virtual Private Cloud.) La .AWS IoTcrée une interface réseau dans chacun des sous-réseaux répertoriés dans la destination du VPC. Cela permet au moteur de règles d'acheminer le trafic directement vers le VPC. Lorsque vous créez une destination VPC, leAWS IoTcrée automatiquement une action de règle VPC. Pour plus d'informations sur les actions de règle VPC, consultezthe section called “VPC” (p. 446).
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données au repos, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterCryptage Amazon MSKdans leAmazon Managed Streaming for Apache Kafka Guide du développeur.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

DestinationArn

Amazon Resource Name (ARN) de la destination VPC. Pour plus d'informations sur la création d'une destination VPC, consultezthe section called “VPC” (p. 446).

topic

Le sujet Kafka pour les messages à envoyer au courtier Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, consultez the section called “Modèles de substitution” (p. 541).

touche (facultatif)

La clé de message Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, consultez the section called “Modèles de substitution” (p. 541).

partition (facultatif)

La partition de message Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, consultez the section called “Modèles de substitution” (p. 541).

ClientPropriétés

Objet qui définit les propriétés du client producteur Apache Kafka.

acks (optionnel)

Nombre d'accusés de réception que le producteur exige que le serveur ait reçu avant de considérer une demande terminée.

Si vous spécifiez 0 comme valeur, le producteur n'attendra aucun accusé de réception du serveur. Si le serveur ne reçoit pas le message, le producteur ne réessaiera pas d'envoyer le message.

Valeurs valides : 0, 1. La valeur par défaut est 1.

bootstrap.servers

Une liste de paires d'hôtes et de ports (`host1:port1,host2:port2`, etc.) utilisé pour établir la connexion initiale à votre cluster Kafka.

compression .type (facultatif)

Type de compression pour toutes les données générées par le producteur.

Valeurs valides : `none`,`gzip`,`snappy`,`lz4`,`zstd`. La valeur par défaut est .`none`.

securité.protocol

Le protocole de sécurité utilisé pour attacher à votre courtier Kafka.

Valeurs valides : `SSL`,`SASL_SSL`. La valeur par défaut est .`SSL`.

key.sérialiseur

Spécifie comment activer les objets clés que vous fournissez avec l'option`ProducerRecord`en octets.

Valeurs valides :`StringSerializer`.

value.sérialiseur

Spécifie comment transformer les objets de valeur que vous fournissez avec le paramètre`ProducerRecord`en octets.

Valeurs valides :`ByteBufferSerializer`.

ssl.truststore

Le fichier truststore au format base64 ou l'emplacement du fichier truststore dansAWS Secrets Manager. Cette valeur n'est pas requise si votre magasin de confiance est approuvé par les autorités de certification Amazon.

Ce champ prend en charge les modèles de substitution. Si vous utilisez Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka, vous pouvez utiliser la fonction SQL get_secret pour récupérer la valeur de ce champ. Pour de plus amples informations sur les modèles de substitution, veuillez consulter the

section called “Modèles de substitution” (p. 541). Pour plus d'informations sur la fonction SQL get_secret, consultezthe section called “get_secret (secretId, SecretType, key, roleArn) ” (p. 511). Si le magasin de fichiers de confiance est sous la forme d'un fichier, utilisez la méthode`SecretBinary` Paramètre . Si le magasin de confiance est sous la forme d'une chaîne, utilisez la méthode`SecretString` Paramètre .

La taille maximale de cette valeur est de 65 Ko.

ssl.truststore.password

Mot de passe pour le magasin de confiance. Cette valeur n'est requise que si vous avez créé un mot de passe pour le magasin de confiance.

ssl.keystore

Le fichier keystore. Cette valeur est requise lorsque vous spécifiez`SSL`comme valeur pour`security.protocol`.

Ce champ prend en charge les modèles de substitution. Vous devez utiliser Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka. Utilisez la fonction SQL get_secret pour récupérer la valeur de ce champ. Pour de plus amples informations sur les modèles de substitution, veuillez consulter the section called “Modèles de substitution” (p. 541). Pour plus d'informations sur la fonction SQL get_secret, consultezthe section called “get_secret (secretId, SecretType, key, roleArn) ” (p. 511). Utilisation de l'`SecretBinary` Paramètre .

ssl.keystore.password

Mot de passe du magasin de clés. Cette valeur est requise si vous spécifiez une valeur pour`ssl.keystore`.

La valeur de ce champ peut être du texte brut. Ce champ prend également en charge les modèles de substitution. Vous devez utiliser Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka. Utilisez la fonction SQL get_secret pour récupérer la valeur de ce champ. Pour de plus amples informations sur les modèles de substitution, veuillez consulter the section called “Modèles de substitution” (p. 541). Pour plus d'informations sur la fonction SQL get_secret, consultezthe section called “get_secret (secretId, SecretType, key, roleArn) ” (p. 511).Utilisation de l'`SecretString` Paramètre .

ssl.key.password

Le mot de passe de la clé privée dans votre fichier keystore.

Ce champ prend en charge les modèles de substitution. Vous devez utiliser Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka. Utilisez la fonction SQL get_secret pour récupérer la valeur de ce champ. Pour de plus amples informations sur les modèles de substitution, veuillez consulter the section called “Modèles de substitution” (p. 541). Pour plus d'informations sur la fonction SQL get_secret, consultezthe section called “get_secret (secretId, SecretType, key, roleArn) ” (p. 511). Utilisation de l'`SecretString` Paramètre .

sasl.mécanisme

Le mécanisme de sécurité utilisé pour se connecter à votre courtier Kafka. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`.

Valeurs valides : `PLAIN`, `SCRAM-SHA-512`, `GSSAPI`.

Note

`SCRAM-SHA-512`est le seul mécanisme de sécurité pris en charge dans les régions cn-north-1, cn-northwest-1, us-gov-east-1 et us-gov-west-1.

sasl.plain.username

Nom d'utilisateur utilisé pour récupérer la chaîne secrète de Secrets Manager. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`PLAIN`pour`sasl.mechanism`.

sasl.plain.password

Mot de passe utilisé pour récupérer la chaîne secrète de Secrets Manager. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`PLAIN`pour`sasl.mechanism`.

sasl.scram.username

Nom d'utilisateur utilisé pour récupérer la chaîne secrète de Secrets Manager. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`SCRAM-SHA-512`pour`sasl.mechanism`.

sasl.scram.password

Mot de passe utilisé pour récupérer la chaîne secrète de Secrets Manager. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`SCRAM-SHA-512`pour`sasl.mechanism`.

sasl.kerberos.keytab

Fichier keytab pour l'authentification Kerberos dans Secrets Manager. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`GSSAPI`pour`sasl.mechanism`.

Ce champ prend en charge les modèles de substitution. Vous devez utiliser Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka. Utilisez la fonction SQL get_secret pour récupérer la valeur de ce champ. Pour de plus amples informations sur les modèles de substitution, veuillez consulter the section called “Modèles de substitution” (p. 541). Pour plus d'informations sur la fonction SQL get_secret, consultezthe section called “get_secret (secretId, SecretType, key, roleArn) ” (p. 511).Utilisation de l'`SecretBinary` Paramètre .

sasl.kerberos.service.name

Le nom principal Kerberos sous lequel Apache Kafka fonctionne. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`GSSAPI`pour`sasl.mechanism`.

sasl.kerberos.krb5.kdc

Nom d'hôte du centre de distribution de clés (KDC) auquel votre client producteur Apache Kafka se connecte. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`GSSAPI`pour`sasl.mechanism`.

sasl.kerberos.krb5.realm

Le domaine auquel votre client producteur Apache Kafka se connecte. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`GSSAPI`pour`sasl.mechanism`.

sasl.kerberos.principal

L'identité Kerberos unique à laquelle Kerberos peut attribuer des tickets pour accéder aux services Kerberos. Cette valeur est requise lorsque vous spécifiez`SASL_SSL`pour`security.protocol`et`GSSAPI`pour`sasl.mechanism`.

Examples

L'exemple JSON suivant définit une action Apache Kafka dans unAWS IoTRègle

```
{
"topicRulePayload": {
```

```
    "sql": "SELECT * FROM 'some/topic'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
    {
        "kafka": {
            "destinationArn": "arn:aws:iot:region:123456789012:ruledestination/
vpc/VPCDestinationARN",
            "topic": "TopicName",
            "clientProperties": {
                "bootstrap.servers": "kafka.com:9092",
                "security.protocol": "SASL_SSL",
                "ssl.truststore": "${get_secret('kafka_client_truststore', 'SecretBinary',
'arn:aws:iam::123456789012:role/kafka-get-secret-role-name')}",
                "ssl.truststore.password": "kafka password",
               "sasl.mechanism": "GSSAPI",
               "sasl.kerberos.service.name": "kafka",
               "sasl.kerberos.krb5.kdc": "kerberosdns.com",
               "sasl.kerberos.keytab": "${get_secret('kafka_keytab',
               'SecretBinary', 'arn:aws:iam::123456789012:role/kafka-get-secret-role-
name')}",
               "sasl.kerberos.krb5.realm": "KERBEROSREALM",
               "sasl.kerberos.principal": "kafka-keytab/kafka-keytab.com"
            }
        }
    }
]

}
```

Remarques importantes concernant votre configuration Kerberos

- Votre centre de distribution de clés (KDC) doit être résolu via le système DNS (Domain Name System) privé dans votre VPC cible. Une approche possible consiste à ajouter l'entrée DNS KDC à une zone hébergée privée. Pour plus d'informations sur cette approche, consultezUtilisation des zones hébergées privées.
- La résolution DNS doit être activée sur chaque VPC. Pour plus d'informations, consultez Utilisation de DNS avec votre VPC.
- Les groupes de sécurité de l'interface réseau et les groupes de sécurité au niveau de l'instance dans la destination du VPC doivent autoriser le trafic à partir de votre VPC sur les ports suivants.
  - Trafic TCP sur le port d'écoute du broker d'amorçage (souvent 9092, mais doit se situer dans la plage 9000 - 9100)
  - Trafic TCP et UDP sur le port 88 pour le KDC
- `SCRAM-SHA-512`est le seul mécanisme de sécurité pris en charge dans les régions cn-north-1, cn-northwest-1, us-gov-east-1 et us-gov-west-1.

Alarmes CloudWatch

L'alarme CloudWatch (`cloudWatchAlarm`) modifie l'état d'une alarme Amazon CloudWatch. Vous pouvez spécifier la raison du changement d'état et la valeur dans cet appel.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`cloudwatch:SetAlarmState`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`alarmName`

Nom de l'alarme CloudWatch.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`stateReason`

Raisons de la modification de l'alarme.

Prend en chargeModèles de substitution (p. 541) : Oui

`stateValue`

Valeur de l'état de l'alarme. Valeurs valides : `OK`, `ALARM`, `INSUFFICIENT_DATA`.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

Rôle IAM qui autorise l'accès à l'alarme CloudWatch. Pour plus d'informations, consultez Requirements (p. 408).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action d'alarme CloudWatch dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchAlarm": {
                    "alarmName": "IotAlarm",
                    "stateReason": "Temperature stabilized.",
                    "stateValue": "OK",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce qu'Amazon CloudWatch ?dans leGuide de l'utilisateur Amazon CloudWatch.
- Utilisation des alarmes Amazon CloudWatchdans leGuide de l'utilisateur Amazon CloudWatch.

CloudWatch Logs

Logs CloudWatch Logs (`cloudwatchLogs`) envoie les données aux journaux Amazon CloudWatch Logs. Vous pouvez spécifier le groupe de journaux auquel l'action envoie des données.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`logs:CreateLogStream`,`logs:DescribeLogStreams`, et`logs:PutLogEvents`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données de journal dans CloudWatch Logs, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterChiffrer les données du journal dans CloudWatch Logs à l'aideAWS KMSdans leManuel de l'utilisateur Amazon CloudWatch Logs.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`logGroupName`

Groupe de journaux CloudWatch auquel l'action envoie des données.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`roleArn`

Rôle IAM qui autorise l'accès au groupe de journaux CloudWatch. Pour plus d'informations, consultez Requirements (p. 410).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action CloudWatch Logs dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchLogs": {
                    "logGroupName": "IotLogs",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation de Amazon CloudWatch Logsdans leManuel de l'utilisateur Amazon CloudWatch Logs

Métriques CloudWatch

La métrique CloudWatch (`cloudwatchMetric`) capture une mesure Amazon CloudWatch. Vous pouvez spécifier le namespace, le nom, la valeur, l'unité et l'horodatage de la métrique.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`cloudwatch:PutMetricData`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`metricName`

Nom de la métrique CloudWatch.

Prend en chargeModèles de substitution (p. 541) : Oui

`metricNamespace`

Nom de l'espace de nom de la métrique CloudWatch.

Prend en chargeModèles de substitution (p. 541) : Oui

`metricUnit`

Unité de métrique prise en charge par CloudWatch.

Prend en chargeModèles de substitution (p. 541) : Oui

`metricValue`

Chaîne qui contient la valeur de métrique CloudWatch.

Prend en chargeModèles de substitution (p. 541) : Oui

`metricTimestamp`

(Facultatif) Chaîne contenant l'horodatage, exprimée en secondes dans l'époque Unix. Par défaut, l'heure actuelle de l'époque Unix.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

Rôle IAM qui autorise l'accès à la métrique CloudWatch. Pour plus d'informations, consultez Requirements (p. 411).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action de métrique CloudWatch dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchMetric": {
                    "metricName": "IotMetric",
                    "metricNamespace": "IotNamespace",
                    "metricUnit": "Count",
                    "metricValue": "1",
                    "metricTimestamp": "1456821314",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action de mesure CloudWatch avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "cloudwatchMetric": {
                    "metricName": "${topic()}",
                    "metricNamespace": "${namespace}",
                    "metricUnit": "${unit}",
                    "metricValue": "${value}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce qu'Amazon CloudWatch ?dans leGuide de l'utilisateur Amazon CloudWatch.
- Utilisation des métriques Amazon CloudWatchdans leGuide de l'utilisateur Amazon CloudWatch.

DynamoDB

DynamoDB (`dynamoDB`) écrit tout ou partie d'un message MQTT dans une table Amazon DynamoDB.

Vous pouvez suivre un didacticiel qui vous montre comment créer et tester une règle avec une action DynamoDB. Pour plus d'informations, consultez Stocker les données de périphérique dans une table DynamoDB (p. 146).

Note

Cette règle écrit des données non JSON dans DynamoDB en tant que données binaires. La console DynamoDB affiche les données sous la forme de texte codé en Base64.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`dynamodb:PutItem`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données au repos dans DynamoDB, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterCMK gérée par le clientdans leManuel de mise en route Amazon DynamoDB.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`tableName`

Le nom de la table DynamoDB.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`hashKeyField`

Nom de la clé de hachage (également appelée clé de partition).

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`hashKeyType`

(Facultatif) Type de données de la clé de hachage (également appelée clé de partition). Valeurs valides :`STRING`,`NUMBER`.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`hashKeyValue`

Valeur de la clé de hachage. Envisagez d'utiliser un modèle de substitution tel que`${topic()}`ou`${timestamp()}`.

Prend en chargeModèles de substitution (p. 541) : Oui

`rangeKeyField`

(Facultatif) Nom de la clé de plage (également appelée clé de tri).

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`rangeKeyType`

(Facultatif) Type de données de la clé de plage (également appelée clé de tri). Valeurs valides :`STRING`,`NUMBER`.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`rangeKeyValue`

(Facultatif) Valeur de la clé de plage. Envisagez d'utiliser un modèle de substitution tel que`${topic()}`ou`${timestamp()}`.

Prend en chargeModèles de substitution (p. 541) : Oui

`payloadField`

(Facultatif) Nom de la colonne dans laquelle la charge utile est écrite. Si vous ne spécifiez pas cette valeur, la charge utile est renseignée dans la colonne nommée`payload`.

Prend en chargeModèles de substitution (p. 541) : Oui

`operation`

(Facultatif) Type d'opération à effectuer. Valeurs valides : `INSERT`, `UPDATE`, `DELETE`.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleARN`

Rôle IAM qui autorise l'accès à la table DynamoDB. Pour plus d'informations, consultez Requirements (p. 413).

Prend en chargeModèles de substitution (p. 541) : Non

Les données écrites dans la table DynamoDB sont le résultat de l'instruction SQL de la règle.

Examples

L'exemple JSON suivant définit une action DynamoDB dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * AS message FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "dynamoDB": {
                    "tableName": "my_ddb_table",
                    "hashKeyField": "key",
                    "hashKeyValue": "${topic()}",
                    "rangeKeyField": "timestamp",
                    "rangeKeyValue": "${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDB"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce qu'Amazon DynamoDB?dans leAmazon DynamoDB Developer Guide
- Mise en route d'avec DynamoDBdans leAmazon DynamoDB Developer Guide
- Stocker les données de périphérique dans une table DynamoDB (p. 146)

DynamoDBv2

Le DynamoDDBv2 (`dynamoDBv2`) écrit tout ou partie d'un message MQTT dans une table Amazon DynamoDB. Chaque attribut de la charge utile est écrit à une colonne distincte dans la base de données DynamoDB.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`dynamodb:PutItem`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- La charge du message MQTT doit contenir une clé de niveau racine qui correspond à la clé de partition primaire de la table et une clé de niveau racine qui correspond à la clé de tri primaire de la table, si elle est définie.
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données au repos dans DynamoDB, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterCMK gérée par le clientdans leManuel de mise en route Amazon DynamoDB.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`putItem`

Objet spécifiant la table DynamoDB dans laquelle les données de message seront écrites. Cet objet doit contenir les informations suivantes :

`tableName`

Le nom de la table DynamoDB.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`roleARN`

Rôle IAM qui autorise l'accès à la table DynamoDB. Pour plus d'informations, consultez Requirements (p. 415).

Prend en chargeModèles de substitution (p. 541) : Non

Les données écrites dans la table DynamoDB sont le résultat de l'instruction SQL de la règle.

Examples

L'exemple JSON suivant définit une action DynamoDBv2 dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * AS message FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
```

```
                "dynamoDBv2": {
                    "putItem": {
                        "tableName": "my_ddb_table"
                    },
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDBv2",
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action DynamoDB avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2015-10-08",
        "actions": [
            {
                "dynamoDBv2": {
                    "putItem": {
                        "tableName": "${topic()}"
                    },
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDBv2"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce qu'Amazon DynamoDB?dans leAmazon DynamoDB Developer Guide
- Mise en route d'avec DynamoDBdans leAmazon DynamoDB Developer Guide

Elasticsearch

La recherche Elasticsearch (`elasticsearch`) écrit les données à partir de messages MQTT dans un domaine Amazon Elasticsearch Service. Vous pouvez ensuite utiliser des outils comme Kibana pour interroger et visualiser des données dans Elasticsearch.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`es:ESHttpPut`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données au repos dans Elastic search, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterChiffrement de données au repos pour Amazon Elasticsearch Servicedans leGuide du développeur Amazon Elasticsearch Service.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`endpoint`

Point de terminaison de votre domaine Amazon Elasticsearch Service.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`index`

Index Elasticsearch. dans lequel vous souhaitez stocker vos données.

Prend en chargeModèles de substitution (p. 541) : Oui

`type`

Type de document que vous stockez.

Prend en chargeModèles de substitution (p. 541) : Oui

`id`

Identifiant unique de chaque document.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleARN`

Rôle IAM qui autorise l'accès au domaine Elasticsearch. Pour plus d'informations, consultez Requirements (p. 416).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action Elasticsearch dans unAWS IoTet la façon dont vous pouvez spécifier les champs pour la`elasticsearch`action. Pour de plus amples informations, veuillez consulterElasticsearchAction.

```
{
    "topicRulePayload": {
        "sql": "SELECT *, timestamp() as timestamp FROM 'iot/test'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "elasticsearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "my-index",
                    "type": "my-type",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_es"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action Elasticsearch avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "elasticsearch": {
                    "endpoint": "https://my-endpoint",
                    "index": "${topic()}",
                    "type": "${type}",
                    "id": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_es"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation de Amazon Elasticsearch Servicedans leGuide du développeur Amazon Elasticsearch Service

HTTPS

HTTPS (`http`) envoie les données à partir d'un message MQTT à une application web ou à un service.

Requirements

Cette action de règle impose les critères suivants :

- Vous devez confirmer et activer les points de terminaison HTTPS avant que le moteur de règles puisse les utiliser. Pour plus d'informations, consultez Utilisation des destinations des règles de rubrique (p. 454).

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`url`

Point de terminaison HTTPS où le message est envoyé à l'aide de la méthode HTTP POST. Si vous utilisez une adresse IP à la place d'un nom d'hôte, il doit s'agir d'une adresse IPv4. Les adresses IPv6 ne sont pas prises en charge.

Prend en chargeModèles de substitution (p. 541) : Oui

`confirmationUrl`

(Facultatif) Si spécifié,AWS IoTutilise l'URL de confirmation pour créer une destination de règle de rubrique correspondante. Vous devez activer la destination de la règle de rubrique avant de l'utiliser dans une action HTTPS. Pour plus d'informations, consultez Utilisation des destinations des règles de rubrique (p. 454). Si vous utilisez des modèles de substitution, vous devez créer manuellement des destinations de règles de rubrique avant que l'action `http` puisse être utilisée. `confirmationUrl` doit être un préfixe de `url`.

La relation entre `url` et `confirmationUrl` est décrite par les éléments suivants :

- Si `url` est codé en dur et que `confirmationUrl` n'est pas fourni, nous traitons implicitement le champ `url` en tant que `confirmationUrl`. AWS IoT crée une destination de règle de rubrique pour `url`.
- Si `url` et `confirmationUrl` sont codés en dur, `url` doit commencer par `confirmationUrl`. AWS IoT crée une destination de règle de rubrique pour `confirmationUrl`.
- Si `url` contient un modèle de substitution, vous devez spécifier `confirmationUrl` et `url` doit commencer par `confirmationUrl`. Si `confirmationUrl` contient des modèles de substitution, vous devez créer manuellement des destinations de règle de rubrique avant que l'action `http` puisse être utilisée. Si `confirmationUrl` ne contient pas de modèles de substitution, AWS IoT crée une destination de règle de rubrique pour `confirmationUrl`.

Prend en chargeModèles de substitution (p. 541) : Oui

`headers`

(Facultatif) Liste des en-têtes à inclure dans les requêtes HTTP vers le point de terminaison. Chaque en-tête doit contenir les informations suivantes :

`key`

La clé de l'en-tête.

Prend en chargeModèles de substitution (p. 541) : Non

`value`

Valeur de l'en-tête.

Prend en chargeModèles de substitution (p. 541) : Oui

Note

Le type de contenu par défaut est application/json lorsque la charge utile est au format JSON. Sinon, il s'agit de application/octet-stream. Vous pouvez le remplacer en spécifiant le type de contenu exact dans l'en-tête avec le type de contenu clé (insensible à la casse).

`auth`

(Facultatif) Authentification utilisée par le moteur de règles pour se connecter à l'URL du point de terminaison spécifiée dans le`url`Argument. Actuellement, Signature Version 4 est le seul type d'authentification pris en charge. Pour de plus amples informations, veuillez consulter Autorisation HTTP.

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit unAWS IoTavec une action HTTPS.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "http": {
                    "url": "https://www.example.com/subpath",
                    "confirmationUrl": "https://www.example.com",
                    "headers": [
                        {
```

```
                        "key": "static_header_key",
                        "value": "static_header_value"
                    },
                    {
                        "key": "substitutable_header_key",
                        "value": "${value_from_payload}"
                    }
                ]
            }
        }
    ]
  }
}
```

Logique de nouvelle tentative d'action HTTP

La .AWS IoTRéessaie l'action HTTPS selon les règles suivantes :

- Le moteur de règles essaie d'envoyer un message au moins une fois.
- Le moteur de règles effectue au plus deux nouvelles tentatives. Le nombre maximum de nouvelles tentatives est trois.
- Le moteur de règles n'effectue pas de nouvelle tentative si :
  - La tentative précédente a fourni une réponse supérieure à 16 384 octets.
  - Le service web ou l'application en aval ferme la connexion TCP après la tentative.
  - La durée totale de l'exécution d'une demande et des nouvelles tentatives a dépassé le délai d'expiration de la demande.
  - La requête renvoie un code d'état HTTP autre que 429, 500-599.

  Note

  Les coûts standard de transfert de données s'appliquent aux nouvelles tentatives.

Voir aussi

- Utilisation des destinations des règles de rubrique (p. 454)
- Données de route directement à partir de AWS IoT Core à vos services Webdans leInternet des objets surAWSBlog

IoT Analytics

L'analyse IoT Analytics (`iotAnalytics`) envoie les données à partir d'un message MQTT à unAWS IoT Analyticscanal.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`iotanalytics:BatchPutMessage`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

  La stratégie attachée au rôle que vous spécifiez doit se présenter comme suit.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotanalytics:BatchPutMessage",
            "Resource": [
                "arn:aws:iotanalytics:us-west-2:account-id:channel/mychannel"
            ]
        }
    ]
}
```

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`batchMode`

(Facultatif) Indique si l'action doit être traitée en tant que lot. La valeur par défaut est `false`.

Quand`batchMode`est`true`et que l'instruction SQL de règle est évaluée à un tableau, chaque élément Array est livré sous la forme d'un message distinct lorsqu'il est transmis par`BatchPutMessage`à laAWS IoTCanal analytique. Le tableau résultant ne peut pas contenir plus de 100 messages.

Prend en chargeModèles de substitution (p. 541) : Non

`channelName`

Nom du canal AWS IoT Analytics dans lequel les données doivent être écrites.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`roleArn`

Rôle IAM qui autorise l'accès àAWS IoT Analyticscanal. Pour plus d'informations, consultez Requirements (p. 420).

Prend en chargeModèles de substitution (p. 541) : Non

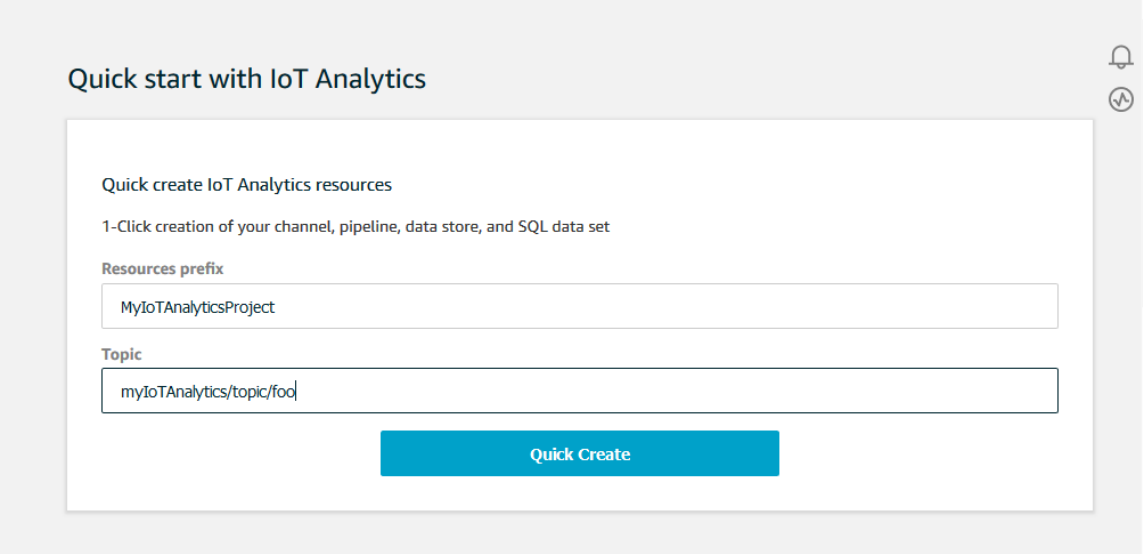
Examples

L'exemple JSON suivant définit une action IoT Analytics dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotAnalytics": {
                    "channelName": "mychannel",
                    "roleArn": "arn:aws:iam::123456789012:role/analyticsRole",
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce que AWS IoT Analytics ? dans le Guide de l'utilisateur AWS IoT Analytics
- La .AWS IoT Analyticsdispose également d'unDémarrage rapidequi vous permet de créer un canal, un magasin de données, un pipeline et des données stockées en un clic. Pour de plus amples informations, veuillez consulterAWS IoT Analyticsguide de démarrage rapide de la consoledans leAWS IoT AnalyticsGuide de l'utilisateur.



IoT Events

Les IoT Events (`iotEvents`) envoie les données à partir d'un message MQTT à unAWS IoT EventsEntrée

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`iotevents:BatchPutMessage`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`batchMode`

(Facultatif) Indique si les actions d'événement doivent être traitées en tant que lot. La valeur par défaut est `false`.

Quand`batchMode`est`true`et que l'instruction SQL de règle est évaluée à un tableau, chaque élément Array est traité comme un message distinct lorsqu'il est envoyé àAWS IoTÉvénements en appelant`BatchPutMessage`. Le tableau résultant ne peut pas contenir plus de 10 messages.

Quand `batchMode` est `true`, vous ne pouvez pas spécifier un `messageId`.

Prend en chargeModèles de substitution (p. 541) : Non

`inputName`

Nom de l'entrée AWS IoT Events.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`messageId`

(Facultatif) Utilisez ce paramètre pour vous assurer qu'une seule entrée (message) avec un`messageId`est traitée par unAWS IoT Eventsdétecteur. Vous pouvez utiliser la méthode`${newuuid()}`Pour générer un ID unique pour chaque demande.

Quand `batchMode` équivaut à `true`, vous ne pouvez pas spécifier de valeur `messageId` ; une nouvelle valeur UUID sera attribuée.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

Rôle IAM qui autoriseAWS IoTpour envoyer une entrée à unAWS IoT Eventsdétecteur. Pour plus d'informations, consultez Requirements (p. 422).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action IoT Events dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotEvents": {
                    "inputName": "MyIoTEventsInput",
                    "messageId": "${newuuid()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_events"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation d'AWS IoT Events?dans leAWS IoT EventsManuel du développeur

IoT SiteWise

L'IoT SiteWise (`iotSiteWise`) envoie les données à partir d'un message MQTT aux propriétés de ressources dansAWS IoT SiteWise.

Vous pouvez suivre un didacticiel qui vous montre comment ingérer des données à partir d'objets AWS IoT. Pour de plus amples informations, veuillez consulterIngestion des données dansAWS IoT SiteWisefromAWS IoTthingsdans leAWS IoT SiteWiseGuide de l'utilisateur.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`iotsitewise:BatchPutAssetPropertyValue`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  Vous pouvez attacher l'exemple de stratégie d'approbation suivant au rôle.

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

  Pour améliorer la sécurité, vous pouvez spécifier un chemin de hiérarchie de ressource AWS IoT SiteWise dans la propriété `Condition`. L'exemple suivant est une stratégie d'approbation qui spécifie un chemin de hiérarchie de ressource.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotsitewise:BatchPutAssetPropertyValue",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iotsitewise:assetHierarchyPath": [
                        "/root node asset ID",
                        "/root node asset ID/*"
                    ]
                }
            }
        }
    ]
}
```

- Lorsque vous envoyez des données àAWS IoT SiteWiseAvec cette action, vos données doivent répondre aux exigences de la`BatchPutAssetPropertyValue`. Pour de plus amples informations, veuillez consulter BatchPutasSetPropertyValue dans la référence de l'API AWS IoT SiteWise.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`putAssetPropertyValueEntries`

Une liste d'entrées de valeurs de propriétés de ressources dont chaque entrée contient les informations suivantes :

`propertyAlias`

(Facultatif) L'alias de propriété associé à votre propriété de ressources. Vous devez spécifier un `propertyAlias` ou à la fois un `assetId` et un `propertyId`. Pour plus d'informations sur les

alias de propriété, consultezMappage des flux de données industrielles avec des propriétés de ressourcesdans leAWS IoT SiteWiseGuide de l'utilisateur.

Prend en chargeModèles de substitution (p. 541) : Oui

`assetId`

(Facultatif) ID de la propriétéAWS IoT SiteWiseasset Vous devez spécifier un `propertyAlias` ou à la fois un `assetId` et un `propertyId`.

Prend en chargeModèles de substitution (p. 541) : Oui

`propertyId`

(Facultatif) ID du bien. Vous devez spécifier un `propertyAlias` ou à la fois un `assetId` et un `propertyId`.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`entryId`

(Facultatif) Identifiant unique de cette entrée. Vous pouvez définir `entryId` afin de mieux suivre les informations indiquant quel message a causé une erreur en cas d'échec. La valeur par défaut est un nouvel UUID.

Prend en chargeModèles de substitution (p. 541) : Oui

`propertyValues`

Liste des valeurs de propriété à insérer, contenant chacune l'horodatage, la qualité et la valeur (TQV) au format suivant :

`timestamp`

Structure d'horodatage contenant les informations suivantes :

`timeInSeconds`

Chaîne contenant l'heure en secondes au format d'heure Unix epoch. Si votre charge utile de message n'a pas d'horodatage, vous pouvez utiliser timestamp() (p. 533), qui renvoie l'heure actuelle en millisecondes. Pour convertir cette heure en secondes, vous pouvez utiliser le modèle de substitution suivant : **`${floor(timestamp() / 1E3)}`**.

Prend en chargeModèles de substitution (p. 541) : Oui

`offsetInNanos`

(Facultatif) Chaîne contenant le décalage de nanoseconde par rapport à l'heure en secondes. Si votre charge utile de message n'a pas d'horodatage, vous pouvez utiliser timestamp() (p. 533), qui renvoie l'heure actuelle en millisecondes. Pour calculer le décalage de nanosecondes à partir de cette heure, vous pouvez utiliser le modèle de substitution suivant : **`${(timestamp() % 1E3) * 1E6}`**.

Prend en chargeModèles de substitution (p. 541) : Oui

En ce qui concerne l'époque Unix,AWS IoT SiteWiseN'accepte que les entrées dont l'horodatage est de 7 jours au maximum dans le passé et jusqu'à 5 minutes dans le futur.

`quality`

(Facultatif) Chaîne qui décrit la qualité de la valeur. Valeurs valides : `GOOD`, `BAD`, `UNCERTAIN`.

Prend en chargeModèles de substitution (p. 541) : Oui

`value`

Structure de valeur qui contient l'un des champs de valeur suivants, en fonction du type de données de la propriété de la ressource :

`booleanValue`

(Facultatif) Chaîne qui contient la valeur booléenne de l'entrée de valeur.

Prend en chargeModèles de substitution (p. 541) : Oui

`doubleValue`

(Facultatif) Chaîne qui contient la valeur double de l'entrée de valeur.

Prend en chargeModèles de substitution (p. 541) : Oui

`integerValue`

(Facultatif) Chaîne qui contient la valeur d'entier de l'entrée de valeur.

Prend en chargeModèles de substitution (p. 541) : Oui

`stringValue`

(Facultatif) Valeur de chaîne de l'entrée de valeur.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

ARN du rôle IAM qui accordeAWS IoTAutorisation d'envoyer une valeur de propriété de ressource àAWS IoT SiteWise. Pour plus d'informations, consultez Requirements (p. 424).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action IoT SiteWise de base dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotSiteWise": {
                    "putAssetPropertyValueEntries": [
                        {
                            "propertyAlias": "/some/property/alias",
                            "propertyValues": [
                                {
                                    "timestamp": {
                                        "timeInSeconds": "${my.payload.timeInSeconds}"
                                    },
                                    "value": {
                                        "integerValue": "${my.payload.value}"
                                    }
                                }
                            ]
                        }
                    ],
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sitewise"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action IoT SiteWise dans unAWS IoTRègle Cet exemple utilise la rubrique comme alias de propriété et la fonction `timestamp()`. Par exemple, si vous publiez des données vers `/company/windfarm/3/turbine/7/rpm`, cette action envoie les données à la propriété de ressource avec un alias de propriété identique à la rubrique que vous avez spécifiée.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM '/company/windfarm/+/turbine/+/+'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "iotSiteWise": {
                    "putAssetPropertyValueEntries": [
                        {
                            "propertyAlias": "${topic()}",
                            "propertyValues": [
                                {
                                    "timestamp": {
                                        "timeInSeconds": "${floor(timestamp() / 1E3)}",
                                        "offsetInNanos": "${(timestamp() % 1E3) * 1E6}"
                                    },
                                    "value": {
                                        "doubleValue": "${my.payload.value}"
                                    }
                                }
                            ]
                        }
                    ],
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sitewise"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce que AWS IoT SiteWise ? dans le Guide de l'utilisateur AWS IoT SiteWise
- Ingestion de données à l'aide d' AWS IoT Core Règlesdans leAWS IoT SiteWiseGuide de l'utilisateur
- Ingestion des données dansAWS IoT SiteWisefromAWS IoTthingsdans leAWS IoT SiteWiseGuide de l'utilisateur
- Dépannage d'unAWS IoT SiteWiseAction de règledans leAWS IoT SiteWiseGuide de l'utilisateur

Kinesis Data Firehose

Kinesis Data Firehose (`firehose`) envoie les données à partir d'un message MQTT dans un flux Amazon Kinesis Data Firehose.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`firehose:PutRecord`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

- Si vous utilisez Kinesis Data Firehose pour envoyer des données à un compartiment Amazon S3, et que vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans Amazon S3, Kinesis Data Firehose doit avoir accès à votre compartiment et l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterAttribution à Kinesis Data Firehose d'un accès à une destination Amazon S3dans leGuide du développeur Amazon Kinesis Data Firehose.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`batchMode`

(Facultatif) Indique s'il faut livrer le flux Kinesis Data Firehose en tant que lot à l'aide de`PutRecordBatch`. La valeur par défaut est `false`.

Quand`batchMode`est`true`et que l'instruction SQL de la règle est évaluée à un tableau, chaque élément Array forme un enregistrement dans la`PutRecordBatch`de la demande. Le tableau résultant ne peut pas contenir plus de 500 enregistrements.

Prend en chargeModèles de substitution (p. 541) : Non

`deliveryStreamName`

Flux Kinesis Data Firehose dans lequel écrire les données du message.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`separator`

(Facultatif) Séparateur de caractères qui est utilisé pour séparer les enregistrements écrits dans le flux Kinesis Data Firehose. Si vous omettez ce paramètre, le flux n'utilise aucun séparateur. Valeurs valides :`,`(comma),`\t`(onglet),`\n`(nouvelle ligne),`\r\n`(nouvelle ligne Windows).

Prend en chargeModèles de substitution (p. 541) : Non

`roleArn`

Rôle IAM qui permet l'accès au flux Kinesis Data Firehose. Pour plus d'informations, consultez Requirements (p. 427).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action Kinesis Data Firehose dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "firehose": {
                    "deliveryStreamName": "my_firehose_stream",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose"
                }
            }
        ]
```

```
    }
}
```

L'exemple JSON suivant définit une action Kinesis Data Firehose avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "firehose": {
                    "deliveryStreamName": "${topic()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation de Amazon Kinesis Data Firehosedans leGuide du développeur Amazon Kinesis Data Firehose

Kinesis Data Streams

Les flux de Kinesis Data Streams (`kinesis`) écrit les données d'un message MQTT dans Amazon Kinesis Data Streams.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`kinesis:PutRecord`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans les Kinesis Data Streams, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterAutorisations d'utiliser les clés principales KMS générées par l'utilisateurdans leGuide du développeur Amazon Kinesis Data Streams.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`stream`

Flux de données Kinesis dans lequel écrire les données.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`partitionKey`

Clé de partition utilisée pour déterminer dans quelle partition les données sont écrites. La clé de partition est généralement composée d'une expression (par exemple,`${topic()}`ou`${timestamp()}`).

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

ARN du rôle IAM qui accordeAWS IoTAutorisation d'accéder au flux de données Kinesis. Pour plus d'informations, consultez Requirements (p. 429).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action Kinesis Data Streams dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "kinesis": {
                    "streamName": "my_kinesis_stream",
                    "partitionKey": "${topic()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_kinesis"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action Kinesis avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "kinesis": {
                    "streamName": "${topic()}",
                    "partitionKey": "${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_kinesis"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation de Amazon Kinesis Data Streamsdans leGuide du développeur Amazon Kinesis Data Streams

Lambda

Lambda (`lambda`) appelle une actionAWS Lambda, en transmettant un message MQTT.AWS IoTappelle les fonctions Lambda de manière asynchrone.

Vous pouvez suivre un didacticiel qui vous montre comment créer et tester une règle avec une action Lambda. Pour plus d'informations, consultez Formater une notification à l'aide d'unAWS Lambdafonction (p. 152).

Requirements

Cette action de règle impose les critères suivants :

- PourAWS IoTPour appeler une fonction Lambda, vous devez configurer une stratégie accordant l'`lambda:InvokeFunction`Autorisation d'AWS IoT. Vous ne pouvez invoquer qu'une fonction Lambda définie dans le même Région AWS où votre politique Lambda existe. Les fonctions Lambda utilisent des stratégies basées sur les ressources, vous devez attacher la stratégie à la fonction Lambda même.

  Utilisez la stratégie suivanteAWS CLIpour attacher une stratégie qui accorde les critères`lambda:InvokeFunction`Autorisation.

```
aws lambda add-permission --function-name function_name --region region --principal
 iot.amazonaws.com --source-arn arn:aws:iot:region:account-id:rule/rule_name --source-
account account-id --statement-id unique_id --action "lambda:InvokeFunction"
```

  La `.add-permission`attend les paramètres suivants :

  `--function-name`

  Nom de la fonction Lambda. Vous ajoutez une nouvelle autorisation pour mettre à jour la stratégie de ressources de la fonction.

  `--region`

  La . Région AWS de la fonction.

  `--principal`

  Le principal qui obtient la permission. Cela devrait être`iot.amazonaws.com`pour permettreAWS IoTAutorisation d'appeler la fonction Lambda.

  `--source-arn`

  ARN de la règle. Vous pouvez utiliser la stratégie`get-topic-rule` AWS CLIpour obtenir l'ARN d'une règle.

  `--source-account`

  La . Compte AWS où la règle est définie.

  `--statement-id`

  Identifiant unique de l'instruction.

  `--action`

  Action Lambda que vous souhaitez autoriser dans cette instruction. AutoriserAWS IoTPour appeler une fonction Lambda, spécifiez`lambda:InvokeFunction`.

  Important

  Si vous ajoutez une autorisation pour unAWS IoTprincipal sans fournir l'ARN source, tout Compte AWS qui crée une règle avec votre action Lambda peut déclencher des règles pour appeler votre fonction Lambda à partir de.AWS IoT.

Pour plus d'informations, consultez Autorisations AWS Lambda .

- Si vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans Lambda, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterChiffrement au reposdans leAWS LambdaManuel du développeur.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`functionArn`

ARN de la fonction Lambda à invoquer.AWS IoTdoit avoir l'autorisation d'appeler la fonction. Pour plus d'informations, consultez Requirements (p. 431).

Si vous ne spécifiez aucune version ni aucun alias pour votre fonction Lambda, la version la plus récente de la fonction est exécutée. Vous pouvez spécifier une version ou un alias si vous souhaitez exécuter une version spécifique de votre fonction Lambda. Pour spécifier une valeur ou un alias, ajoutez la version ou l'alias à l'ARN de la fonction Lambda.

```
arn:aws:lambda:us-east-2:123456789012:function:myLambdaFunction:someAlias
```

Pour plus d'informations sur la gestion des versions et les alias, consultezAWS Lambdaversion de fonction et alias.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

Examples

L'exemple JSON suivant définit une action Lambda dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-
east-2:123456789012:function:myLambdaFunction"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action Lambda avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
```

```
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:
${topic()}"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation d'AWS Lambda?dans leAWS LambdaManuel du développeur
- Formater une notification à l'aide d'unAWS Lambdafonction (p. 152)

Republish

La republication (`republish`) republie un message MQTT dans une autre rubrique MQTT.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`iot:Publish`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`topic`

Rubrique MQTT dans laquelle republier le message.

Pour republier dans une rubrique réservée, qui commence par`$`, utilisez`$$`à la place. Par exemple, pour republier dans la rubrique des clichés instantanés du périphérique`$aws/things/MyThing/shadow/update`, spécifiez le sujet comme`$$aws/things/MyThing/shadow/update`.

> **Note**
>
> Republication versRubriques de tâche réservée (p. 98)n'est pas pris en charge.

Prend en chargeModèles de substitution (p. 541) : Oui

`qos`

(Facultatif) Le niveau de qualité de service (QoS) à utiliser pour republier des messages. Valeurs valides : `0`,`1`. La valeur par défaut est `.0`. Pour de plus amples informations sur la QoS MQTT, veuillez consulter.MQTT (p. 82).

Prend en chargeModèles de substitution (p. 541) : Non

`roleArn`

Rôle IAM qui autoriseAWS IoTPour publier dans la rubrique MQTT. Pour plus d'informations, consultez Requirements (p. 433).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action de republication dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "republish": {
                    "topic": "another/topic",
                    "qos": 1,
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action de republication avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "republish": {
                    "topic": "${topic()}/republish",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"
                }
            }
        ]
    }
}
```

S3

La S3 (`s3`) écrit les données à partir d'un message MQTT dans un compartiment Amazon Simple Storage Service (Amazon S3).

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer les`s3:PutObject`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans Amazon S3, le service doit avoir l'autorisation d'utiliser la CMK pour le compte

du mandataire. Pour de plus amples informations, veuillez consulterAWSCMK gérées par le client et clés CMK gérées par le clientdans leManuel du développeur Amazon Simple Storage Service.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`bucket`

Compartiment Amazon S3 dans lequel écrire les données.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`cannedacl`

(Facultatif) La liste ACL prédéfinie Amazon S3 qui contrôle l'accès à l'objet identifié par la clé d'objet. Pour plus d'informations, y compris sur les valeurs autorisées, consultezListe ACL prête à l'emploi.

Prend en chargeModèles de substitution (p. 541) : Non

`key`

Chemin d'accès au fichier dans lequel les données sont écrites.

Prenons un exemple où ce paramètre est`${topic()}/${timestamp()}`et la règle reçoit un message dans lequel le sujet est`some/topic`. Si l'horodatage actuel est`1460685389`, alors cette action écrit les données dans un fichier appelé`1460685389`dans le`some/topic`Dossier du compartiment S3.

Note

Si vous utilisez une clé statique,AWS IoTécrase un seul fichier à chaque appel de la règle. Nous vous recommandons d'utiliser l'horodatage de message ou un autre identifiant de message unique, de sorte qu'un nouveau fichier est enregistré dans Amazon S3 pour chaque message reçu.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

Rôle IAM qui autorise l'accès au compartiment Amazon S3. Pour plus d'informations, consultez Requirements (p. 434).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action S3 dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "s3": {
                    "bucketName": "my-bucket",
```

```
                    "cannedacl": "public-read",
                    "key": "${topic()}/${timestamp()}",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3"
                }
            }
        ]
    }
}
```

Voir aussi

- Qu'est-ce qu'Amazon S3 ?dans leManuel du développeur Amazon Simple Storage Service

Salesforce IoT

L'IoT Salesforce (`salesforce`) envoie les données du message MQTT qui a déclenché la règle pour un flux d'entrée Salesforce IoT

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`url`

URL exposée par le flux d'entrée Salesforce IoT. L'URL est disponible depuis la plateforme Salesforce IoT au moment où vous créez un flux d'entrée. Pour plus d'informations, consultez la documentation Salesforce IoT.

Prend en chargeModèles de substitution (p. 541) : Non

`token`

Jeton utilisé pour authentifier l'accès au flux d'entrée Salesforce IoT spécifié. Le jeton est disponible depuis la plateforme Salesforce IoT au moment où vous créez un flux d'entrée. Pour plus d'informations, consultez la documentation Salesforce IoT.

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action IoT Salesforce dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "salesforce": {
                    "token": "ABCDEFGHI123456789abcdefghi123456789",
                    "url": "https://ingestion-cluster-id.my-env.sfdcnow.com/streams/stream-
id/connection-id/my-event"
                }
            }
        ]
    }
```

```
}
```

SNS

SNS (`sns`) envoie les données à partir d'un message MQTT sous forme de notification Push Amazon Simple Notification Service (Amazon SNS).

Vous pouvez suivre un didacticiel qui vous montre comment créer et tester une règle avec une action SNS. Pour plus d'informations, consultez Envoi d'une notification Amazon SNS (p. 139).

> Note
>
> L'action SNS ne prend pas en chargeRubriques Amazon SNS FIFO (premier entré, premier sorti). Puisque le moteur de règles est un service entièrement distribué, il n'existe aucune garantie quant à l'ordre des messages lorsque l'action SNS est appelée.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`sns:Publish`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans Amazon SNS, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterGestion des clésdans leGuide du développeur Amazon Simple Notification Service.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`targetArn`

Rubrique SNS ou appareil individuel auxquels les notifications push sont envoyées.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`messageFormat`

(Facultatif) Format du message. Amazon SNS utilise ce paramètre pour déterminer si la charge utile doit être analysée et si les parties pertinentes de la charge utile spécifiques à la plateforme doivent être extraites. Valeurs valides : `JSON`,`RAW`. La valeur par défaut est`RAW`.

Prend en chargeModèles de substitution (p. 541) : Non

`roleArn`

Rôle IAM qui autorise l'accès à SNS. Pour plus d'informations, consultez Requirements (p. 437).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action SNS dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-east-2:123456789012:my_sns_topic",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action SNS avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-east-1:123456789012:${topic()}",
                    "messageFormat": "JSON",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation d'Amazon Simple Notification Service ?dans leGuide du développeur Amazon Simple Notification Service
- Envoi d'une notification Amazon SNS (p. 139)

SQS

SQS (`sqs`) envoie les données à partir d'un message MQTT dans une file d'attente Amazon Simple Queue Service (Amazon SQS).

> Note
>
> L'action SQS ne prend pas en chargeFiles d'attente Amazon SQS FIFO (premier entré, premier sorti). Puisque le moteur de règles est un service entièrement distribué, il n'existe aucune garantie quant à l'ordre des messages lorsque l'action SQS est déclenchée.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`sqs:SendMessage`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

- Si vous utilisez unAWS Key Management Service(AWS KMS) gérée par le client (CMK) pour chiffrer les données au repos dans Amazon SQS, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterGestion des clésdans leGuide du développeur Amazon Simple Queue Service.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`queueUrl`

URL de la file d'attente Amazon SQS dans laquelle écrire les données.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`useBase64`

Définissez ce paramètre à`true`pour configurer l'action de règle de manière à base64-encoder les données du message avant d'écrire les données dans la file d'attente Amazon SQS. La valeur par défaut est `false`.

Prend en chargeModèles de substitution (p. 541) : Non

`roleArn`

Rôle IAM qui autorise l'accès à la file d'attente Amazon SQS. Pour plus d'informations, consultez Requirements (p. 438).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action SQS dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sqs": {
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/
my_sqs_queue",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sqs"
                }
            }
        ]
    }
}
```

L'exemple JSON suivant définit une action SQS avec des modèles de substitution dans unAWS IoTRègle

```
{
    "topicRulePayload": {
```

```
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sqs": {
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/
${topic()}",
                    "useBase64": true,
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sqs"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation d'Amazon Simple Queue Service dans le Manuel du développeur Amazon Simple Queue Service

Step Functions

Step Functions (`stepFunctions`) démarre une actionAWS Step FunctionsMachine d'état.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`states:StartExecution`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`stateMachineName`

Nom de la machine d'état Step Functions à démarrer.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`executionNamePrefix`

(Facultatif) Le nom donné à l'exécution de la machine d'état se compose de ce préfixe, suivi d'un UUID. Step Functions crée un nom unique pour chaque exécution de la machine d'état, si aucun n'est fourni.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

ARN du rôle qui accordeAWS IoTAutorisation de démarrer la machine d'état. Pour plus d'informations, consultez Requirements (p. 440).

Prend en chargeModèles de substitution (p. 541) : Non

Examples

L'exemple JSON suivant définit une action Step Functions dans unAWS IoTRègle

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "stepFunctions": {
                    "stateMachineName": "myStateMachine",
                    "executionNamePrefix": "myExecution",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_step_functions"
                }
            }
        ]
    }
}
```

Voir aussi

- Présentation d'AWS Step Functions?dans leAWS Step FunctionsManuel du développeur

Timestream

L'action Règle de flux de temps écrit les attributs (mesures) d'un message MQTT dans une table Amazon Timestream. Pour plus d'informations sur Amazon Timestream, consultezQu'est-ce qu'Amazon Timestream ?.

> **Note**
>
> Amazon Timestream n'est pas disponible dans tous les Région AWS Les Si Amazon Timestream n'est pas disponible dans votre région, il n'apparaît pas dans la liste des actions de règle.

Les attributs que cette règle stocke dans la base de données Timestream sont ceux qui résultent de l'instruction de requête de la règle. La valeur de chaque attribut dans le résultat de l'instruction requête est analysée pour déduire son type de données (comme dans unthe section called “DynamoDBv2” (p. 415)action). La valeur de chaque attribut est écrite dans son propre enregistrement dans la table Timestream. Pour spécifier ou modifier le type de données d'un attribut, utilisez l'option`cast()` (p. 501)dans l'instruction de la requête. Pour plus d'informations sur le contenu de chaque enregistrement de fréquence, consultezthe section called “Contenu de l'enregistrement Amazon Timestream” (p. 443).

> **Note**
>
> Avec SQL V2 (2016-03-23), les valeurs numériques qui sont des nombres entiers, tels que`10.0`, sont convertis leur représentation Integer (`10`). Les lancer explicitement dans un`Decimal`, par exemple en utilisant la propriétécast() (p. 501), n'empêche pas ce comportement : le résultat est toujours une`Integer`Valeur . Cela peut provoquer des erreurs de non-correspondance de type qui empêchent l'enregistrement des données dans la base de données Timestream. Pour traiter de manière fiable les valeurs numériques de nombre entier en tant que`Decimal`, utilisez SQL V1 (2015-10-08) pour l'instruction de requête de règle.

Requirements

Cette action de règle impose les critères suivants :

- Un rôle IAM queAWS IoTpeut supposer d'effectuer le`timestream:DescribeEndpoints`and`timestream:WriteRecords`. Pour plus d'informations, consultez Attribuer à AWS IoT l'accès requis (p. 395).

  DansAWS IoT, vous pouvez choisir, mettre à jour ou créer un rôle pour autoriserAWS IoTPour effectuer cette action de règle.
- Si vous utilisez unAWS Key Management Service(AWS KMS) clé principale client (CMK) client (CMK) client pour chiffrer les données au repos dans Timestream, le service doit avoir l'autorisation d'utiliser la CMK pour le compte du mandataire. Pour de plus amples informations, veuillez consulterCommentAWSUtilisation de servicesAWSKMS.

Parameters

Lorsque vous créez un rôleAWS IoTAvec cette action, vous devez spécifier les informations suivantes :

`databaseName`

Nom d'une base de données Amazon Timestream qui possède la table destinée à recevoir les enregistrements créés par cette action. Voir aussi **`tableName`**.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`dimensions`

Attributs de métadonnées de la série chronologique qui sont écrits dans chaque enregistrement de mesure. Par exemple, le nom et la zone de disponibilité d'une instance EC2 ou le nom du fabricant d'une éolienne sont des dimensions.

`name`

Nom de la dimension de métadonnées. Il s'agit du nom de la colonne dans l'enregistrement de table de base de données.

Les dimensions ne peuvent pas être nommées :`measure_name`,`measure_value`, ou`time`. Ces noms sont réservés. Les noms de dimension ne peuvent pas commencer par`ts_ou`measure_valueet ils ne peuvent pas contenir le deux-points (`:`).

Prend en chargeModèles de substitution (p. 541) : Non

`value`

Valeur à écrire dans cette colonne de l'enregistrement de base de données.

Prend en chargeModèles de substitution (p. 541) : Oui

`roleArn`

Amazon Resource Name (ARN) du rôle qui accordeAWS IoTautorisation d'écrire dans la table de base de données Timestream. Pour plus d'informations, consultez Requirements (p. 442).

Prend en chargeModèles de substitution (p. 541) : Non

`tableName`

Nom de la table de base de données dans laquelle écrire les enregistrements de mesure. Voir aussi **`databaseName`**.

Prend en chargeModèles de substitution (p. 541) : API et APIAWS CLIuniquement

`timestamp`

Valeur à utiliser pour l'horodatage de l'entrée. Si le champ est vide, l'heure à laquelle l'entrée a été traitée est utilisée.

`unit`

Précision de la valeur d'horodatage résultant de l'expression décrite à la section `value`.

Valeurs valides : `SECONDS` |`MILLISECONDS`|`MICROSECONDS`|`NANOSECONDS`. La valeur par défaut est`MILLISECONDS`.

`value`

Expression qui renvoie une valeur temporelle de longue époque.

Vous pouvez utiliser la stratégiethe section called “time_to_epoch (String, String)” (p. 532)pour créer un horodatage valide à partir d'une valeur de date ou d'heure passée dans la charge utile du message.

Contenu de l'enregistrement Amazon Timestream

Les données écrites dans la table Amazon Timestream par cette action incluent un horodatage, des métadonnées de l'action de règle Timestream et le résultat de l'instruction de requête de la règle.

Pour chaque attribut (mesure) du résultat de l'instruction requête, cette action de règle écrit un enregistrement dans la table Timestream spécifiée avec ces colonnes.

Nom de la colonne	Type d'attribut	Valeur	Commentaires
`Nom de taille`	DIMENSION	Valeur spécifiée dans l'entrée d'action de règle de flux de temps.	EACLDimensionspécifié dans l'entrée d'action de règle crée une colonne dans la base de données Timestream avec le nom de la dimension.
nom_mesure	NOM_MESURE	Le nom de l'attribut	Nom de l'attribut dans le résultat de l'instruction de requête dont la valeur est spécifiée dans l'`measure_value::`*`data-type`*column.
mesure_value#*`Type de données`*	MESURE_VALUE	Valeur de l'attribut dans le résultat de l'instruction de requête. Le nom de l'attribut se trouve dans la`measure_name`column.	La valeur est interprétée* et cast comme la meilleure correspondance de :`bigint`,`boolean`,`double`, ou`varchar`. Amazon Timestream crée une colonne distincte pour chaque type de données. La valeur du message peut être convertie en un autre type de données à l'aide de la

Nom de la colonne	Type d'attribut	Valeur	Commentaires
			commande`cast()` (p. 501)dans l'instruction de requête de la règle.
time	TIMESTAMP	Date et heure de l'enregistrement dans la base de données.	Cette valeur est assignée par le moteur de règles ou par le`timestamp`, si elle est définie.

* La valeur d'attribut lue à partir de la charge utile du message est interprétée comme suit. Consultezthe section called “Examples” (p. 444)Pour une illustration de chacun de ces cas.

- Une valeur non citée de`true`ou`false`est interprété comme un`boolean`Type
- Une décimale numérique est interprétée comme un`double`Type
- Une valeur numérique sans point décimal est interprétée comme un`bigint`Type
- Une chaîne entre guillemets est interprétée comme un`varchar`Type
- Les objets et les valeurs de tableau sont convertis en chaînes JSON et stockés en tant que`varchar`Type

Examples

L'exemple JSON suivant définit une action de règle Timestream avec un modèle de substitution dans unAWS IoTRègle

```
{
  "topicRulePayload": {
    "sql": "SELECT * FROM 'iot/topic'",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
      {
        "timestream": {
          "roleArn": "arn:aws:iam::123456789012:role/aws_iot_timestream",
          "tableName": "devices_metrics",
          "dimensions": [
            {
              "name": "device_id",
              "value": "${clientId()}"
            },
            {
              "name": "device_firmware_sku",
              "value": "My Static Metadata"
            }
          ],
          "databaseName": "record_devices"
        }
      }
    ]
  }
}
```

L'utilisation de l'action de règle de rubrique de flux de temps définie dans l'exemple précédent avec la charge utile de message suivante entraîne les enregistrements de Amazon Timestream écrits dans le tableau qui suit.

```
{
  "boolean_value": true,
  "integer_value": 123456789012,
  "double_value": 123.456789012,
  "string_value": "String value",
  "boolean_value_as_string": "true",
  "integer_value_as_string": "123456789012",
  "double_value_as_string": "123.456789012",
  "array_of_integers": [23,36,56,72],
  "array of strings": ["red", "green","blue"],
  "complex_value": {
    "simple_element": 42,
    "array_of_integers": [23,36,56,72],
    "array of strings": ["red", "green","blue"]
  }
}
```

Le tableau suivant affiche les colonnes de base de données et les enregistrements créés à l'aide de l'action de règle de rubrique spécifiée pour traiter la charge utile de message précédente. La `.device_firmware_sku` and`device_id` sont les DIMENSIONS définies dans l'action de règle de rubrique. L'action de règle Rubrique Temps de diffusion crée le `time` et la colonne `measure_name` and `measure_value::*`, qu'il remplit avec les valeurs du résultat de l'instruction de requête de l'action de règle de rubrique.

device_firmw	device_id	nom_mesure	measure_va bigint	measure_va varchar	measure_va double	measure_va booléen	time
Mes métadonnées statiques	IotConsole-159	exemple_valeur_738-0		{"simple_elem ent » :42, "array_of_integers » : [23,36,56,72], "tableau de chaînes » : ["rouge », "vert », "bleu "]}	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-159	integer_value_as_string		123456789012	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-159	exemple_valeur_738-0		-	-	TRUE	20-08-2019
Mes métadonnées statiques	IotConsole-159	exemple_valeur_entière	123456789012	-	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-159	exemple_valeur_738-0		Valeur de chaîne	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-159	array_of_integers		[23,36,56,72]	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-159	exemple_tableau_de_chaînes		["rouge », "vert », "bleu "]	-	-	20-08-2019

device_firmw	device_id	nom_mesure	measure_va bigint	measure_va varchar	measure_va double	measure_va booléen	time
Mes métadonnées statiques	IotConsole-15	example_value_as_string		TRUE	-	-	20-08-2019
Mes métadonnées statiques	IotConsole-15	example_double_as		-	123.456789012		20-08-2019
Mes métadonnées statiques	IotConsole-15	example_value_as_string		123.45679	-	-	20-08-2019

VPC

L'action VPC (Virtual Private Cloud) achemine le trafic vers votre Amazon Virtual Private Cloud (Amazon VPC). Contrairement aux autres actions de règle, l'action VPC ne nécessite aucune configuration et est automatiquement activée lorsque vous spécifiez une destination VPC pour votre action de règle. Une destination VPC contient une liste de sous-réseaux à l'intérieur du VPC. Le moteur de règles crée une elastic network interface dans chaque sous-réseau que vous spécifiez dans cette liste.

> Note
>
> Pour plus d'informations sur les interfaces réseau, consultezInterfaces réseau ElasticDans le Guide de l'utilisateur Amazon EC2.

Actuellement, l'action VPC fonctionne avec les actions de règle suivantes :

- the section called “Apache Kafka” (p. 403)

À des fins de tarification, une action de règle VPC est mesurée en plus de l'action qui envoie un message à une ressource lorsque la ressource se trouve dans votre VPC. Pour en savoir plus sur la tarification, veuillez consulter Tarification AWS IoT Core .

Pour plus d'informations sur la création de destinations VPC, consultezthe section called “Création d'une destination de règle de rubrique VPC” (p. 455).

Résolution des problèmes d'une règle

Si vous rencontrez un problème avec vos règles, vous devez activer CloudWatch Logs. Vous pouvez analyser vos journaux pour déterminer si le problème relève de l'autorisation ou si, par exemple, une condition de clause WHERE n'a pas été respectée. Pour de plus amples informations, veuillez consulterConfiguration des CloudWatch Logs.

Accès à des ressources entre comptes à l'aideAWS IoTRègles

Vous pouvez configurerAWS IoTpour l'accès entre comptes, afin que les données ingérées sur les rubriques MQTT d'un compte puissent être acheminées vers leAWStels qu'Amazon SQS et Lambda, d'un

autre compte. L'exemple suivant illustre la configuration deAWS IoTpour l'ingestion de données entre comptes, à partir d'une rubrique MQTT d'un compte, à une destination dans un autre compte.

Les règles entre comptes peuvent être configurées à l'aide deAutorisations basées sur les ressourcessur la ressource de destination. Par conséquent, seules les destinations qui prennent en charge les autorisations basées sur les ressources peuvent être activées pour l'accès entre comptes avecAWS IoTRègles Les destinations prises en charge sont Amazon SQS, Amazon SNS, Amazon S3 etAWS Lambda.

Prerequisites

- Vous maîtrisezAWS IoTRègles
- Une compréhension de l'IAMUtilisateurs,Rôles, etAutorisation basée sur les ressources
- HAVANTAWS CLIinstallé

Configuration entre comptes pour Amazon SQS

Scénario Le compte A envoie les données à partir d'un message MQTT dans la file d'attente Amazon SQS du compte B.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action de la règle :`sqs:SendMessage`
2222-2222-2222	Compte B	File d'attente Amazon SQS • ARN: *arn:aws:sqs:region:2222-2222-2222:ExampleQueue* • URL :*https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue*

Exécutez les tâches du compte A

> **Note**
>
> Pour exécuter les commandes suivantes, votre utilisateur IAM doit disposer des autorisations pour`iot:CreateTopicRule`avec l'Amazon Resource Name (ARN) de la règle comme ressource, et les autorisations à`iam:PassRole`avec une ressource comme ARN du rôle.

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte A.
2. Création d'un rôle IAM qui approuveAWS IoTet joint une stratégie qui autorise l'accès à la file d'attente Amazon SQS du compte B. Voir les exemples de commandes et de documents de stratégie dansGrantAWS IoTAccès requis.
3. Pour créer une règle attachée à une rubrique, exécutez la commandeCreate-topic-rule, commande.

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

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à l'`iot/test`dans la file d'attente Amazon SQS spécifiée. L'instruction SQL filtre les messages et le rôle accordé à l'ARNAWS IoTAutorisations pour ajouter le message à la file d'attente Amazon SQS.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "actions": [
  {
   "sqs": {
    "queueUrl": "https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role",
    "useBase64": false
   }
  }
 ]
}
```

Pour plus d'informations sur la façon de définir une action Amazon SQS dans unAWS IoT, voirAWS IoTactions de règle - Amazon SQS.

Exécutez les tâches du compte B

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte B.
2. Pour accorder des autorisations à la ressource de file d'attente Amazon SQS sur le compte A, exécutez la commandecommande add-permission.

```
aws sqs add-permission --queue-url https://sqs.region.amazonaws.com/2222-2222-2222/
ExampleQueue --label SendMessagesToMyQueue --aws-account-ids 1111-1111-1111 --actions
 SendMessage
```

Configuration entre comptes pour Amazon SNS

Scénario Le compte A envoie les données à partir d'un message MQTT vers une rubrique Amazon SNS du compte B.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action de la règle :`sns:Publish`
2222-2222-2222	Compte B	ARN de rubrique Amazon SNS :*arn:aws:sns:region:2222-2222-2222:ExampleTopic*

Exécutez les tâches du compte A

Notes

Pour exécuter les commandes suivantes, votre utilisateur IAM doit disposer des autorisations pour`iot:CreateTopicRule`avec ARN de règle en tant que ressource et des autorisations sur le`iam:PassRole`avec une ressource comme ARN de rôle.

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte A.
2. Création d'un rôle IAM qui approuveAWS IoTet joint une stratégie qui autorise l'accès à la rubrique Amazon SNS du compte B. Pour obtenir des exemples de commandes et de documents de stratégie, voirGrantAWS IoTAccès requis.
3. Pour créer une règle attachée à une rubrique, exécutez la commandeCreate-topic-rule, commande.

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

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à l'`iot/test`dans la rubrique Amazon SNS spécifiée. L'instruction SQL filtre les messages et le rôle ARN accorde à l'ARNAWS IoTpour envoyer le message à la rubrique Amazon SNS.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "actions": [
  {
   "sns": {
    "targetArn": "arn:aws:sns:region:2222-2222-2222:ExampleTopic",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role"
   }
  }
 ]
}
```

Pour plus d'informations sur la façon de définir une action Amazon SNS dans unAWS IoT, voirAWS IoTactions de règle - Amazon SNS.

Exécutez les tâches du compte B

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte B.
2. Pour autoriser le compte A sur la ressource de rubrique Amazon SNS, exécutez lecommande add-permission.

```
aws sns add-permission --topic-arn arn:aws:sns:region:2222-2222-2222:ExampleTopic --
label Publish-Permission --aws-account-id 1111-1111-1111 --action-name Publish
```

Configuration entre comptes pour Amazon S3

Scénario Le compte A envoie les données à partir d'un message MQTT vers un compartiment Amazon S3 de compte B.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action de la règle :`s3:PutObject`
2222-2222-2222	Compte B	ARN de compartiment Amazon S3 :*arn:aws:s3:::ExampleBucket*

Exécutez les tâches du compte A

> **Note**
>
> Pour exécuter les commandes suivantes, votre utilisateur IAM doit disposer des autorisations pour`iot:CreateTopicRule`avec l'ARN de règle en tant que ressource et les autorisations pour`iam:PassRole`avec une ressource comme ARN de rôle.

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte A.

2. Créer un rôle IAM qui approuveAWS IoTet attache une stratégie qui autorise l'accès au compartiment Amazon S3 du compte B. Pour obtenir des exemples de commandes et de documents de stratégie, voirGrantAWS IoTAccès requis.
3. Pour créer une règle attachée à votre compartiment S3 cible, exécutez la commandeCreate-topic-rule, commande.

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

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à l'`iot/test`dans le compartiment Amazon S3 spécifié. L'instruction SQL filtre les messages et le rôle ARN accorde à l'ARNAWS IoTAutorisations pour ajouter le message au compartiment Amazon S3.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "actions": [
  {
   "s3": {
    "bucketName": "ExampleBucket",
    "key": "${topic()}/${timestamp()}",
    "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role"
   }
  }
 ]
}
```

Pour plus d'informations sur la façon de définir une action Amazon S3 dans unAWS IoT, voirAWS IoTActions de règle - Amazon S3.

Exécutez les tâches du compte B

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte B.
2. Créez une stratégie de regroupement qui approuve le principal du compte A.

   Voici un exemple de fichier de charge utile qui définit une stratégie de compartiment qui approuve le mandataire d'un autre compte.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "AddCannedAcl",
   "Effect": "Allow",
   "Principal": {
    "AWS": [
     "arn:aws:iam::1111-1111-1111:root"
    ]
   },
   "Action": "s3:PutObject",
   "Resource": "arn:aws:s3:::ExampleBucket/*"
  }
 ]
}
```

   Pour de plus amples informations, veuillez consulterExemples de stratégie de compartiment.
3. Pour attacher la stratégie de compartiment au compartiment spécifié, exécutez la commandePut-bucket-policy, commande.

```
aws s3api put-bucket-policy --bucket ExampleBucket --policy file://./my-bucket-policy.json
```

4. Pour que l'accès entre comptes fonctionne, assurez-vous que vous disposez desBlock all public access (Bloquer tous les accès publics)Paramètres de . Pour de plus amples informations, veuillez consulterBonnes pratiques de sécurité pour Amazon S3.

Configuration entre comptes pourAWS Lambda

Scénario Le compte A appelle unAWS Lambdafonction du compte B, en passant un message MQTT.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action de la règle :`lambda:InvokeFunction`
2222-2222-2222	Compte B	ARN de fonction Lambda : *arn:aws:lambda:region:2222-2222-2222:function:exampl function*

Exécutez les tâches du compte A

Notes

Pour exécuter les commandes suivantes, votre utilisateur IAM doit disposer des autorisations pour`iot:CreateTopicRule`avec ARN de règle en tant que ressource, et les autorisations pour`iam:PassRole`avec la ressource comme ARN de rôle.

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte A.
2. Exécutez leCreate-topic-rule, commandepour créer une règle qui définit l'accès entre comptes à la fonction Lambda du compte B.

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

Voici un exemple de fichier de charge utile avec une règle qui insère tous les messages envoyés à l'`iot/test`dans la fonction Lambda spécifiée. L'instruction SQL filtre les messages et le rôle accordé à l'ARNAWS IoTAutorisation de transmettre les données à la fonction Lambda.

```
{
 "sql": "SELECT * FROM 'iot/test'",
 "ruleDisabled": false,
 "actions": [
  {
   "lambda": {
    "functionArn": "arn:aws:lambda:region:1111-1111-1111:function:example-function"
   }
  }
 ]
}
```

Pour plus d'informations sur la définition d'unAWS LambdaAction dans une actionAWS IoTRègle, lisezAWS IoTActions de règle - Lambda.

Exécutez les tâches du compte B

1. ConfigurationAWS CLIà l'aide de l'utilisateur IAM du compte B.
2. Run (Exécuter Lambda)Commande add-permission de Lambdaà donnerAWS IoTrègle l'autorisation pour déclencher la fonction Lambda. Pour exécuter la commande suivante, votre utilisateur IAM doit avoir l'autorisation de`lambda:AddPermission`action.

```
aws lambda add-permission --function-name example-function --region us-east-1 --
principal iot.amazonaws.com --source-arn arn:aws:iot:region:1111-1111-1111:rule/
example-rule --source-account 1111-1111-1111 --statement-id "unique_id" --action
 "lambda:InvokeFunction"
```

Options:

--principal

Ce champ donne l'autorisation deAWS IoT(représenté par`iot.amazonaws.com`) pour appeler la fonction Lambda.

--source-arn

Ce champ garantit que seul`arn:aws:iot:region:1111-1111-1111:rule/example-rule`inAWS IoTdéclenche cette fonction Lambda et aucune autre règle dans le même compte ou différent ne peut déclencher cette fonction Lambda.

--source-account

Ce champ garantit queAWS IoTdéclenche cette fonction Lambda uniquement pour le compte de`1111-1111-1111`.

> Notes
>
> Si un message d'erreur « La règle n'a pas pu être trouvée » s'affiche depuis votreAWS LambdasousConfiguration, ignorez le message d'erreur et procédez au test de la connexion.

Gestion des erreurs (action d'erreur)

Lorsqu'AWS IoT reçoit un message d'un appareil, le moteur de règles vérifie si ce message correspond à une règle. Si tel est le cas, l'instruction de requête de la règle est évaluée et les actions de la règle sont déclenchées, le résultat de l'instruction de requête étant alors transmis.

Si un problème se produit lors du déclenchement d'une action, le moteur de règles déclenche une action d'erreur, si une telle action est spécifiée pour la règle. Cela peut se produire dans les cas suivants :

- Une règle n'a pas l'autorisation d'accéder à un compartiment Amazon S3.
- Un erreur de l'utilisateur entraîne un dépassement du débit alloué à DynamoDB.

Format du message d'action d'erreur

Un seul message est généré par la règle et par message. Par exemple, si deux actions de règle échouent dans une même règle, l'action d'erreur reçoit un message contenant les deux erreurs.

Le message d'action d'erreur ressemble à l'exemple suivant.

```
{
```

```
  "ruleName": "TestAction",
  "topic": "testme/action",
  "cloudwatchTraceId": "7e146a2c-95b5-6caf-98b9-50e3969734c7",
  "clientId": "iotconsole-1511213971966-0",
  "base64OriginalPayload": "ewogICJtZXNzYWdlIjogIkhlbGxvIHZyb20gQVdTIElvVCBjb25zb2xlIgp9",
  "failures": [
    {
      "failedAction": "S3Action",
      "failedResource": "us-east-1-s3-verify-user",
      "errorMessage": "Failed to put S3 object. The error received was The
 specified bucket does not exist (Service: Amazon S3; Status Code: 404; Error
 Code: NoSuchBucket; Request ID: 9DF5416B9B47B9AF; S3 Extended Request ID:
 yMah1cwPhqTH267QLPhTKeVPKJB8BO5ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y=).
 Message arrived on: error/action, Action: s3, Bucket: us-
east-1-s3-verify-user, Key: \"aaa\". Value of x-amz-id-2:
 yMah1cwPhqTH267QLPhTKeVPKJB8BO5ndBHzOmWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y="
    }
  ]
}
```

ruleName

Nom de la règle qui a déclenché l'action d'erreur.

topic

Rubrique dans laquelle le message d'origine a été reçu.

cloudwatchTraceId

Identité unique faisant référence aux journaux d'erreurs dans CloudWatch.

clientId

ID client de l'éditeur du message.

base64OriginalPayload

Charge utile du message d'origine codée en base64.

échecs

failedAction

Nom de l'action qui a échoué, par exemple « S3Action ».

failedResource

Nom de la ressource. Par exemple, le nom d'un compartiment S3.

errorMessage

Description et explication de l'erreur.

Exemple d'action d'erreur

Voici un exemple de règle à laquelle a été ajoutée une action d'erreur. La règle suivante comporte une action qui écrit des données de message dans une table DynamoDB et une action d'erreur qui écrit des données dans un compartiment Amazon S3 :

```
{
    "sql" : "SELECT * FROM ..."
    "actions" : [{
        "dynamoDB" : {
            "table" : "PoorlyConfiguredTable",
            "hashKeyField" : "AConstantString",
```

```
            "hashKeyValue" : "AHashKey"}}
    ],
    "errorAction" : {
        "s3" : {
            "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",
            "bucket" : "message-processing-errors",
            "key" : "${replace(topic(), '/', '-') + '-' + timestamp() + '-' + newuuid()}"
        }
    }
}
```

Vous pouvez utiliser n'importe quelle fonction ou substitution dans une instruction SQL d'action d'erreur, sauf pour les fonctions externes (par exemple, `get_thing_shadow`, `aws_lambda`, et `machinelearning_predict`.)

Pour plus d'informations sur les règles et sur la manière de spécifier une action d'erreur, consultez Création d'une règle AWS IoT.

Pour plus d'informations sur l'utilisation de CloudWatch pour superviser la réussite ou l'échec des règles, consultez.Métriques et dimensions d’AWS IoT (p. 363).

Utilisation des destinations des règles de rubrique

Une destination est une ressource qui définit l'emplacement où le moteur de règles peut acheminer les données. La .AWS IoTprend en charge deux types de destinations : Destinations HTTP et destinations VPC. Les destinations permettent au moteur de règles d'envoyer des données à d'autres services qui ne sont pas intégrés à AWS IoT de manière native. Une destination peut être réutilisée entre les règles.

Les destinations HTTP peuvent nécessiter une confirmation ou une configuration avant de pouvoir être utilisées. Les paragraphes suivants décrivent le fonctionnement du flux de confirmation pour les destinations HTTP.

Les destinations des règles de rubrique permettent de vérifier que vous possédez l'accès ou avez accès au point de terminaison vers lequel vous souhaitez acheminer les données. Lorsque vous créez une action de règle avec un point de terminaison HTTP (par exemple, un`http`) ou mettre à jour le point de terminaison d'une action de règle existante,AWS IoTenvoie un message de confirmation au point de terminaison qui contient un jeton unique. Pour vérifier votre point de terminaison, vous devez appeler `ConfirmTopicRuleDestination` avec le jeton pour vérifier que vous possédez ou avez accès au point de terminaison. Le jeton est valide pendant trois jours. Passé ce délai, vous devrez créer une nouvelle action `http` ou appeler l'API `UpdateTopicRuleDestination` pour redémarrer le processus de confirmation. Vous pouvez également confirmer la destination de la règle de rubrique en accédant à `enableUrl` ou en fournissant le jeton à partir de la page Destination de la console AWS IoT.

QuandAWS IoTreçoit le jeton envoyé à votre point de terminaison, la destination HTTP est confirmée. Vous devez activer la destination avant qu'elle puisse être utilisée par le moteur de règles. L'état d'une destination peut être :

ENABLED

La destination est activée. L'état d'une destination doit être `ENABLED` (ACTIVÉ) pour qu'elle soit utilisée dans une règle. Vous pouvez uniquement activer une destination dont l'état est DISABLED (DÉSACTIVÉ).

DISABLED

La destination a été confirmée, mais désactivée. Cet état est utile si vous souhaitez empêcher temporairement le trafic vers votre point de terminaison sans avoir à passer à nouveau par le processus de confirmation. Vous pouvez uniquement désactiver une destination dont l'état est ENABLED (ACTIVÉ).

IN_PROGRESS

La confirmation de la destination est en cours.

ERROR

La confirmation de la destination a expiré.

Une fois que les destinations sont confirmées et activées, elles peuvent être utilisées avec n'importe quelle règle de votre compte.

Création d'une destination de règle de rubrique HTTP

Une destination est créée lorsque vous utilisez AWS IoT pour créer une règle avec une action `http`. Vous pouvez également créer une destination à l'aide de l'API `CreateTopicRuleDestination` ou de la console AWS IoT.

Lorsque vous créez une destination, AWS IoT vérifie que l'URL du point de terminaison est valide. Si l'URL est valide, un message de confirmation est envoyé à cette URL. Le format de la demande de confirmation est le suivant :

```
HTTP POST {confirmationUrl}/?confirmationToken={confirmationToken}
Headers:
x-amz-rules-engine-message-type: DestinationConfirmation
x-amz-rules-engine-destination-arn:"arn:aws:iot:us-east-1:123456789012:ruledestination/
http/7a280e37-b9c6-47a2-a751-0703693f46e4"
Content-Type: application/json
Body:
{
    "arn":"arn:aws:iot:us-east-1:123456789012:ruledestination/http/7a280e37-b9c6-47a2-
a751-0703693f46e4",
    "confirmationToken": "AYADeMXLrPrNY2wqJAKsFNn-…NBJndA",
    "enableUrl": "https://iot.us-east-1.amazonaws.com/confirmdestination/
AYADeMXLrPrNY2wqJAKsFNn-…NBJndA",
    "messageType": "DestinationConfirmation"
}
```

Les champs de ce message sont définis comme suit :

arn

Amazon Resource Name (ARN) de la destination de la règle de rubrique à confirmer.

confirmationToken

Le jeton de confirmation. Dans l'exemple, le jeton est tronqué. Votre jeton sera plus long.

enableUrl

L'URL à laquelle vous accédez pour confirmer la destination d'une règle de rubrique.

messageType

Type du message.

Création d'une destination de règle de rubrique VPC

Vous créez une destination de Virtual Private Cloud (VPC) à l'aide de l'outilCreateTopicRuleDestinationou l'API AWS IoT Core console

Note

Les destinations des règles de rubrique VPC qui ne reçoivent aucun trafic pendant 30 jours d'affilée seront désactivées.

Lorsque vous créez une destination VPC, vous devez spécifier les informations suivantes.

vpcId

ID unique de la destination VPC.

subnetIds

Liste des sous-réseaux dans lesquels le moteur de règles crée des interfaces réseau élastiques. Le moteur de règles alloue une interface réseau unique pour chaque sous-réseau de la liste.

SecurityGroups (facultatif)

Une liste des groupes de sécurité à appliquer aux interfaces réseau.

roleArn

ARN d'un rôle autorisé à créer des interfaces réseau en votre nom.

Cet ARN doit avoir une stratégie qui ressemble à l'exemple suivant.

```
{
    "Version": "2012-10-17",
    "Statement": [
    {
        "Effect": "Allow",
        "Action": [
            "ec2:CreateNetworkInterface",
            "ec2:DescribeNetworkInterfaces",
            "ec2:CreateNetworkInterfacePermission",
            "ec2:DeleteNetworkInterface",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs",
            "ec2:DescribeVpcAttribute"
            ],
            "Resource": "*"
        }
    ]
}
```

Création d'une destination VPC à l'aide deAWS CLI

L'exemple suivant montre comment créer une destination VPC à l'aide de l'AWS CLI.

```
aws --region regions iot create-topic-rule-destination --destination-configuration
 'vpcConfiguration={subnetIds=["subnet-123456789101230456"],securityGroups=[],vpcId="vpc-
123456789101230456",roleArn="arn:aws:iam::123456789012:role/role-name"}'
```

Une fois que vous avez exécuté cette commande, l'état de destination du VPC sera`IN PROGRESS`. Après quelques minutes, son statut passe à`ERROR`(si la commande ne réussit pas) ou`ENABLED`. Lorsque le statut de destination est`ENABLED`, elle est prête à être utilisée.

Vous pouvez utiliser la commande suivante pour obtenir votre destination VPC.

```
aws --region region iot get-topic-rule-destination --arn "VPCDestinationARN"
```

Création d'une destination VPC à l'aide de AWS IoT Core console

Les étapes suivantes décrivent comment créer une destination VPC à l'aide de l' AWS IoT Core console

1. Accédez à la console AWS IoT Core . ChoisissezDestinationsen vertu de laLoidans le panneau de gauche.
2. Entrez des valeurs pour les champs suivants.
   - ID de VPC
   - ID de sous-réseau (sub
   - Security Group
3. Sélectionnez un rôle qui a les autorisations requises pour créer les IN. L'exemple de rôle précédent contient ces autorisations.

Lorsque l'état de destination du VPC estENABLED, elle est prête à être utilisée.

Confirmation d'une destination de règle de rubrique HTTP

Vous pouvez confirmer une destination en envoyant une requête HTTP GET à `enableUrl` qui est inclus dans le message de confirmation. Vous pouvez appeler `ConfirmTopicRuleDestination` avec le jeton à partir du message de confirmation ou vous pouvez utiliser la console AWS IoT.

Le jeton doit être valide pour que la destination soit placée à l'état `ENABLED`. Si le jeton a expiré, vous devez appeler `UpdateTopicRuleDestination` pour redémarrer le processus de confirmation.

> Note
>
> Si vous confirmez la destination de règle de rubrique en appelant`ConfirmTopicRuleDestination`, le statut de destination sera`DISABLED`et vous devez activer explicitement votre destination en appelant`UpdateTopicRuleDestination`avant de l'utiliser.

Désactivation d'une destination de règle de rubrique

Pour désactiver une destination, appelez `UpdateTopicRuleDestination` et définissez l'état de la destination de règle de rubrique sur `DISABLED`.

Activation d'une destination de règle de rubrique

Pour activer une destination, appelez `UpdateTopicRuleDestination` et définissez l'état de la règle de rubrique sur `ENABLED`. Vous n'avez pas besoin de valider à nouveau l'URL.

Envoi d'un nouveau message de confirmation

Pour déclencher un nouveau message de confirmation pour une destination, appelez `UpdateTopicRuleDestination` et définissez l'état de la destination de règle de rubrique sur `IN_PROGRESS`.

Suppression d'une destination de règle de rubrique

Pour supprimer une destination de règle de rubrique, appelez `DeleteTopicRuleDestination`.

Autorités de certification prises en charge par les points de terminaison HTTPS dans les destinations de

Les autorités de certification suivantes sont prises en charge par les points de terminaison HTTPS dans les destinations de règles de rubrique.

```
Alias name: swisssignplatinumg2ca
Certificate fingerprints:
  MD5:  C9:98:27:77:28:1E:3D:0E:15:3C:84:00:B8:85:03:E6
  SHA1: 56:E0:FA:C0:3B:8F:18:23:55:18:E5:D3:11:CA:E8:C2:43:31:AB:66
  SHA256:
 3B:22:2E:56:67:11:E9:92:30:0D:C0:B1:5A:B9:47:3D:AF:DE:F8:C8:4D:0C:EF:7D:33:17:B4:C1:82:1D:14:36

Alias name: hellenicacademicandresearchinstitutionsrootca2011
Certificate fingerprints:
  MD5:  73:9F:4C:4B:73:5B:79:E9:FA:BA:1C:EF:6E:CB:D5:C9
  SHA1: FE:45:65:9B:79:03:5B:98:A1:61:B5:51:2E:AC:DA:58:09:48:22:4D
  SHA256:
 BC:10:4F:15:A4:8B:E7:09:DC:A5:42:A7:E1:D4:B9:DF:6F:05:45:27:E8:02:EA:A9:2D:59:54:44:25:8A:FE:71

Alias name: teliasonerarootcav1
Certificate fingerprints:
  MD5:  37:41:49:1B:18:56:9A:26:F5:AD:C2:66:FB:40:A5:4C
  SHA1: 43:13:BB:96:F1:D5:86:9B:C1:4E:6A:92:F6:CF:F6:34:69:87:82:37
  SHA256:
 DD:69:36:FE:21:F8:F0:77:C1:23:A1:A5:21:C1:22:24:F7:22:55:B7:3E:03:A7:26:06:93:E8:A2:4B:0F:A3:89

Alias name: geotrustprimarycertificationauthority
Certificate fingerprints:
  MD5:  02:26:C3:01:5E:08:30:37:43:A9:D0:7D:CF:37:E6:BF
  SHA1: 32:3C:11:8E:1B:F7:B8:B6:52:54:E2:E2:10:0D:D6:02:90:37:F0:96
  SHA256:
 37:D5:10:06:C5:12:EA:AB:62:64:21:F1:EC:8C:92:01:3F:C5:F8:2A:E9:8E:E5:33:EB:46:19:B8:DE:B4:D0:6C

Alias name: trustisfpsrootca
Certificate fingerprints:
  MD5:  30:C9:E7:1E:6B:E6:14:EB:65:B2:16:69:20:31:67:4D
  SHA1: 3B:C0:38:0B:33:C3:F6:A6:0C:86:15:22:93:D9:DF:F5:4B:81:C0:04
  SHA256:
 C1:B4:82:99:AB:A5:20:8F:E9:63:0A:CE:55:CA:68:A0:3E:DA:5A:51:9C:88:02:A0:D3:A6:73:BE:8F:8E:55:7D

Alias name: quovadisrootca3g3
Certificate fingerprints:
  MD5:  DF:7D:B9:AD:54:6F:68:A1:DF:89:57:03:97:43:B0:D7
  SHA1: 48:12:BD:92:3C:A8:C4:39:06:E7:30:6D:27:96:E6:A4:CF:22:2E:7D
  SHA256:
 88:EF:81:DE:20:2E:B0:18:45:2E:43:F8:64:72:5C:EA:5F:BD:1F:C2:D9:D2:05:73:07:09:C5:D8:B8:69:0F:46

Alias name: buypassclass2ca
Certificate fingerprints:
  MD5:  46:A7:D2:FE:45:FB:64:5A:A8:59:90:9B:78:44:9B:29
  SHA1: 49:0A:75:74:DE:87:0A:47:FE:58:EE:F6:C7:6B:EB:C6:0B:12:40:99
  SHA256:
 9A:11:40:25:19:7C:5B:B9:5D:94:E6:3D:55:CD:43:79:08:47:B6:46:B2:3C:DF:11:AD:A4:A0:0E:FF:15:FB:48

Alias name: secureglobalca
```

```
Certificate fingerprints:
  MD5:  CF:F4:27:0D:D4:ED:DC:65:16:49:6D:3D:DA:BF:6E:DE
  SHA1: 3A:44:73:5A:E5:81:90:1F:24:86:61:46:1E:3B:9C:C4:5F:F5:3A:1B
  SHA256:
 42:00:F5:04:3A:C8:59:0E:BB:52:7D:20:9E:D1:50:30:29:FB:CB:D4:1C:A1:B5:06:EC:27:F1:5A:DE:7D:AC:69

Alias name: chunghwaepkirootca
Certificate fingerprints:
  MD5:  1B:2E:00:CA:26:06:90:3D:AD:FE:6F:15:68:D3:6B:B3
  SHA1: 67:65:0D:F1:7E:8E:7E:5B:82:40:A4:F4:56:4B:CF:E2:3D:69:C6:F0
  SHA256:
 C0:A6:F4:DC:63:A2:4B:FD:CF:54:EF:2A:6A:08:2A:0A:72:DE:35:80:3E:2F:F5:FF:52:7A:E5:D8:72:06:DF:D5

Alias name: verisignclass2g2ca
Certificate fingerprints:
  MD5:  2D:BB:E5:25:D3:D1:65:82:3A:B7:0E:FA:E6:EB:E2:E1
  SHA1: B3:EA:C4:47:76:C9:C8:1C:EA:F2:9D:95:B6:CC:A0:08:1B:67:EC:9D
  SHA256:
 3A:43:E2:20:FE:7F:3E:A9:65:3D:1E:21:74:2E:AC:2B:75:C2:0F:D8:98:03:05:BC:50:2C:AF:8C:2D:9B:41:A1

Alias name: szafirrootca2
Certificate fingerprints:
  MD5:  11:64:C1:89:B0:24:B1:8C:B1:07:7E:89:9E:51:9E:99
  SHA1: E2:52:FA:95:3F:ED:DB:24:60:BD:6E:28:F3:9C:CC:CF:5E:B3:3F:DE
  SHA256:
 A1:33:9D:33:28:1A:0B:56:E5:57:D3:D3:2B:1C:E7:F9:36:7E:B0:94:BD:5F:A7:2A:7E:50:04:C8:DE:D7:CA:FE

Alias name: quovadisrootca1g3
Certificate fingerprints:
  MD5:  A4:BC:5B:3F:FE:37:9A:FA:64:F0:E2:FA:05:3D:0B:AB
  SHA1: 1B:8E:EA:57:96:29:1A:C9:39:EA:B8:0A:81:1A:73:73:C0:93:79:67
  SHA256:
 8A:86:6F:D1:B2:76:B5:7E:57:8E:92:1C:65:82:8A:2B:ED:58:E9:F2:F2:88:05:41:34:B7:F1:F4:BF:C9:CC:74

Alias name: utndatacorpsgcca
Certificate fingerprints:
  MD5:  B3:A5:3E:77:21:6D:AC:4A:C0:C9:FB:D5:41:3D:CA:06
  SHA1: 58:11:9F:0E:12:82:87:EA:50:FD:D9:87:45:6F:4F:78:DC:FA:D6:D4
  SHA256:
 85:FB:2F:91:DD:12:27:5A:01:45:B6:36:53:4F:84:02:4A:D6:8B:69:B8:EE:88:68:4F:F7:11:37:58:05:B3:48

Alias name: autoridaddecertificacionfirmaprofesionalcifa62634068
Certificate fingerprints:
  MD5:  73:3A:74:7A:EC:BB:A3:96:A6:C2:E4:E2:C8:9B:C0:C3
  SHA1: AE:C5:FB:3F:C8:E1:BF:C4:E5:4F:03:07:5A:9A:E8:00:B7:F7:B6:FA
  SHA256:
 04:04:80:28:BF:1F:28:64:D4:8F:9A:D4:D8:32:94:36:6A:82:88:56:55:3F:3B:14:30:3F:90:14:7F:5D:40:EF

Alias name: securesignrootca11
Certificate fingerprints:
  MD5:  B7:52:74:E2:92:B4:80:93:F2:75:E4:CC:D7:F2:EA:26
  SHA1: 3B:C4:9F:48:F8:F3:73:A0:9C:1E:BD:F8:5B:B1:C3:65:C7:D8:11:B3
  SHA256:
 BF:0F:EE:FB:9E:3A:58:1A:D5:F9:E9:DB:75:89:98:57:43:D2:61:08:5C:4D:31:4F:6F:5D:72:59:AA:42:16:12

Alias name: amazon-ca-g4-acm2
Certificate fingerprints:
  MD5:  B2:F1:03:2B:93:64:05:80:B8:A8:17:36:B9:1B:52:3C
  SHA1: A7:E6:45:32:1F:7A:B7:AD:C0:70:EA:73:5F:AB:ED:C3:DA:B4:D0:C8
  SHA256:
 D7:A8:7C:69:95:D0:E2:04:2A:32:70:A7:E2:87:FE:A7:E8:F4:C1:70:62:F7:90:C3:EB:BB:53:F2:AC:39:26:BE

Alias name: isrgrootx1
Certificate fingerprints:
  MD5:  0C:D2:F9:E0:DA:17:73:E9:ED:86:4D:A5:E3:70:E7:4E
  SHA1: CA:BD:2A:79:A1:07:6A:31:F2:1D:25:36:35:CB:03:9D:43:29:A5:E8
```

```
  SHA256:
 96:BC:EC:06:26:49:76:F3:74:60:77:9A:CF:28:C5:A7:CF:E8:A3:C0:AA:E1:1A:8F:FC:EE:05:C0:BD:DF:08:C6

Alias name: amazon-ca-g4-acm1
Certificate fingerprints:
  MD5:  E2:F1:18:19:61:5C:43:E0:D4:A8:5D:0B:FA:7C:89:1B
  SHA1: F2:0D:28:B6:29:C2:2C:5E:84:05:E6:02:4D:97:FE:8F:A0:84:93:A0
  SHA256:
 B0:11:A4:F7:29:6C:74:D8:2B:F5:62:DF:87:D7:28:C7:1F:B5:8C:F4:E6:73:F2:78:FC:DA:F3:FF:83:A6:8C:87

Alias name: etugracertificationauthority
Certificate fingerprints:
  MD5:  B8:A1:03:63:B0:BD:21:71:70:8A:6F:13:3A:BB:79:49
  SHA1: 51:C6:E7:08:49:06:6E:F3:92:D4:5C:A0:0D:6D:A3:62:8F:C3:52:39
  SHA256:
 B0:BF:D5:2B:B0:D7:D9:BD:92:BF:5D:4D:C1:3D:A2:55:C0:2C:54:2F:37:83:65:EA:89:39:11:F5:5E:55:F2:3C

Alias name: geotrustuniversalca2
Certificate fingerprints:
  MD5:  34:FC:B8:D0:36:DB:9E:14:B3:C2:F2:DB:8F:E4:94:C7
  SHA1: 37:9A:19:7B:41:85:45:35:0C:A6:03:69:F3:3C:2E:AF:47:4F:20:79
  SHA256:
 A0:23:4F:3B:C8:52:7C:A5:62:8E:EC:81:AD:5D:69:89:5D:A5:68:0D:C9:1D:1C:B8:47:7F:33:F8:78:B9:5B:0B

Alias name: digicertglobalrootca
Certificate fingerprints:
  MD5:  79:E4:A9:84:0D:7D:3A:96:D7:C0:4F:E2:43:4C:89:2E
  SHA1: A8:98:5D:3A:65:E5:E5:C4:B2:D7:D6:6D:40:C6:DD:2F:B1:9C:54:36
  SHA256:
 43:48:A0:E9:44:4C:78:CB:26:5E:05:8D:5E:89:44:B4:D8:4F:96:62:BD:26:DB:25:7F:89:34:A4:43:C7:01:61

Alias name: staatdernederlandenevrootca
Certificate fingerprints:
  MD5:  FC:06:AF:7B:E8:1A:F1:9A:B4:E8:D2:70:1F:C0:F5:BA
  SHA1: 76:E2:7E:C1:4F:DB:82:C1:C0:A6:75:B5:05:BE:3D:29:B4:ED:DB:BB
  SHA256:
 4D:24:91:41:4C:FE:95:67:46:EC:4C:EF:A6:CF:6F:72:E2:8A:13:29:43:2F:9D:8A:90:7A:C4:CB:5D:AD:C1:5A

Alias name: utnuserfirstclientauthemailca
Certificate fingerprints:
  MD5:  D7:34:3D:EF:1D:27:09:28:E1:31:02:5B:13:2B:DD:F7
  SHA1: B1:72:B1:A5:6D:95:F9:1F:E5:02:87:E1:4D:37:EA:6A:44:63:76:8A
  SHA256:
 43:F2:57:41:2D:44:0D:62:74:76:97:4F:87:7D:A8:F1:FC:24:44:56:5A:36:7A:E6:0E:DD:C2:7A:41:25:31:AE

Alias name: actalisauthenticationrootca
Certificate fingerprints:
  MD5:  69:C1:0D:4F:07:A3:1B:C3:FE:56:3D:04:BC:11:F6:A6
  SHA1: F3:73:B3:87:06:5A:28:84:8A:F2:F3:4A:CE:19:2B:DD:C7:8E:9C:AC
  SHA256:
 55:92:60:84:EC:96:3A:64:B9:6E:2A:BE:01:CE:0B:A8:6A:64:FB:FE:BC:C7:AA:B5:AF:C1:55:B3:7F:D7:60:66

Alias name: amazonrootca4
Certificate fingerprints:
  MD5:  89:BC:27:D5:EB:17:8D:06:6A:69:D5:FD:89:47:B4:CD
  SHA1: F6:10:84:07:D6:F8:BB:67:98:0C:C2:E2:44:C2:EB:AE:1C:EF:63:BE
  SHA256:
 E3:5D:28:41:9E:D0:20:25:CF:A6:90:38:CD:62:39:62:45:8D:A5:C6:95:FB:DE:A3:C2:2B:0B:FB:25:89:70:92

Alias name: amazonrootca3
Certificate fingerprints:
  MD5:  A0:D4:EF:0B:F7:B5:D8:49:95:2A:EC:F5:C4:FC:81:87
  SHA1: 0D:44:DD:8C:3C:8C:1A:1A:58:75:64:81:E9:0F:2E:2A:FF:B3:D2:6E
  SHA256:
 18:CE:6C:FE:7B:F1:4E:60:B2:E3:47:B8:DF:E8:68:CB:31:D0:2E:BB:3A:DA:27:15:69:F5:03:43:B4:6D:B3:A4
```

```
Alias name: amazonrootca2
Certificate fingerprints:
  MD5:  C8:E5:8D:CE:A8:42:E2:7A:C0:2A:5C:7C:9E:26:BF:66
  SHA1: 5A:8C:EF:45:D7:A6:98:59:76:7A:8C:8B:44:96:B5:78:CF:47:4B:1A
  SHA256:
 1B:A5:B2:AA:8C:65:40:1A:82:96:01:18:F8:0B:EC:4F:62:30:4D:83:CE:C4:71:3A:19:C3:9C:01:1E:A4:6D:B4

Alias name: amazonrootca1
Certificate fingerprints:
  MD5:  43:C6:BF:AE:EC:FE:AD:2F:18:C6:88:68:30:FC:C8:E6
  SHA1: 8D:A7:F9:65:EC:5E:FC:37:91:0F:1C:6E:59:FD:C1:CC:6A:6E:DE:16
  SHA256:
 8E:CD:E6:88:4F:3D:87:B1:12:5B:A3:1A:C3:FC:B1:3D:70:16:DE:7F:57:CC:90:4F:E1:CB:97:C6:AE:98:19:6E

Alias name: affirmtrustpremium
Certificate fingerprints:
  MD5:  C4:5D:0E:48:B6:AC:28:30:4E:0A:BC:F9:38:16:87:57
  SHA1: D8:A6:33:2C:E0:03:6F:B1:85:F6:63:4F:7D:6A:06:65:26:32:28:27
  SHA256:
 70:A7:3F:7F:37:6B:60:07:42:48:90:45:34:B1:14:82:D5:BF:0E:69:8E:CC:49:8D:F5:25:77:EB:F2:E9:3B:9A

Alias name: keynectisrootca
Certificate fingerprints:
  MD5:  CC:4D:AE:FB:30:6B:D8:38:FE:50:EB:86:61:4B:D2:26
  SHA1: 9C:61:5C:4D:4D:85:10:3A:53:26:C2:4D:BA:EA:E4:A2:D2:D5:CC:97
  SHA256:
 42:10:F1:99:49:9A:9A:C3:3C:8D:E0:2B:A6:DB:AA:14:40:8B:DD:8A:6E:32:46:89:C1:92:2D:06:97:15:A3:32

Alias name: equifaxsecureglobalebusinessca1
Certificate fingerprints:
  MD5:  51:F0:2A:33:F1:F5:55:39:07:F2:16:7A:47:C7:5D:63
  SHA1: 3A:74:CB:7A:47:DB:70:DE:89:1F:24:35:98:64:B8:2D:82:BD:1A:36
  SHA256:
 86:AB:5A:65:71:D3:32:9A:BC:D2:E4:E6:37:66:8B:A8:9C:73:1E:C2:93:B6:CB:A6:0F:71:63:40:A0:91:CE:AE

Alias name: affirmtrustpremiumca
Certificate fingerprints:
  MD5:  C4:5D:0E:48:B6:AC:28:30:4E:0A:BC:F9:38:16:87:57
  SHA1: D8:A6:33:2C:E0:03:6F:B1:85:F6:63:4F:7D:6A:06:65:26:32:28:27
  SHA256:
 70:A7:3F:7F:37:6B:60:07:42:48:90:45:34:B1:14:82:D5:BF:0E:69:8E:CC:49:8D:F5:25:77:EB:F2:E9:3B:9A

Alias name: baltimorecodesigningca
Certificate fingerprints:
  MD5:  90:F5:28:49:56:D1:5D:2C:B0:53:D4:4B:EF:6F:90:22
  SHA1: 30:46:D8:C8:88:FF:69:30:C3:4A:FC:CD:49:27:08:7C:60:56:7B:0D
  SHA256:
 A9:15:45:DB:D2:E1:9C:4C:CD:F9:09:AA:71:90:0D:18:C7:35:1C:89:B3:15:F0:F1:3D:05:C1:3A:8F:FB:46:87

Alias name: gdcatrustauthr5root
Certificate fingerprints:
  MD5:  63:CC:D9:3D:34:35:5C:6F:53:A3:E2:08:70:48:1F:B4
  SHA1: 0F:36:38:5B:81:1A:25:C3:9B:31:4E:83:CA:E9:34:66:70:CC:74:B4
  SHA256:
 BF:FF:8F:D0:44:33:48:7D:6A:8A:A6:0C:1A:29:76:7A:9F:C2:BB:B0:5E:42:0F:71:3A:13:B9:92:89:1D:38:93

Alias name: certinomisrootca
Certificate fingerprints:
  MD5:  14:0A:FD:8D:A8:28:B5:38:69:DB:56:7E:61:22:03:3F
  SHA1: 9D:70:BB:01:A5:A4:A0:18:11:2E:F7:1C:01:B9:32:C5:34:E7:88:A8
  SHA256:
 2A:99:F5:BC:11:74:B7:3C:BB:1D:62:08:84:E0:1C:34:E5:1C:CB:39:78:DA:12:5F:0E:33:26:88:83:BF:41:58

Alias name: verisignclass3publicprimarycertificationauthorityg5
Certificate fingerprints:
  MD5:  CB:17:E4:31:67:3E:E2:09:FE:45:57:93:F3:0A:FA:1C
```

```
  SHA1: 4E:B6:D5:78:49:9B:1C:CF:5F:58:1E:AD:56:BE:3D:9B:67:44:A5:E5
  SHA256:
 9A:CF:AB:7E:43:C8:D8:80:D0:6B:26:2A:94:DE:EE:E4:B4:65:99:89:C3:D0:CA:F1:9B:AF:64:05:E4:1A:B7:DF

Alias name: verisignclass3publicprimarycertificationauthorityg4
Certificate fingerprints:
  MD5:  3A:52:E1:E7:FD:6F:3A:E3:6F:F3:6F:99:1B:F9:22:41
  SHA1: 22:D5:D8:DF:8F:02:31:D1:8D:F7:9D:B7:CF:8A:2D:64:C9:3F:6C:3A
  SHA256:
 69:DD:D7:EA:90:BB:57:C9:3E:13:5D:C8:5E:A6:FC:D5:48:0B:60:32:39:BD:C4:54:FC:75:8B:2A:26:CF:7F:79

Alias name: verisignclass3publicprimarycertificationauthorityg3
Certificate fingerprints:
  MD5:  CD:68:B6:A7:C7:C4:CE:75:E0:1D:4F:57:44:61:92:09
  SHA1: 13:2D:0D:45:53:4B:69:97:CD:B2:D5:C3:39:E2:55:76:60:9B:5C:C6
  SHA256:
 EB:04:CF:5E:B1:F3:9A:FA:76:2F:2B:B1:20:F2:96:CB:A5:20:C1:B9:7D:B1:58:95:65:B8:1C:B9:A1:7B:72:44

Alias name: swisssignsilverg2ca
Certificate fingerprints:
  MD5:  E0:06:A1:C9:7D:CF:C9:FC:0D:C0:56:75:96:D8:62:13
  SHA1: 9B:AA:E5:9F:56:EE:21:CB:43:5A:BE:25:93:DF:A7:F0:40:D1:1D:CB
  SHA256:
 BE:6C:4D:A2:BB:B9:BA:59:B6:F3:93:97:68:37:42:46:C3:C0:05:99:3F:A9:8F:02:0D:1D:ED:BE:D4:8A:81:D5

Alias name: swisssignsilvercag2
Certificate fingerprints:
  MD5:  E0:06:A1:C9:7D:CF:C9:FC:0D:C0:56:75:96:D8:62:13
  SHA1: 9B:AA:E5:9F:56:EE:21:CB:43:5A:BE:25:93:DF:A7:F0:40:D1:1D:CB
  SHA256:
 BE:6C:4D:A2:BB:B9:BA:59:B6:F3:93:97:68:37:42:46:C3:C0:05:99:3F:A9:8F:02:0D:1D:ED:BE:D4:8A:81:D5

Alias name: atostrustedroot2011
Certificate fingerprints:
  MD5:  AE:B9:C4:32:4B:AC:7F:5D:66:CC:77:94:BB:2A:77:56
  SHA1: 2B:B1:F5:3E:55:0C:1D:C5:F1:D4:E6:B7:6A:46:4B:55:06:02:AC:21
  SHA256:
 F3:56:BE:A2:44:B7:A9:1E:B3:5D:53:CA:9A:D7:86:4A:CE:01:8E:2D:35:D5:F8:F9:6D:DF:68:A6:F4:1A:A4:74

Alias name: comodoecccertificationauthority
Certificate fingerprints:
  MD5:  7C:62:FF:74:9D:31:53:5E:68:4A:D5:78:AA:1E:BF:23
  SHA1: 9F:74:4E:9F:2B:4D:BA:EC:0F:31:2C:50:B6:56:3B:8E:2D:93:C3:11
  SHA256:
 17:93:92:7A:06:14:54:97:89:AD:CE:2F:8F:34:F7:F0:B6:6D:0F:3A:E3:A3:B8:4D:21:EC:15:DB:BA:4F:AD:C7

Alias name: securetrustca
Certificate fingerprints:
  MD5:  DC:32:C3:A7:6D:25:57:C7:68:09:9D:EA:2D:A9:A2:D1
  SHA1: 87:82:C6:C3:04:35:3B:CF:D2:96:92:D2:59:3E:7D:44:D9:34:FF:11
  SHA256:
 F1:C1:B5:0A:E5:A2:0D:D8:03:0E:C9:F6:BC:24:82:3D:D3:67:B5:25:57:59:B4:E7:1B:61:FC:E9:F7:37:5D:73

Alias name: soneraclass1ca
Certificate fingerprints:
  MD5:  33:B7:84:F5:5F:27:D7:68:27:DE:14:DE:12:2A:ED:6F
  SHA1: 07:47:22:01:99:CE:74:B9:7C:B0:3D:79:B2:64:A2:C8:55:E9:33:FF
  SHA256:
 CD:80:82:84:CF:74:6F:F2:FD:6E:B5:8A:A1:D5:9C:4A:D4:B3:CA:56:FD:C6:27:4A:89:26:A7:83:5F:32:31:3D

Alias name: cadisigrootr2
Certificate fingerprints:
  MD5:  26:01:FB:D8:27:A7:17:9A:45:54:38:1A:43:01:3B:03
  SHA1: B5:61:EB:EA:A4:DE:E4:25:4B:69:1A:98:A5:57:47:C2:34:C7:D9:71
  SHA256:
 E2:3D:4A:03:6D:7B:70:E9:F5:95:B1:42:20:79:D2:B9:1E:DF:BB:1F:B6:51:A0:63:3E:AA:8A:9D:C5:F8:07:03
```

```
Alias name: cadisigrootr1
Certificate fingerprints:
  MD5:  BE:EC:11:93:9A:F5:69:21:BC:D7:C1:C0:67:89:CC:2A
  SHA1: 8E:1C:74:F8:A6:20:B9:E5:8A:F4:61:FA:EC:2B:47:56:51:1A:52:C6
  SHA256:
 F9:6F:23:F4:C3:E7:9C:07:7A:46:98:8D:5A:F5:90:06:76:A0:F0:39:CB:64:5D:D1:75:49:B2:16:C8:24:40:CE

Alias name: verisignclass3g5ca
Certificate fingerprints:
  MD5:  CB:17:E4:31:67:3E:E2:09:FE:45:57:93:F3:0A:FA:1C
  SHA1: 4E:B6:D5:78:49:9B:1C:CF:5F:58:1E:AD:56:BE:3D:9B:67:44:A5:E5
  SHA256:
 9A:CF:AB:7E:43:C8:D8:80:D0:6B:26:2A:94:DE:EE:E4:B4:65:99:89:C3:D0:CA:F1:9B:AF:64:05:E4:1A:B7:DF

Alias name: utnuserfirsthardwareca
Certificate fingerprints:
  MD5:  4C:56:41:E5:0D:BB:2B:E8:CA:A3:ED:18:08:AD:43:39
  SHA1: 04:83:ED:33:99:AC:36:08:05:87:22:ED:BC:5E:46:00:E3:BE:F9:D7
  SHA256:
 6E:A5:47:41:D0:04:66:7E:ED:1B:48:16:63:4A:A3:A7:9E:6E:4B:96:95:0F:82:79:DA:FC:8D:9B:D8:81:21:37

Alias name: addtrustqualifiedca
Certificate fingerprints:
  MD5:  27:EC:39:47:CD:DA:5A:AF:E2:9A:01:65:21:A9:4C:BB
  SHA1: 4D:23:78:EC:91:95:39:B5:00:7F:75:8F:03:3B:21:1E:C5:4D:8B:CF
  SHA256:
 80:95:21:08:05:DB:4B:BC:35:5E:44:28:D8:FD:6E:C2:CD:E3:AB:5F:B9:7A:99:42:98:8E:B8:F4:DC:D0:60:16

Alias name: verisignclass3g3ca
Certificate fingerprints:
  MD5:  CD:68:B6:A7:C7:C4:CE:75:E0:1D:4F:57:44:61:92:09
  SHA1: 13:2D:0D:45:53:4B:69:97:CD:B2:D5:C3:39:E2:55:76:60:9B:5C:C6
  SHA256:
 EB:04:CF:5E:B1:F3:9A:FA:76:2F:2B:B1:20:F2:96:CB:A5:20:C1:B9:7D:B1:58:95:65:B8:1C:B9:A1:7B:72:44

Alias name: thawtepersonalfreemailca
Certificate fingerprints:
  MD5:  53:4B:1D:17:58:58:1A:30:A1:90:F8:6E:5C:F2:CF:65
  SHA1: E6:18:83:AE:84:CA:C1:C1:CD:52:AD:E8:E9:25:2B:45:A6:4F:B7:E2
  SHA256:
 5B:38:BD:12:9E:83:D5:A0:CA:D2:39:21:08:94:90:D5:0D:4A:AE:37:04:28:F8:DD:FF:FF:FA:4C:15:64:E1:84

Alias name: certplusclass3pprimaryca
Certificate fingerprints:
  MD5:  E1:4B:52:73:D7:1B:DB:93:30:E5:BD:E4:09:6E:BE:FB
  SHA1: 21:6B:2A:29:E6:2A:00:CE:82:01:46:D8:24:41:41:B9:25:11:B2:79
  SHA256:
 CC:C8:94:89:37:1B:AD:11:1C:90:61:9B:EA:24:0A:2E:6D:AD:D9:9F:9F:6E:1D:4D:41:E5:8E:D6:DE:3D:02:85

Alias name: swisssigngoldg2ca
Certificate fingerprints:
  MD5:  24:77:D9:A8:91:D1:3B:FA:88:2D:C2:FF:F8:CD:33:93
  SHA1: D8:C5:38:8A:B7:30:1B:1B:6E:D4:7A:E6:45:25:3A:6F:9F:1A:27:61
  SHA256:
 62:DD:0B:E9:B9:F5:0A:16:3E:A0:F8:E7:5C:05:3B:1E:CA:57:EA:55:C8:68:8F:64:7C:68:81:F2:C8:35:7B:95

Alias name: swisssigngoldcag2
Certificate fingerprints:
  MD5:  24:77:D9:A8:91:D1:3B:FA:88:2D:C2:FF:F8:CD:33:93
  SHA1: D8:C5:38:8A:B7:30:1B:1B:6E:D4:7A:E6:45:25:3A:6F:9F:1A:27:61
  SHA256:
 62:DD:0B:E9:B9:F5:0A:16:3E:A0:F8:E7:5C:05:3B:1E:CA:57:EA:55:C8:68:8F:64:7C:68:81:F2:C8:35:7B:95

Alias name: dtrustrootclass3ca22009
Certificate fingerprints:
```

```
  MD5:  CD:E0:25:69:8D:47:AC:9C:89:35:90:F7:FD:51:3D:2F
  SHA1: 58:E8:AB:B0:36:15:33:FB:80:F7:9B:1B:6D:29:D3:FF:8D:5F:00:F0
  SHA256:
 49:E7:A4:42:AC:F0:EA:62:87:05:00:54:B5:25:64:B6:50:E4:F4:9E:42:E3:48:D6:AA:38:E0:39:E9:57:B1:C1

Alias name: acraizfnmtrcm
Certificate fingerprints:
  MD5:  E2:09:04:B4:D3:BD:D1:A0:14:FD:1A:D2:47:C4:57:1D
  SHA1: EC:50:35:07:B2:15:C4:95:62:19:E2:A8:9A:5B:42:99:2C:4C:2C:20
  SHA256:
 EB:C5:57:0C:29:01:8C:4D:67:B1:AA:12:7B:AF:12:F7:03:B4:61:1E:BC:17:B7:DA:B5:57:38:94:17:9B:93:FA

Alias name: securitycommunicationevrootca1
Certificate fingerprints:
  MD5:  22:2D:A6:01:EA:7C:0A:F7:F0:6C:56:43:3F:77:76:D3
  SHA1: FE:B8:C4:32:DC:F9:76:9A:CE:AE:3D:D8:90:8F:FD:28:86:65:64:7D
  SHA256:
 A2:2D:BA:68:1E:97:37:6E:2D:39:7D:72:8A:AE:3A:9B:62:96:B9:FD:BA:60:BC:2E:11:F6:47:F2:C6:75:FB:37

Alias name: starfieldclass2ca
Certificate fingerprints:
  MD5:  32:4A:4B:BB:C8:63:69:9B:BE:74:9A:C6:DD:1D:46:24
  SHA1: AD:7E:1C:28:B0:64:EF:8F:60:03:40:20:14:C3:D0:E3:37:0E:B5:8A
  SHA256:
 14:65:FA:20:53:97:B8:76:FA:A6:F0:A9:95:8E:55:90:E4:0F:CC:7F:AA:4F:B7:C2:C8:67:75:21:FB:5F:B6:58

Alias name: opentrustrootcag3
Certificate fingerprints:
  MD5:  21:37:B4:17:16:92:7B:67:46:70:A9:96:D7:A8:13:24
  SHA1: 6E:26:64:F3:56:BF:34:55:BF:D1:93:3F:7C:01:DE:D8:13:DA:8A:A6
  SHA256:
 B7:C3:62:31:70:6E:81:07:8C:36:7C:B8:96:19:8F:1E:32:08:DD:92:69:49:DD:8F:57:09:A4:10:F7:5B:62:92

Alias name: opentrustrootcag2
Certificate fingerprints:
  MD5:  57:24:B6:59:24:6B:AE:C8:FE:1C:0C:20:F2:C0:4E:EB
  SHA1: 79:5F:88:60:C5:AB:7C:3D:92:E6:CB:F4:8D:E1:45:CD:11:EF:60:0B
  SHA256:
 27:99:58:29:FE:6A:75:15:C1:BF:E8:48:F9:C4:76:1D:B1:6C:22:59:29:25:7B:F4:0D:08:94:F2:9E:A8:BA:F2

Alias name: buypassclass2rootca
Certificate fingerprints:
  MD5:  46:A7:D2:FE:45:FB:64:5A:A8:59:90:9B:78:44:9B:29
  SHA1: 49:0A:75:74:DE:87:0A:47:FE:58:EE:F6:C7:6B:EB:C6:0B:12:40:99
  SHA256:
 9A:11:40:25:19:7C:5B:B9:5D:94:E6:3D:55:CD:43:79:08:47:B6:46:B2:3C:DF:11:AD:A4:A0:0E:FF:15:FB:48

Alias name: opentrustrootcag1
Certificate fingerprints:
  MD5:  76:00:CC:81:29:CD:55:5E:88:6A:7A:2E:F7:4D:39:DA
  SHA1: 79:91:E8:34:F7:E2:EE:DD:08:95:01:52:E9:55:2D:14:E9:58:D5:7E
  SHA256:
 56:C7:71:28:D9:8C:18:D9:1B:4C:FD:FF:BC:25:EE:91:03:D4:75:8E:A2:AB:AD:82:6A:90:F3:45:7D:46:0E:B4

Alias name: globalsignr2ca
Certificate fingerprints:
  MD5:  94:14:77:7E:3E:5E:FD:8F:30:BD:41:B0:CF:E7:D0:30
  SHA1: 75:E0:AB:B6:13:85:12:27:1C:04:F8:5F:DD:DE:38:E4:B7:24:2E:FE
  SHA256:
 CA:42:DD:41:74:5F:D0:B8:1E:B9:02:36:2C:F9:D8:BF:71:9D:A1:BD:1B:1E:FC:94:6F:5B:4C:99:F4:2C:1B:9E

Alias name: buypassclass3rootca
Certificate fingerprints:
  MD5:  3D:3B:18:9E:2C:64:5A:E8:D5:88:CE:0E:F9:37:C2:EC
  SHA1: DA:FA:F7:FA:66:84:EC:06:8F:14:50:BD:C7:C2:81:A5:BC:A9:64:57
```

```
  SHA256:
 ED:F7:EB:BC:A2:7A:2A:38:4D:38:7B:7D:40:10:C6:66:E2:ED:B4:84:3E:4C:29:B4:AE:1D:5B:93:32:E6:B2:4D

Alias name: ecacc
Certificate fingerprints:
  MD5:  EB:F5:9D:29:0D:61:F9:42:1F:7C:C2:BA:6D:E3:15:09
  SHA1: 28:90:3A:63:5B:52:80:FA:E6:77:4C:0B:6D:A7:D6:BA:A6:4A:F2:E8
  SHA256:
 88:49:7F:01:60:2F:31:54:24:6A:E2:8C:4D:5A:EF:10:F1:D8:7E:BB:76:62:6F:4A:E0:B7:F9:5B:A7:96:87:99

Alias name: epkirootcertificationauthority
Certificate fingerprints:
  MD5:  1B:2E:00:CA:26:06:90:3D:AD:FE:6F:15:68:D3:6B:B3
  SHA1: 67:65:0D:F1:7E:8E:7E:5B:82:40:A4:F4:56:4B:CF:E2:3D:69:C6:F0
  SHA256:
 C0:A6:F4:DC:63:A2:4B:FD:CF:54:EF:2A:6A:08:2A:0A:72:DE:35:80:3E:2F:F5:FF:52:7A:E5:D8:72:06:DF:D5

Alias name: verisignclass1g2ca
Certificate fingerprints:
  MD5:  DB:23:3D:F9:69:FA:4B:B9:95:80:44:73:5E:7D:41:83
  SHA1: 27:3E:E1:24:57:FD:C4:F9:0C:55:E8:2B:56:16:7F:62:F5:32:E5:47
  SHA256:
 34:1D:E9:8B:13:92:AB:F7:F4:AB:90:A9:60:CF:25:D4:BD:6E:C6:5B:9A:51:CE:6E:D0:67:D0:0E:C7:CE:9B:7F

Alias name: certigna
Certificate fingerprints:
  MD5:  AB:57:A6:5B:7D:42:82:19:B5:D8:58:26:28:5E:FD:FF
  SHA1: B1:2E:13:63:45:86:A4:6F:1A:B2:60:68:37:58:2D:C4:AC:FD:94:97
  SHA256:
 E3:B6:A2:DB:2E:D7:CE:48:84:2F:7A:C5:32:41:C7:B7:1D:54:14:4B:FB:40:C1:1F:3F:1D:0B:42:F5:EE:A1:2D

Alias name: camerfirmaglobalchambersignroot
Certificate fingerprints:
  MD5:  C5:E6:7B:BF:06:D0:4F:43:ED:C4:7A:65:8A:FB:6B:19
  SHA1: 33:9B:6B:14:50:24:9B:55:7A:01:87:72:84:D9:E0:2F:C3:D2:D8:E9
  SHA256:
 EF:3C:B4:17:FC:8E:BF:6F:97:87:6C:9E:4E:CE:39:DE:1E:A5:FE:64:91:41:D1:02:8B:7D:11:C0:B2:29:8C:ED

Alias name: cfcaevroot
Certificate fingerprints:
  MD5:  74:E1:B6:ED:26:7A:7A:44:30:33:94:AB:7B:27:81:30
  SHA1: E2:B8:29:4B:55:84:AB:6B:58:C2:90:46:6C:AC:3F:B8:39:8F:84:83
  SHA256:
 5C:C3:D7:8E:4E:1D:5E:45:54:7A:04:E6:87:3E:64:F9:0C:F9:53:6D:1C:CC:2E:F8:00:F3:55:C4:C5:FD:70:FD

Alias name: soneraclass2rootca
Certificate fingerprints:
  MD5:  A3:EC:75:0F:2E:88:DF:FA:48:01:4E:0B:5C:48:6F:FB
  SHA1: 37:F7:6D:E6:07:7C:90:C5:B1:3E:93:1A:B7:41:10:B4:F2:E4:9A:27
  SHA256:
 79:08:B4:03:14:C1:38:10:0B:51:8D:07:35:80:7F:FB:FC:F8:51:8A:00:95:33:71:05:BA:38:6B:15:3D:D9:27

Alias name: certumtrustednetworkca
Certificate fingerprints:
  MD5:  D5:E9:81:40:C5:18:69:FC:46:2C:89:75:62:0F:AA:78
  SHA1: 07:E0:32:E0:20:B7:2C:3F:19:2F:06:28:A2:59:3A:19:A7:0F:06:9E
  SHA256:
 5C:58:46:8D:55:F5:8E:49:7E:74:39:82:D2:B5:00:10:B6:D1:65:37:4A:CF:83:A7:D4:A3:2D:B7:68:C4:40:8E

Alias name: securitycommunicationrootca2
Certificate fingerprints:
  MD5:  6C:39:7D:A4:0E:55:59:B2:3F:D6:41:B1:12:50:DE:43
  SHA1: 5F:3B:8C:F2:F8:10:B3:7D:78:B4:CE:EC:19:19:C3:73:34:B9:C7:74
  SHA256:
 51:3B:2C:EC:B8:10:D4:CD:E5:DD:85:39:1A:DF:C6:C2:DD:60:D8:7B:B7:36:D2:B5:21:48:4A:A4:7A:0E:BE:F6
```

```
Alias name: globalsigneccrootcar5
Certificate fingerprints:
  MD5:  9F:AD:3B:1C:02:1E:8A:BA:17:74:38:81:0C:A2:BC:08
  SHA1: 1F:24:C6:30:CD:A4:18:EF:20:69:FF:AD:4F:DD:5F:46:3A:1B:69:AA
  SHA256:
 17:9F:BC:14:8A:3D:D0:0F:D2:4E:A1:34:58:CC:43:BF:A7:F5:9C:81:82:D7:83:A5:13:F6:EB:EC:10:0C:89:24

Alias name: globalsigneccrootcar4
Certificate fingerprints:
  MD5:  20:F0:27:68:D1:7E:A0:9D:0E:E6:2A:CA:DF:5C:89:8E
  SHA1: 69:69:56:2E:40:80:F4:24:A1:E7:19:9F:14:BA:F3:EE:58:AB:6A:BB
  SHA256:
 BE:C9:49:11:C2:95:56:76:DB:6C:0A:55:09:86:D7:6E:3B:A0:05:66:7C:44:2C:97:62:B4:FB:B7:73:DE:22:8C

Alias name: chambersofcommerceroot2008
Certificate fingerprints:
  MD5:  5E:80:9E:84:5A:0E:65:0B:17:02:F3:55:18:2A:3E:D7
  SHA1: 78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C
  SHA256:
 06:3E:4A:FA:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46:17:D8:93:D7:FE:94:4E:10:A7:93:7E:E2:9D:96:93:C0

Alias name: pscprocert
Certificate fingerprints:
  MD5:  E6:24:E9:12:01:AE:0C:DE:8E:85:C4:CE:A3:12:DD:EC
  SHA1: 70:C1:8D:74:B4:28:81:0A:E4:FD:A5:75:D7:01:9F:99:B0:3D:50:74
  SHA256:
 3C:FC:3C:14:D1:F6:84:FF:17:E3:8C:43:CA:44:0C:00:B9:67:EC:93:3E:8B:FE:06:4C:A1:D7:2C:90:F2:AD:B0

Alias name: thawteprimaryrootcag3
Certificate fingerprints:
  MD5:  FB:1B:5D:43:8A:94:CD:44:C6:76:F2:43:4B:47:E7:31
  SHA1: F1:8B:53:8D:1B:E9:03:B6:A6:F0:56:43:5B:17:15:89:CA:F3:6B:F2
  SHA256:
 4B:03:F4:58:07:AD:70:F2:1B:FC:2C:AE:71:C9:FD:E4:60:4C:06:4C:F5:FF:B6:86:BA:E5:DB:AA:D7:FD:D3:4C

Alias name: quovadisrootca
Certificate fingerprints:
  MD5:  27:DE:36:FE:72:B7:00:03:00:9D:F4:F0:1E:6C:04:24
  SHA1: DE:3F:40:BD:50:93:D3:9B:6C:60:F6:DA:BC:07:62:01:00:89:76:C9
  SHA256:
 A4:5E:DE:3B:BB:F0:9C:8A:E1:5C:72:EF:C0:72:68:D6:93:A2:1C:99:6F:D5:1E:67:CA:07:94:60:FD:6D:88:73

Alias name: thawteprimaryrootcag2
Certificate fingerprints:
  MD5:  74:9D:EA:60:24:C4:FD:22:53:3E:CC:3A:72:D9:29:4F
  SHA1: AA:DB:BC:22:23:8F:C4:01:A1:27:BB:38:DD:F4:1D:DB:08:9E:F0:12
  SHA256:
 A4:31:0D:50:AF:18:A6:44:71:90:37:2A:86:AF:AF:8B:95:1F:FB:43:1D:83:7F:1E:56:88:B4:59:71:ED:15:57

Alias name: deprecateditsecca
Certificate fingerprints:
  MD5:  A5:96:0C:F6:B5:AB:27:E5:01:C6:00:88:9E:60:33:E5
  SHA1: 12:12:0B:03:0E:15:14:54:F4:DD:B3:F5:DE:13:6E:83:5A:29:72:9D
  SHA256:
 9A:59:DA:86:24:1A:FD:BA:A3:39:FA:9C:FD:21:6A:0B:06:69:4D:E3:7E:37:52:6B:BE:63:C8:BC:83:74:2E:CB

Alias name: usertrustrsacertificationauthority
Certificate fingerprints:
  MD5:  1B:FE:69:D1:91:B7:19:33:A3:72:A8:0F:E1:55:E5:B5
  SHA1: 2B:8F:1B:57:33:0D:BB:A2:D0:7A:6C:51:F7:0E:E9:0D:DA:B9:AD:8E
  SHA256:
 E7:93:C9:B0:2F:D8:AA:13:E2:1C:31:22:8A:CC:B0:81:19:64:3B:74:9C:89:89:64:B1:74:6D:46:C3:D4:CB:D2

Alias name: entrustrootcag2
Certificate fingerprints:
  MD5:  4B:E2:C9:91:96:65:0C:F4:0E:5A:93:92:A0:0A:FE:B2
```

```
  SHA1: 8C:F4:27:FD:79:0C:3A:D1:66:06:8D:E8:1E:57:EF:BB:93:22:72:D4
  SHA256:
 43:DF:57:74:B0:3E:7F:EF:5F:E4:0D:93:1A:7B:ED:F1:BB:2E:6B:42:73:8C:4E:6D:38:41:10:3D:3A:A7:F3:39

Alias name: networksolutionscertificateauthority
Certificate fingerprints:
  MD5:  D3:F3:A6:16:C0:FA:6B:1D:59:B1:2D:96:4D:0E:11:2E
  SHA1: 74:F8:A3:C3:EF:E7:B3:90:06:4B:83:90:3C:21:64:60:20:E5:DF:CE
  SHA256:
 15:F0:BA:00:A3:AC:7A:F3:AC:88:4C:07:2B:10:11:A0:77:BD:77:C0:97:F4:01:64:B2:F8:59:8A:BD:83:86:0C

Alias name: trustcenterclass4caii
Certificate fingerprints:
  MD5:  9D:FB:F9:AC:ED:89:33:22:F4:28:48:83:25:23:5B:E0
  SHA1: A6:9A:91:FD:05:7F:13:6A:42:63:0B:B1:76:0D:2D:51:12:0C:16:50
  SHA256:
 32:66:96:7E:59:CD:68:00:8D:9D:D3:20:81:11:85:C7:04:20:5E:8D:95:FD:D8:4F:1C:7B:31:1E:67:04:FC:32

Alias name: oistewisekeyglobalrootgaca
Certificate fingerprints:
  MD5:  BC:6C:51:33:A7:E9:D3:66:63:54:15:72:1B:21:92:93
  SHA1: 59:22:A1:E1:5A:EA:16:35:21:F8:98:39:6A:46:46:B0:44:1B:0F:A9
  SHA256:
 41:C9:23:86:6A:B4:CA:D6:B7:AD:57:80:81:58:2E:02:07:97:A6:CB:DF:4F:FF:78:CE:83:96:B3:89:37:D7:F5

Alias name: verisignuniversalrootcertificationauthority
Certificate fingerprints:
  MD5:  8E:AD:B5:01:AA:4D:81:E4:8C:1D:D1:E1:14:00:95:19
  SHA1: 36:79:CA:35:66:87:72:30:4D:30:A5:FB:87:3B:0F:A7:7B:B7:0D:54
  SHA256:
 23:99:56:11:27:A5:71:25:DE:8C:EF:EA:61:0D:DF:2F:A0:78:B5:C8:06:7F:4E:82:82:90:BF:B8:60:E8:4B:3C

Alias name: ttelesecglobalrootclass3ca
Certificate fingerprints:
  MD5:  CA:FB:40:A8:4E:39:92:8A:1D:FE:8E:2F:C4:27:EA:EF
  SHA1: 55:A6:72:3E:CB:F2:EC:CD:C3:23:74:70:19:9D:2A:BE:11:E3:81:D1
  SHA256:
 FD:73:DA:D3:1C:64:4F:F1:B4:3B:EF:0C:CD:DA:96:71:0B:9C:D9:87:5E:CA:7E:31:70:7A:F3:E9:6D:52:2B:BD

Alias name: starfieldservicesrootg2ca
Certificate fingerprints:
  MD5:  17:35:74:AF:7B:61:1C:EB:F4:F9:3C:E2:EE:40:F9:A2
  SHA1: 92:5A:8F:8D:2C:6D:04:E0:66:5F:59:6A:FF:22:D8:63:E8:25:6F:3F
  SHA256:
 56:8D:69:05:A2:C8:87:08:A4:B3:02:51:90:ED:CF:ED:B1:97:4A:60:6A:13:C6:E5:29:0F:CB:2A:E6:3E:DA:B5

Alias name: addtrustexternalroot
Certificate fingerprints:
  MD5:  1D:35:54:04:85:78:B0:3F:42:42:4D:BF:20:73:0A:3F
  SHA1: 02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4:5B:68:85:18:68
  SHA256:
 68:7F:A4:51:38:22:78:FF:F0:C8:B1:1F:8D:43:D5:76:67:1C:6E:B2:BC:EA:B4:13:FB:83:D9:65:D0:6D:2F:F2

Alias name: turktrustelektroniksertifikahizmetsaglayicisih5
Certificate fingerprints:
  MD5:  DA:70:8E:F0:22:DF:93:26:F6:5F:9F:D3:15:06:52:4E
  SHA1: C4:18:F6:4D:46:D1:DF:00:3D:27:30:13:72:43:A9:12:11:C6:75:FB
  SHA256:
 49:35:1B:90:34:44:C1:85:CC:DC:5C:69:3D:24:D8:55:5C:B2:08:D6:A8:14:13:07:69:9F:4A:F0:63:19:9D:78

Alias name: camerfirmachambersca
Certificate fingerprints:
  MD5:  5E:80:9E:84:5A:0E:65:0B:17:02:F3:55:18:2A:3E:D7
  SHA1: 78:6A:74:AC:76:AB:14:7F:9C:6A:30:50:BA:9E:A8:7E:FE:9A:CE:3C
  SHA256:
 06:3E:4A:FA:C4:91:DF:D3:32:F3:08:9B:85:42:E9:46:17:D8:93:D7:FE:94:4E:10:A7:93:7E:E2:9D:96:93:C0
```

```
Alias name: certsignrootca
Certificate fingerprints:
  MD5:  18:98:C0:D6:E9:3A:FC:F9:B0:F5:0C:F7:4B:01:44:17
  SHA1: FA:B7:EE:36:97:26:62:FB:2D:B0:2A:F6:BF:03:FD:E8:7C:4B:2F:9B
  SHA256:
 EA:A9:62:C4:FA:4A:6B:AF:EB:E4:15:19:6D:35:1C:CD:88:8D:4F:53:F3:FA:8A:E6:D7:C4:66:A9:4E:60:42:BB

Alias name: verisignuniversalrootca
Certificate fingerprints:
  MD5:  8E:AD:B5:01:AA:4D:81:E4:8C:1D:D1:E1:14:00:95:19
  SHA1: 36:79:CA:35:66:87:72:30:4D:30:A5:FB:87:3B:0F:A7:7B:B7:0D:54
  SHA256:
 23:99:56:11:27:A5:71:25:DE:8C:EF:EA:61:0D:DF:2F:A0:78:B5:C8:06:7F:4E:82:82:90:BF:B8:60:E8:4B:3C

Alias name: geotrustuniversalca
Certificate fingerprints:
  MD5:  92:65:58:8B:A2:1A:31:72:73:68:5C:B4:A5:7A:07:48
  SHA1: E6:21:F3:35:43:79:05:9A:4B:68:30:9D:8A:2F:74:22:15:87:EC:79
  SHA256:
 A0:45:9B:9F:63:B2:25:59:F5:FA:5D:4C:6D:B3:F9:F7:2F:F1:93:42:03:35:78:F0:73:BF:1D:1B:46:CB:B9:12

Alias name: luxtrustglobalroot2
Certificate fingerprints:
  MD5:  B2:E1:09:00:61:AF:F7:F1:91:6F:C4:AD:8D:5E:3B:7C
  SHA1: 1E:0E:56:19:0A:D1:8B:25:98:B2:04:44:FF:66:8A:04:17:99:5F:3F
  SHA256:
 54:45:5F:71:29:C2:0B:14:47:C4:18:F9:97:16:8F:24:C5:8F:C5:02:3B:F5:DA:5B:E2:EB:6E:1D:D8:90:2E:D5

Alias name: twcaglobalrootca
Certificate fingerprints:
  MD5:  F9:03:7E:CF:E6:9E:3C:73:7A:2A:90:07:69:FF:2B:96
  SHA1: 9C:BB:48:53:F6:A4:F6:D3:52:A4:E8:32:52:55:60:13:F5:AD:AF:65
  SHA256:
 59:76:90:07:F7:68:5D:0F:CD:50:87:2F:9F:95:D5:75:5A:5B:2B:45:7D:81:F3:69:2B:61:0A:98:67:2F:0E:1B

Alias name: tubitakkamusmsslkoksertifikasisurum1
Certificate fingerprints:
  MD5:  DC:00:81:DC:69:2F:3E:2F:B0:3B:F6:3D:5A:91:8E:49
  SHA1: 31:43:64:9B:EC:CE:27:EC:ED:3A:3F:0B:8F:0D:E4:E8:91:DD:EE:CA
  SHA256:
 46:ED:C3:68:90:46:D5:3A:45:3F:B3:10:4A:B8:0D:CA:EC:65:8B:26:60:EA:16:29:DD:7E:86:79:90:64:87:16

Alias name: affirmtrustnetworkingca
Certificate fingerprints:
  MD5:  42:65:CA:BE:01:9A:9A:4C:A9:8C:41:49:CD:C0:D5:7F
  SHA1: 29:36:21:02:8B:20:ED:02:F5:66:C5:32:D1:D6:ED:90:9F:45:00:2F
  SHA256:
 0A:81:EC:5A:92:97:77:F1:45:90:4A:F3:8D:5D:50:9F:66:B5:E2:C5:8F:CD:B5:31:05:8B:0E:17:F3:F0:B4:1B

Alias name: affirmtrustcommercialca
Certificate fingerprints:
  MD5:  82:92:BA:5B:EF:CD:8A:6F:A6:3D:55:F9:84:F6:D6:B7
  SHA1: F9:B5:B6:32:45:5F:9C:BE:EC:57:5F:80:DC:E9:6E:2C:C7:B2:78:B7
  SHA256:
 03:76:AB:1D:54:C5:F9:80:3C:E4:B2:E2:01:A0:EE:7E:EF:7B:57:B6:36:E8:A9:3C:9B:8D:48:60:C9:6F:5F:A7

Alias name: godaddyrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  80:3A:BC:22:C1:E6:FB:8D:9B:3B:27:4A:32:1B:9A:01
  SHA1: 47:BE:AB:C9:22:EA:E8:0E:78:78:34:62:A7:9F:45:C2:54:FD:E6:8B
  SHA256:
 45:14:0B:32:47:EB:9C:C8:C5:B4:F0:D7:B5:30:91:F7:32:92:08:9E:6E:5A:63:E2:74:9D:D3:AC:A9:19:8E:DA

Alias name: starfieldrootg2ca
Certificate fingerprints:
```

```
  MD5:  D6:39:81:C6:52:7E:96:69:FC:FC:CA:66:ED:05:F2:96
  SHA1: B5:1C:06:7C:EE:2B:0C:3D:F8:55:AB:2D:92:F4:FE:39:D4:E7:0F:0E
  SHA256:
 2C:E1:CB:0B:F9:D2:F9:E1:02:99:3F:BE:21:51:52:C3:B2:DD:0C:AB:DE:1C:68:E5:31:9B:83:91:54:DB:B7:F5

Alias name: dtrustrootclass3ca2ev2009
Certificate fingerprints:
  MD5:  AA:C6:43:2C:5E:2D:CD:C4:34:C0:50:4F:11:02:4F:B6
  SHA1: 96:C9:1B:0B:95:B4:10:98:42:FA:D0:D8:22:79:FE:60:FA:B9:16:83
  SHA256:
 EE:C5:49:6B:98:8C:E9:86:25:B9:34:09:2E:EC:29:08:BE:D0:B0:F3:16:C2:D4:73:0C:84:EA:F1:F3:D3:48:81

Alias name: buypassclass3ca
Certificate fingerprints:
  MD5:  3D:3B:18:9E:2C:64:5A:E8:D5:88:CE:0E:F9:37:C2:EC
  SHA1: DA:FA:F7:FA:66:84:EC:06:8F:14:50:BD:C7:C2:81:A5:BC:A9:64:57
  SHA256:
 ED:F7:EB:BC:A2:7A:2A:38:4D:38:7B:7D:40:10:C6:66:E2:ED:B4:84:3E:4C:29:B4:AE:1D:5B:93:32:E6:B2:4D

Alias name: verisignclass2g3ca
Certificate fingerprints:
  MD5:  F8:BE:C4:63:22:C9:A8:46:74:8B:B8:1D:1E:4A:2B:F6
  SHA1: 61:EF:43:D7:7F:CA:D4:61:51:BC:98:E0:C3:59:12:AF:9F:EB:63:11
  SHA256:
 92:A9:D9:83:3F:E1:94:4D:B3:66:E8:BF:AE:7A:95:B6:48:0C:2D:6C:6C:2A:1B:E6:5D:42:36:B6:08:FC:A1:BB

Alias name: digicerttrustedrootg4
Certificate fingerprints:
  MD5:  78:F2:FC:AA:60:1F:2F:B4:EB:C9:37:BA:53:2E:75:49
  SHA1: DD:FB:16:CD:49:31:C9:73:A2:03:7D:3F:C8:3A:4D:7D:77:5D:05:E4
  SHA256:
 55:2F:7B:DC:F1:A7:AF:9E:6C:E6:72:01:7F:4F:12:AB:F7:72:40:C7:8E:76:1A:C2:03:D1:D9:D2:0A:C8:99:88

Alias name: quovadisrootca2g3
Certificate fingerprints:
  MD5:  AF:0C:86:6E:BF:40:2D:7F:0B:3E:12:50:BA:12:3D:06
  SHA1: 09:3C:61:F3:8B:8B:DC:7D:55:DF:75:38:02:05:00:E1:25:F5:C8:36
  SHA256:
 8F:E4:FB:0A:F9:3A:4D:0D:67:DB:0B:EB:B2:3E:37:C7:1B:F3:25:DC:BC:DD:24:0E:A0:4D:AF:58:B4:7E:18:40

Alias name: geotrustprimarycertificationauthorityg3
Certificate fingerprints:
  MD5:  B5:E8:34:36:C9:10:44:58:48:70:6D:2E:83:D4:B8:05
  SHA1: 03:9E:ED:B8:0B:E7:A0:3C:69:53:89:3B:20:D2:D9:32:3A:4C:2A:FD
  SHA256:
 B4:78:B8:12:25:0D:F8:78:63:5C:2A:A7:EC:7D:15:5E:AA:62:5E:E8:29:16:E2:CD:29:43:61:88:6C:D1:FB:D4

Alias name: geotrustprimarycertificationauthorityg2
Certificate fingerprints:
  MD5:  01:5E:D8:6B:BD:6F:3D:8E:A1:31:F8:12:E0:98:73:6A
  SHA1: 8D:17:84:D5:37:F3:03:7D:EC:70:FE:57:8B:51:9A:99:E6:10:D7:B0
  SHA256:
 5E:DB:7A:C4:3B:82:A0:6A:87:61:E8:D7:BE:49:79:EB:F2:61:1F:7D:D7:9B:F9:1C:1C:6B:56:6A:21:9E:D7:66

Alias name: godaddyclass2ca
Certificate fingerprints:
  MD5:  91:DE:06:25:AB:DA:FD:32:17:0C:BB:25:17:2A:84:67
  SHA1: 27:96:BA:E6:3F:18:01:E2:77:26:1B:A0:D7:77:70:02:8F:20:EE:E4
  SHA256:
 C3:84:6B:F2:4B:9E:93:CA:64:27:4C:0E:C6:7C:1E:CC:5E:02:4F:FC:AC:D2:D7:40:19:35:0E:81:FE:54:6A:E4

Alias name: trustcoreca1
Certificate fingerprints:
  MD5:  27:92:23:1D:0A:F5:40:7C:E9:E6:6B:9D:D8:F5:E7:6C
  SHA1: 58:D1:DF:95:95:67:6B:63:C0:F0:5B:1C:17:4D:8B:84:0B:C8:78:BD
```

```
  SHA256:
 5A:88:5D:B1:9C:01:D9:12:C5:75:93:88:93:8C:AF:BB:DF:03:1A:B2:D4:8E:91:EE:15:58:9B:42:97:1D:03:9C

Alias name: hellenicacademicandresearchinstitutionseccrootca2015
Certificate fingerprints:
  MD5:  81:E5:B4:17:EB:C2:F5:E1:4B:0D:41:7B:49:92:FE:EF
  SHA1: 9F:F1:71:8D:92:D5:9A:F3:7D:74:97:B4:BC:6F:84:68:0B:BA:B6:66
  SHA256:
 44:B5:45:AA:8A:25:E6:5A:73:CA:15:DC:27:FC:36:D2:4C:1C:B9:95:3A:06:65:39:B1:15:82:DC:48:7B:48:33

Alias name: utnuserfirstobjectca
Certificate fingerprints:
  MD5:  A7:F2:E4:16:06:41:11:50:30:6B:9C:E3:B4:9C:B0:C9
  SHA1: E1:2D:FB:4B:41:D7:D9:C3:2B:30:51:4B:AC:1D:81:D8:38:5E:2D:46
  SHA256:
 6F:FF:78:E4:00:A7:0C:11:01:1C:D8:59:77:C4:59:FB:5A:F9:6A:3D:F0:54:08:20:D0:F4:B8:60:78:75:E5:8F

Alias name: ttelesecglobalrootclass3
Certificate fingerprints:
  MD5:  CA:FB:40:A8:4E:39:92:8A:1D:FE:8E:2F:C4:27:EA:EF
  SHA1: 55:A6:72:3E:CB:F2:EC:CD:C3:23:74:70:19:9D:2A:BE:11:E3:81:D1
  SHA256:
 FD:73:DA:D3:1C:64:4F:F1:B4:3B:EF:0C:CD:DA:96:71:0B:9C:D9:87:5E:CA:7E:31:70:7A:F3:E9:6D:52:2B:BD

Alias name: ttelesecglobalrootclass2
Certificate fingerprints:
  MD5:  2B:9B:9E:E4:7B:6C:1F:00:72:1A:CC:C1:77:79:DF:6A
  SHA1: 59:0D:2D:7D:88:4F:40:2E:61:7E:A5:62:32:17:65:CF:17:D8:94:E9
  SHA256:
 91:E2:F5:78:8D:58:10:EB:A7:BA:58:73:7D:E1:54:8A:8E:CA:CD:01:45:98:BC:0B:14:3E:04:1B:17:05:25:52

Alias name: addtrustclass1ca
Certificate fingerprints:
  MD5:  1E:42:95:02:33:92:6B:B9:5F:C0:7F:DA:D6:B2:4B:FC
  SHA1: CC:AB:0E:A0:4C:23:01:D6:69:7B:DD:37:9F:CD:12:EB:24:E3:94:9D
  SHA256:
 8C:72:09:27:9A:C0:4E:27:5E:16:D0:7F:D3:B7:75:E8:01:54:B5:96:80:46:E3:1F:52:DD:25:76:63:24:E9:A7

Alias name: amzninternalrootca
Certificate fingerprints:
  MD5:  08:09:73:AC:E0:78:41:7C:0A:26:33:51:E8:CF:E6:60
  SHA1: A7:B7:F6:15:8A:FF:1E:C8:85:13:38:BC:93:EB:A2:AB:A4:09:EF:06
  SHA256:
 0E:DE:63:C1:DC:7A:8E:11:F1:AB:BC:05:4F:59:EE:49:9D:62:9A:2F:DE:9C:A7:16:32:A2:64:29:3E:8B:66:AA

Alias name: starfieldrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  D6:39:81:C6:52:7E:96:69:FC:FC:CA:66:ED:05:F2:96
  SHA1: B5:1C:06:7C:EE:2B:0C:3D:F8:55:AB:2D:92:F4:FE:39:D4:E7:0F:0E
  SHA256:
 2C:E1:CB:0B:F9:D2:F9:E1:02:99:3F:BE:21:51:52:C3:B2:DD:0C:AB:DE:1C:68:E5:31:9B:83:91:54:DB:B7:F5

Alias name: camerfirmachambersignca
Certificate fingerprints:
  MD5:  9E:80:FF:78:01:0C:2E:C1:36:BD:FE:96:90:6E:08:F3
  SHA1: 4A:BD:EE:EC:95:0D:35:9C:89:AE:C7:52:A1:2C:5B:29:F6:D6:AA:0C
  SHA256:
 13:63:35:43:93:34:A7:69:80:16:A0:D3:24:DE:72:28:4E:07:9D:7B:52:20:BB:8F:BD:74:78:16:EE:BE:BA:CA

Alias name: secomscrootca2
Certificate fingerprints:
  MD5:  6C:39:7D:A4:0E:55:59:B2:3F:D6:41:B1:12:50:DE:43
  SHA1: 5F:3B:8C:F2:F8:10:B3:7D:78:B4:CE:EC:19:19:C3:73:34:B9:C7:74
  SHA256:
 51:3B:2C:EC:B8:10:D4:CD:E5:DD:85:39:1A:DF:C6:C2:DD:60:D8:7B:B7:36:D2:B5:21:48:4A:A4:7A:0E:BE:F6
```

```
Alias name: entrustevca
Certificate fingerprints:
  MD5:  D6:A5:C3:ED:5D:DD:3E:00:C1:3D:87:92:1F:1D:3F:E4
  SHA1: B3:1E:B1:B7:40:E3:6C:84:02:DA:DC:37:D4:4D:F5:D4:67:49:52:F9
  SHA256:
 73:C1:76:43:4F:1B:C6:D5:AD:F4:5B:0E:76:E7:27:28:7C:8D:E5:76:16:C1:E6:E6:14:1A:2B:2C:BC:7D:8E:4C

Alias name: secomscrootca1
Certificate fingerprints:
  MD5:  F1:BC:63:6A:54:E0:B5:27:F5:CD:E7:1A:E3:4D:6E:4A
  SHA1: 36:B1:2B:49:F9:81:9E:D7:4C:9E:BC:38:0F:C6:56:8F:5D:AC:B2:F7
  SHA256:
 E7:5E:72:ED:9F:56:0E:EC:6E:B4:80:00:73:A4:3F:C3:AD:19:19:5A:39:22:82:01:78:95:97:4A:99:02:6B:6C

Alias name: affirmtrustcommercial
Certificate fingerprints:
  MD5:  82:92:BA:5B:EF:CD:8A:6F:A6:3D:55:F9:84:F6:D6:B7
  SHA1: F9:B5:B6:32:45:5F:9C:BE:EC:57:5F:80:DC:E9:6E:2C:C7:B2:78:B7
  SHA256:
 03:76:AB:1D:54:C5:F9:80:3C:E4:B2:E2:01:A0:EE:7E:EF:7B:57:B6:36:E8:A9:3C:9B:8D:48:60:C9:6F:5F:A7

Alias name: digicertassuredidrootg3
Certificate fingerprints:
  MD5:  7C:7F:65:31:0C:81:DF:8D:BA:3E:99:E2:5C:AD:6E:FB
  SHA1: F5:17:A2:4F:9A:48:C6:C9:F8:A2:00:26:9F:DC:0F:48:2C:AB:30:89
  SHA256:
 7E:37:CB:8B:4C:47:09:0C:AB:36:55:1B:A6:F4:5D:B8:40:68:0F:BA:16:6A:95:2D:B1:00:71:7F:43:05:3F:C2

Alias name: affirmtrustnetworking
Certificate fingerprints:
  MD5:  42:65:CA:BE:01:9A:9A:4C:A9:8C:41:49:CD:C0:D5:7F
  SHA1: 29:36:21:02:8B:20:ED:02:F5:66:C5:32:D1:D6:ED:90:9F:45:00:2F
  SHA256:
 0A:81:EC:5A:92:97:77:F1:45:90:4A:F3:8D:5D:50:9F:66:B5:E2:C5:8F:CD:B5:31:05:8B:0E:17:F3:F0:B4:1B

Alias name: izenpecom
Certificate fingerprints:
  MD5:  A6:B0:CD:85:80:DA:5C:50:34:A3:39:90:2F:55:67:73
  SHA1: 2F:78:3D:25:52:18:A7:4A:65:39:71:B5:2C:A2:9C:45:15:6F:E9:19
  SHA256:
 25:30:CC:8E:98:32:15:02:BA:D9:6F:9B:1F:BA:1B:09:9E:2D:29:9E:0F:45:48:BB:91:4F:36:3B:C0:D4:53:1F

Alias name: amazon-ca-g4-legacy
Certificate fingerprints:
  MD5:  6C:E5:BD:67:A4:4F:E3:FD:C2:4C:46:E6:06:5B:6D:55
  SHA1: EA:E7:DE:F9:0A:BE:9F:0B:68:CE:B7:24:0D:80:74:03:BF:6E:B1:6E
  SHA256:
 CD:72:C4:7F:B4:AD:28:A4:67:2B:E1:86:47:D4:40:E9:3B:16:2D:95:DB:3C:2F:94:BB:81:D9:09:F7:91:24:5E

Alias name: digicertassuredidrootg2
Certificate fingerprints:
  MD5:  92:38:B9:F8:63:24:82:65:2C:57:33:E6:FE:81:8F:9D
  SHA1: A1:4B:48:D9:43:EE:0A:0E:40:90:4F:3C:E0:A4:C0:91:93:51:5D:3F
  SHA256:
 7D:05:EB:B6:82:33:9F:8C:94:51:EE:09:4E:EB:FE:FA:79:53:A1:14:ED:B2:F4:49:49:45:2F:AB:7D:2F:C1:85

Alias name: comodoaaaservicesroot
Certificate fingerprints:
  MD5:  49:79:04:B0:EB:87:19:AC:47:B0:BC:11:51:9B:74:D0
  SHA1: D1:EB:23:A4:6D:17:D6:8F:D9:25:64:C2:F1:F1:60:17:64:D8:E3:49
  SHA256:
 D7:A7:A0:FB:5D:7E:27:31:D7:71:E9:48:4E:BC:DE:F7:1D:5F:0C:3E:0A:29:48:78:2B:C8:3E:E0:EA:69:9E:F4

Alias name: entrustnetpremium2048secureserverca
Certificate fingerprints:
  MD5:  EE:29:31:BC:32:7E:9A:E6:E8:B5:F7:51:B4:34:71:90
```

```
  SHA1: 50:30:06:09:1D:97:D4:F5:AE:39:F7:CB:E7:92:7D:7D:65:2D:34:31
  SHA256:
 6D:C4:71:72:E0:1C:BC:B0:BF:62:58:0D:89:5F:E2:B8:AC:9A:D4:F8:73:80:1E:0C:10:B9:C8:37:D2:1E:B1:77

Alias name: trustcorrootcertca2
Certificate fingerprints:
  MD5:  A2:E1:F8:18:0B:BA:45:D5:C7:41:2A:BB:37:52:45:64
  SHA1: B8:BE:6D:CB:56:F1:55:B9:63:D4:12:CA:4E:06:34:C7:94:B2:1C:C0
  SHA256:
 07:53:E9:40:37:8C:1B:D5:E3:83:6E:39:5D:AE:A5:CB:83:9E:50:46:F1:BD:0E:AE:19:51:CF:10:FE:C7:C9:65

Alias name: entrust2048ca
Certificate fingerprints:
  MD5:  EE:29:31:BC:32:7E:9A:E6:E8:B5:F7:51:B4:34:71:90
  SHA1: 50:30:06:09:1D:97:D4:F5:AE:39:F7:CB:E7:92:7D:7D:65:2D:34:31
  SHA256:
 6D:C4:71:72:E0:1C:BC:B0:BF:62:58:0D:89:5F:E2:B8:AC:9A:D4:F8:73:80:1E:0C:10:B9:C8:37:D2:1E:B1:77

Alias name: trustcorrootcertca1
Certificate fingerprints:
  MD5:  6E:85:F1:DC:1A:00:D3:22:D5:B2:B2:AC:6B:37:05:45
  SHA1: FF:BD:CD:E7:82:C8:43:5E:3C:6F:26:86:5C:CA:A8:3A:45:5B:C3:0A
  SHA256:
 D4:0E:9C:86:CD:8F:E4:68:C1:77:69:59:F4:9E:A7:74:FA:54:86:84:B6:C4:06:F3:90:92:61:F4:DC:E2:57:5C

Alias name: baltimorecybertrustroot
Certificate fingerprints:
  MD5:  AC:B6:94:A5:9C:17:E0:D7:91:52:9B:B1:97:06:A6:E4
  SHA1: D4:DE:20:D0:5E:66:FC:53:FE:1A:50:88:2C:78:DB:28:52:CA:E4:74
  SHA256:
 16:AF:57:A9:F6:76:B0:AB:12:60:95:AA:5E:BA:DE:F2:2A:B3:11:19:D6:44:AC:95:CD:4B:93:DB:F3:F2:6A:EB

Alias name: eecertificationcentrerootca
Certificate fingerprints:
  MD5:  43:5E:88:D4:7D:1A:4A:7E:FD:84:2E:52:EB:01:D4:6F
  SHA1: C9:A8:B9:E7:55:80:5E:58:E3:53:77:A7:25:EB:AF:C3:7B:27:CC:D7
  SHA256:
 3E:84:BA:43:42:90:85:16:E7:75:73:C0:99:2F:09:79:CA:08:4E:46:85:68:1F:F1:95:CC:BA:8A:22:9B:8A:76

Alias name: dstacescax6
Certificate fingerprints:
  MD5:  21:D8:4C:82:2B:99:09:33:A2:EB:14:24:8D:8E:5F:E8
  SHA1: 40:54:DA:6F:1C:3F:40:74:AC:ED:0F:EC:CD:DB:79:D1:53:FB:90:1D
  SHA256:
 76:7C:95:5A:76:41:2C:89:AF:68:8E:90:A1:C7:0F:55:6C:FD:6B:60:25:DB:EA:10:41:6D:7E:B6:83:1F:8C:40

Alias name: comodocertificationauthority
Certificate fingerprints:
  MD5:  5C:48:DC:F7:42:72:EC:56:94:6D:1C:CC:71:35:80:75
  SHA1: 66:31:BF:9E:F7:4F:9E:B6:C9:D5:A6:0C:BA:6A:BE:D1:F7:BD:EF:7B
  SHA256:
 0C:2C:D6:3D:F7:80:6F:A3:99:ED:E8:09:11:6B:57:5B:F8:79:89:F0:65:18:F9:80:8C:86:05:03:17:8B:AF:66

Alias name: thawteserverca
Certificate fingerprints:
  MD5:  EE:FE:61:69:65:6E:F8:9C:C6:2A:F4:D7:2B:63:EF:A2
  SHA1: 9F:AD:91:A6:CE:6A:C6:C5:00:47:C4:4E:C9:D4:A5:0D:92:D8:49:79
  SHA256:
 87:C6:78:BF:B8:B2:5F:38:F7:E9:7B:33:69:56:BB:CF:14:4B:BA:CA:A5:36:47:E6:1A:23:25:BC:10:55:31:6B

Alias name: secomvalicertclass1ca
Certificate fingerprints:
  MD5:  65:58:AB:15:AD:57:6C:1E:A8:A7:B5:69:AC:BF:FF:EB
  SHA1: E5:DF:74:3C:B6:01:C4:9B:98:43:DC:AB:8C:E8:6A:81:10:9F:E4:8E
  SHA256:
 F4:C1:49:55:1A:30:13:A3:5B:C7:BF:FE:17:A7:F3:44:9B:C1:AB:5B:5A:0A:E7:4B:06:C2:3B:90:00:4C:01:04
```

```
Alias name: godaddyrootg2ca
Certificate fingerprints:
  MD5:  80:3A:BC:22:C1:E6:FB:8D:9B:3B:27:4A:32:1B:9A:01
  SHA1: 47:BE:AB:C9:22:EA:E8:0E:78:78:34:62:A7:9F:45:C2:54:FD:E6:8B
  SHA256:
 45:14:0B:32:47:EB:9C:C8:C5:B4:F0:D7:B5:30:91:F7:32:92:08:9E:6E:5A:63:E2:74:9D:D3:AC:A9:19:8E:DA

Alias name: globalchambersignroot2008
Certificate fingerprints:
  MD5:  9E:80:FF:78:01:0C:2E:C1:36:BD:FE:96:90:6E:08:F3
  SHA1: 4A:BD:EE:EC:95:0D:35:9C:89:AE:C7:52:A1:2C:5B:29:F6:D6:AA:0C
  SHA256:
 13:63:35:43:93:34:A7:69:80:16:A0:D3:24:DE:72:28:4E:07:9D:7B:52:20:BB:8F:BD:74:78:16:EE:BE:BA:CA

Alias name: equifaxsecureebusinessca1
Certificate fingerprints:
  MD5:  14:C0:08:E5:A3:85:03:A3:BE:78:E9:67:4F:27:CA:EE
  SHA1: AE:E6:3D:70:E3:76:FB:C7:3A:EB:B0:A1:C1:D4:C4:7A:A7:40:B3:F4
  SHA256:
 2E:3A:2B:B5:11:25:05:83:6C:A8:96:8B:E2:CB:37:27:CE:9B:56:84:5C:6E:E9:8E:91:85:10:4A:FB:9A:F5:96

Alias name: quovadisrootca3
Certificate fingerprints:
  MD5:  31:85:3C:62:94:97:63:B9:AA:FD:89:4E:AF:6F:E0:CF
  SHA1: 1F:49:14:F7:D8:74:95:1D:DD:AE:02:C0:BE:FD:3A:2D:82:75:51:85
  SHA256:
 18:F1:FC:7F:20:5D:F8:AD:DD:EB:7F:E0:07:DD:57:E3:AF:37:5A:9C:4D:8D:73:54:6B:F4:F1:FE:D1:E1:8D:35

Alias name: usertrustecccertificationauthority
Certificate fingerprints:
  MD5:  FA:68:BC:D9:B5:7F:AD:FD:C9:1D:06:83:28:CC:24:C1
  SHA1: D1:CB:CA:5D:B2:D5:2A:7F:69:3B:67:4D:E5:F0:5A:1D:0C:95:7D:F0
  SHA256:
 4F:F4:60:D5:4B:9C:86:DA:BF:BC:FC:57:12:E0:40:0D:2B:ED:3F:BC:4D:4F:BD:AA:86:E0:6A:DC:D2:A9:AD:7A

Alias name: quovadisrootca2
Certificate fingerprints:
  MD5:  5E:39:7B:DD:F8:BA:EC:82:E9:AC:62:BA:0C:54:00:2B
  SHA1: CA:3A:FB:CF:12:40:36:4B:44:B2:16:20:88:80:48:39:19:93:7C:F7
  SHA256:
 85:A0:DD:7D:D7:20:AD:B7:FF:05:F8:3D:54:2B:20:9D:C7:FF:45:28:F7:D6:77:B1:83:89:FE:A5:E5:C4:9E:86

Alias name: soneraclass2ca
Certificate fingerprints:
  MD5:  A3:EC:75:0F:2E:88:DF:FA:48:01:4E:0B:5C:48:6F:FB
  SHA1: 37:F7:6D:E6:07:7C:90:C5:B1:3E:93:1A:B7:41:10:B4:F2:E4:9A:27
  SHA256:
 79:08:B4:03:14:C1:38:10:0B:51:8D:07:35:80:7F:FB:FC:F8:51:8A:00:95:33:71:05:BA:38:6B:15:3D:D9:27

Alias name: twcarootcertificationauthority
Certificate fingerprints:
  MD5:  AA:08:8F:F6:F9:7B:B7:F2:B1:A7:1E:9B:EA:EA:BD:79
  SHA1: CF:9E:87:6D:D3:EB:FC:42:26:97:A3:B5:A3:7A:A0:76:A9:06:23:48
  SHA256:
 BF:D8:8F:E1:10:1C:41:AE:3E:80:1B:F8:BE:56:35:0E:E9:BA:D1:A6:B9:BD:51:5E:DC:5C:6D:5B:87:11:AC:44

Alias name: baltimorecybertrustca
Certificate fingerprints:
  MD5:  AC:B6:94:A5:9C:17:E0:D7:91:52:9B:B1:97:06:A6:E4
  SHA1: D4:DE:20:D0:5E:66:FC:53:FE:1A:50:88:2C:78:DB:28:52:CA:E4:74
  SHA256:
 16:AF:57:A9:F6:76:B0:AB:12:60:95:AA:5E:BA:DE:F2:2A:B3:11:19:D6:44:AC:95:CD:4B:93:DB:F3:F2:6A:EB

Alias name: cia-crt-g3-01-ca
Certificate fingerprints:
```

```
  MD5:  E3:66:DD:D6:A0:D5:40:8F:FF:29:E2:C0:CB:6E:62:1A
  SHA1: 2B:EE:2C:BA:A3:1D:B5:FE:60:40:41:95:08:ED:46:82:39:4D:ED:E2
  SHA256:
 20:48:AD:4C:EC:90:7F:FA:4A:15:D4:CE:45:E3:C8:E4:2C:EA:78:33:DC:C7:D3:40:48:FC:60:47:27:42:99:EC

Alias name: entrustrootcertificationauthorityg2
Certificate fingerprints:
  MD5:  4B:E2:C9:91:96:65:0C:F4:0E:5A:93:92:A0:0A:FE:B2
  SHA1: 8C:F4:27:FD:79:0C:3A:D1:66:06:8D:E8:1E:57:EF:BB:93:22:72:D4
  SHA256:
 43:DF:57:74:B0:3E:7F:EF:5F:E4:0D:93:1A:7B:ED:F1:BB:2E:6B:42:73:8C:4E:6D:38:41:10:3D:3A:A7:F3:39

Alias name: verisignclass3g4ca
Certificate fingerprints:
  MD5:  3A:52:E1:E7:FD:6F:3A:E3:6F:F3:6F:99:1B:F9:22:41
  SHA1: 22:D5:D8:DF:8F:02:31:D1:8D:F7:9D:B7:CF:8A:2D:64:C9:3F:6C:3A
  SHA256:
 69:DD:D7:EA:90:BB:57:C9:3E:13:5D:C8:5E:A6:FC:D5:48:0B:60:32:39:BD:C4:54:FC:75:8B:2A:26:CF:7F:79

Alias name: xrampglobalcaroot
Certificate fingerprints:
  MD5:  A1:0B:44:B3:CA:10:D8:00:6E:9D:0F:D8:0F:92:0A:D1
  SHA1: B8:01:86:D1:EB:9C:86:A5:41:04:CF:30:54:F3:4C:52:B7:E5:58:C6
  SHA256:
 CE:CD:DC:90:50:99:D8:DA:DF:C5:B1:D2:09:B7:37:CB:E2:C1:8C:FB:2C:10:C0:FF:0B:CF:0D:32:86:FC:1A:A2

Alias name: identrustcommercialrootca1
Certificate fingerprints:
  MD5:  B3:3E:77:73:75:EE:A0:D3:E3:7E:49:63:49:59:BB:C7
  SHA1: DF:71:7E:AA:4A:D9:4E:C9:55:84:99:60:2D:48:DE:5F:BC:F0:3A:25
  SHA256:
 5D:56:49:9B:E4:D2:E0:8B:CF:CA:D0:8A:3E:38:72:3D:50:50:3B:DE:70:69:48:E4:2F:55:60:30:19:E5:28:AE

Alias name: camerfirmachamberscommerceca
Certificate fingerprints:
  MD5:  B0:01:EE:14:D9:AF:29:18:94:76:8E:F1:69:33:2A:84
  SHA1: 6E:3A:55:A4:19:0C:19:5C:93:84:3C:C0:DB:72:2E:31:30:61:F0:B1
  SHA256:
 0C:25:8A:12:A5:67:4A:EF:25:F2:8B:A7:DC:FA:EC:EE:A3:48:E5:41:E6:F5:CC:4E:E6:3B:71:B3:61:60:6A:C3

Alias name: verisignclass3g2ca
Certificate fingerprints:
  MD5:  A2:33:9B:4C:74:78:73:D4:6C:E7:C1:F3:8D:CB:5C:E9
  SHA1: 85:37:1C:A6:E5:50:14:3D:CE:28:03:47:1B:DE:3A:09:E8:F8:77:0F
  SHA256:
 83:CE:3C:12:29:68:8A:59:3D:48:5F:81:97:3C:0F:91:95:43:1E:DA:37:CC:5E:36:43:0E:79:C7:A8:88:63:8B

Alias name: deutschetelekomrootca2
Certificate fingerprints:
  MD5:  74:01:4A:91:B1:08:C4:58:CE:47:CD:F0:DD:11:53:08
  SHA1: 85:A4:08:C0:9C:19:3E:5D:51:58:7D:CD:D6:13:30:FD:8C:DE:37:BF
  SHA256:
 B6:19:1A:50:D0:C3:97:7F:7D:A9:9B:CD:AA:C8:6A:22:7D:AE:B9:67:9E:C7:0B:A3:B0:C9:D9:22:71:C1:70:D3

Alias name: certumca
Certificate fingerprints:
  MD5:  2C:8F:9F:66:1D:18:90:B1:47:26:9D:8E:86:82:8C:A9
  SHA1: 62:52:DC:40:F7:11:43:A2:2F:DE:9E:F7:34:8E:06:42:51:B1:81:18
  SHA256:
 D8:E0:FE:BC:1D:B2:E3:8D:00:94:0F:37:D2:7D:41:34:4D:99:3E:73:4B:99:D5:65:6D:97:78:D4:D8:14:36:24

Alias name: cybertrustglobalroot
Certificate fingerprints:
  MD5:  72:E4:4A:87:E3:69:40:80:77:EA:BC:E3:F4:FF:F0:E1
  SHA1: 5F:43:E5:B1:BF:F8:78:8C:AC:1C:C7:CA:4A:9A:C6:22:2B:CC:34:C6
```

```
  SHA256:
 96:0A:DF:00:63:E9:63:56:75:0C:29:65:DD:0A:08:67:DA:0B:9C:BD:6E:77:71:4A:EA:FB:23:49:AB:39:3D:A3

Alias name: globalsignrootca
Certificate fingerprints:
  MD5:  3E:45:52:15:09:51:92:E1:B7:5D:37:9F:B1:87:29:8A
  SHA1: B1:BC:96:8B:D4:F4:9D:62:2A:A8:9A:81:F2:15:01:52:A4:1D:82:9C
  SHA256:
 EB:D4:10:40:E4:BB:3E:C7:42:C9:E3:81:D3:1E:F2:A4:1A:48:B6:68:5C:96:E7:CE:F3:C1:DF:6C:D4:33:1C:99

Alias name: secomevrootca1
Certificate fingerprints:
  MD5:  22:2D:A6:01:EA:7C:0A:F7:F0:6C:56:43:3F:77:76:D3
  SHA1: FE:B8:C4:32:DC:F9:76:9A:CE:AE:3D:D8:90:8F:FD:28:86:65:64:7D
  SHA256:
 A2:2D:BA:68:1E:97:37:6E:2D:39:7D:72:8A:AE:3A:9B:62:96:B9:FD:BA:60:BC:2E:11:F6:47:F2:C6:75:FB:37

Alias name: globalsignr3ca
Certificate fingerprints:
  MD5:  C5:DF:B8:49:CA:05:13:55:EE:2D:BA:1A:C3:3E:B0:28
  SHA1: D6:9B:56:11:48:F0:1C:77:C5:45:78:C1:09:26:DF:5B:85:69:76:AD
  SHA256:
 CB:B5:22:D7:B7:F1:27:AD:6A:01:13:86:5B:DF:1C:D4:10:2E:7D:07:59:AF:63:5A:7C:F4:72:0D:C9:63:C5:3B

Alias name: staatdernederlandenrootcag3
Certificate fingerprints:
  MD5:  0B:46:67:07:DB:10:2F:19:8C:35:50:60:D1:0B:F4:37
  SHA1: D8:EB:6B:41:51:92:59:E0:F3:E7:85:00:C0:3D:B6:88:97:C9:EE:FC
  SHA256:
 3C:4F:B0:B9:5A:B8:B3:00:32:F4:32:B8:6F:53:5F:E1:72:C1:85:D0:FD:39:86:58:37:CF:36:18:7F:A6:F4:28

Alias name: staatdernederlandenrootcag2
Certificate fingerprints:
  MD5:  7C:A5:0F:F8:5B:9A:7D:6D:30:AE:54:5A:E3:42:A2:8A
  SHA1: 59:AF:82:79:91:86:C7:B4:75:07:CB:CF:03:57:46:EB:04:DD:B7:16
  SHA256:
 66:8C:83:94:7D:A6:3B:72:4B:EC:E1:74:3C:31:A0:E6:AE:D0:DB:8E:C5:B3:1B:E3:77:BB:78:4F:91:B6:71:6F

Alias name: aolrootca2
Certificate fingerprints:
  MD5:  D6:ED:3C:CA:E2:66:0F:AF:10:43:0D:77:9B:04:09:BF
  SHA1: 85:B5:FF:67:9B:0C:79:96:1F:C8:6E:44:22:00:46:13:DB:17:92:84
  SHA256:
 7D:3B:46:5A:60:14:E5:26:C0:AF:FC:EE:21:27:D2:31:17:27:AD:81:1C:26:84:2D:00:6A:F3:73:06:CC:80:BD

Alias name: dstrootcax3
Certificate fingerprints:
  MD5:  41:03:52:DC:0F:F7:50:1B:16:F0:02:8E:BA:6F:45:C5
  SHA1: DA:C9:02:4F:54:D8:F6:DF:94:93:5F:B1:73:26:38:CA:6A:D7:7C:13
  SHA256:
 06:87:26:03:31:A7:24:03:D9:09:F1:05:E6:9B:CF:0D:32:E1:BD:24:93:FF:C6:D9:20:6D:11:BC:D6:77:07:39

Alias name: trustcenteruniversalcai
Certificate fingerprints:
  MD5:  45:E1:A5:72:C5:A9:36:64:40:9E:F5:E4:58:84:67:8C
  SHA1: 6B:2F:34:AD:89:58:BE:62:FD:B0:6B:5C:CE:BB:9D:D9:4F:4E:39:F3
  SHA256:
 EB:F3:C0:2A:87:89:B1:FB:7D:51:19:95:D6:63:B7:29:06:D9:13:CE:0D:5E:10:56:8A:8A:77:E2:58:61:67:E7

Alias name: aolrootca1
Certificate fingerprints:
  MD5:  14:F1:08:AD:9D:FA:64:E2:89:E7:1C:CF:A8:AD:7D:5E
  SHA1: 39:21:C1:15:C1:5D:0E:CA:5C:CB:5B:C4:F0:7D:21:D8:05:0B:56:6A
  SHA256:
 77:40:73:12:C6:3A:15:3D:5B:C0:0B:4E:51:75:9C:DF:DA:C2:37:DC:2A:33:B6:79:46:E9:8E:9B:FA:68:0A:E3
```

```
Alias name: affirmtrustpremiumecc
Certificate fingerprints:
  MD5:  64:B0:09:55:CF:B1:D5:99:E2:BE:13:AB:A6:5D:EA:4D
  SHA1: B8:23:6B:00:2F:1D:16:86:53:01:55:6C:11:A4:37:CA:EB:FF:C3:BB
  SHA256:
 BD:71:FD:F6:DA:97:E4:CF:62:D1:64:7A:DD:25:81:B0:7D:79:AD:F8:39:7E:B4:EC:BA:9C:5E:84:88:82:14:23

Alias name: microseceszignorootca2009
Certificate fingerprints:
  MD5:  F8:49:F4:03:BC:44:2D:83:BE:48:69:7D:29:64:FC:B1
  SHA1: 89:DF:74:FE:5C:F4:0F:4A:80:F9:E3:37:7D:54:DA:91:E1:01:31:8E
  SHA256:
 3C:5F:81:FE:A5:FA:B8:2C:64:BF:A2:EA:EC:AF:CD:E8:E0:77:FC:86:20:A7:CA:E5:37:16:3D:F3:6E:DB:F3:78

Alias name: verisignclass1g3ca
Certificate fingerprints:
  MD5:  B1:47:BC:18:57:D1:18:A0:78:2D:EC:71:E8:2A:95:73
  SHA1: 20:42:85:DC:F7:EB:76:41:95:57:8E:13:6B:D4:B7:D1:E9:8E:46:A5
  SHA256:
 CB:B5:AF:18:5E:94:2A:24:02:F9:EA:CB:C0:ED:5B:B8:76:EE:A3:C1:22:36:23:D0:04:47:E4:F3:BA:55:4B:65

Alias name: certplusrootcag2
Certificate fingerprints:
  MD5:  A7:EE:C4:78:2D:1B:EE:2D:B9:29:CE:D6:A7:96:32:31
  SHA1: 4F:65:8E:1F:E9:06:D8:28:02:E9:54:47:41:C9:54:25:5D:69:CC:1A
  SHA256:
 6C:C0:50:41:E6:44:5E:74:69:6C:4C:FB:C9:F8:0F:54:3B:7E:AB:BB:44:B4:CE:6F:78:7C:6A:99:71:C4:2F:17

Alias name: certplusrootcag1
Certificate fingerprints:
  MD5:  7F:09:9C:F7:D9:B9:5C:69:69:56:D5:37:3E:14:0D:42
  SHA1: 22:FD:D0:B7:FD:A2:4E:0D:AC:49:2C:A0:AC:A6:7B:6A:1F:E3:F7:66
  SHA256:
 15:2A:40:2B:FC:DF:2C:D5:48:05:4D:22:75:B3:9C:7F:CA:3E:C0:97:80:78:B0:F0:EA:76:E5:61:A6:C7:43:3E

Alias name: addtrustexternalca
Certificate fingerprints:
  MD5:  1D:35:54:04:85:78:B0:3F:42:42:4D:BF:20:73:0A:3F
  SHA1: 02:FA:F3:E2:91:43:54:68:60:78:57:69:4D:F5:E4:5B:68:85:18:68
  SHA256:
 68:7F:A4:51:38:22:78:FF:F0:C8:B1:1F:8D:43:D5:76:67:1C:6E:B2:BC:EA:B4:13:FB:83:D9:65:D0:6D:2F:F2

Alias name: entrustrootcertificationauthority
Certificate fingerprints:
  MD5:  D6:A5:C3:ED:5D:DD:3E:00:C1:3D:87:92:1F:1D:3F:E4
  SHA1: B3:1E:B1:B7:40:E3:6C:84:02:DA:DC:37:D4:4D:F5:D4:67:49:52:F9
  SHA256:
 73:C1:76:43:4F:1B:C6:D5:AD:F4:5B:0E:76:E7:27:28:7C:8D:E5:76:16:C1:E6:E6:14:1A:2B:2C:BC:7D:8E:4C

Alias name: verisignclass3ca
Certificate fingerprints:
  MD5:  EF:5A:F1:33:EF:F1:CD:BB:51:02:EE:12:14:4B:96:C4
  SHA1: A1:DB:63:93:91:6F:17:E4:18:55:09:40:04:15:C7:02:40:B0:AE:6B
  SHA256:
 A4:B6:B3:99:6F:C2:F3:06:B3:FD:86:81:BD:63:41:3D:8C:50:09:CC:4F:A3:29:C2:CC:F0:E2:FA:1B:14:03:05

Alias name: digicertassuredidrootca
Certificate fingerprints:
  MD5:  87:CE:0B:7B:2A:0E:49:00:E1:58:71:9B:37:A8:93:72
  SHA1: 05:63:B8:63:0D:62:D7:5A:BB:C8:AB:1E:4B:DF:B5:A8:99:B2:4D:43
  SHA256:
 3E:90:99:B5:01:5E:8F:48:6C:00:BC:EA:9D:11:1E:E7:21:FA:BA:35:5A:89:BC:F1:DF:69:56:1E:3D:C6:32:5C

Alias name: globalsignrootcar3
Certificate fingerprints:
  MD5:  C5:DF:B8:49:CA:05:13:55:EE:2D:BA:1A:C3:3E:B0:28
```

```
  SHA1: D6:9B:56:11:48:F0:1C:77:C5:45:78:C1:09:26:DF:5B:85:69:76:AD
  SHA256:
 CB:B5:22:D7:B7:F1:27:AD:6A:01:13:86:5B:DF:1C:D4:10:2E:7D:07:59:AF:63:5A:7C:F4:72:0D:C9:63:C5:3B

Alias name: globalsignrootcar2
Certificate fingerprints:
  MD5:  94:14:77:7E:3E:5E:FD:8F:30:BD:41:B0:CF:E7:D0:30
  SHA1: 75:E0:AB:B6:13:85:12:27:1C:04:F8:5F:DD:DE:38:E4:B7:24:2E:FE
  SHA256:
 CA:42:DD:41:74:5F:D0:B8:1E:B9:02:36:2C:F9:D8:BF:71:9D:A1:BD:1B:1E:FC:94:6F:5B:4C:99:F4:2C:1B:9E

Alias name: verisignclass1ca
Certificate fingerprints:
  MD5:  86:AC:DE:2B:C5:6D:C3:D9:8C:28:88:D3:8D:16:13:1E
  SHA1: CE:6A:64:A3:09:E4:2F:BB:D9:85:1C:45:3E:64:09:EA:E8:7D:60:F1
  SHA256:
 51:84:7C:8C:BD:2E:9A:72:C9:1E:29:2D:2A:E2:47:D7:DE:1E:3F:D2:70:54:7A:20:EF:7D:61:0F:38:B8:84:2C

Alias name: thawtepremiumserverca
Certificate fingerprints:
  MD5:  A6:6B:60:90:23:9B:3F:2D:BB:98:6F:D6:A7:19:0D:46
  SHA1: E0:AB:05:94:20:72:54:93:05:60:62:02:36:70:F7:CD:2E:FC:66:66
  SHA256:
 3F:9F:27:D5:83:20:4B:9E:09:C8:A3:D2:06:6C:4B:57:D3:A2:47:9C:36:93:65:08:80:50:56:98:10:5D:BC:E9

Alias name: verisigntsaca
Certificate fingerprints:
  MD5:  F2:89:95:6E:4D:05:F0:F1:A7:21:55:7D:46:11:BA:47
  SHA1: 20:CE:B1:F0:F5:1C:0E:19:A9:F3:8D:B1:AA:8E:03:8C:AA:7A:C7:01
  SHA256:
 CB:6B:05:D9:E8:E5:7C:D8:82:B1:0B:4D:B7:0D:E4:BB:1D:E4:2B:A4:8A:7B:D0:31:8B:63:5B:F6:E7:78:1A:9D

Alias name: thawteprimaryrootca
Certificate fingerprints:
  MD5:  8C:CA:DC:0B:22:CE:F5:BE:72:AC:41:1A:11:A8:D8:12
  SHA1: 91:C6:D6:EE:3E:8A:C8:63:84:E5:48:C2:99:29:5C:75:6C:81:7B:81
  SHA256:
 8D:72:2F:81:A9:C1:13:C0:79:1D:F1:36:A2:96:6D:B2:6C:95:0A:97:1D:B4:6B:41:99:F4:EA:54:B7:8B:FB:9F

Alias name: visaecommerceroot
Certificate fingerprints:
  MD5:  FC:11:B8:D8:08:93:30:00:6D:23:F9:7E:EB:52:1E:02
  SHA1: 70:17:9B:86:8C:00:A4:FA:60:91:52:22:3F:9F:3E:32:BD:E0:05:62
  SHA256:
 69:FA:C9:BD:55:FB:0A:C7:8D:53:BB:EE:5C:F1:D5:97:98:9F:D0:AA:AB:20:A2:51:51:BD:F1:73:3E:E7:D1:22

Alias name: digicertglobalrootg3
Certificate fingerprints:
  MD5:  F5:5D:A4:50:A5:FB:28:7E:1E:0F:0D:CC:96:57:56:CA
  SHA1: 7E:04:DE:89:6A:3E:66:6D:00:E6:87:D3:3F:FA:D9:3B:E8:3D:34:9E
  SHA256:
 31:AD:66:48:F8:10:41:38:C7:38:F3:9E:A4:32:01:33:39:3E:3A:18:CC:02:29:6E:F9:7C:2A:C9:EF:67:31:D0

Alias name: xrampglobalca
Certificate fingerprints:
  MD5:  A1:0B:44:B3:CA:10:D8:00:6E:9D:0F:D8:0F:92:0A:D1
  SHA1: B8:01:86:D1:EB:9C:86:A5:41:04:CF:30:54:F3:4C:52:B7:E5:58:C6
  SHA256:
 CE:CD:DC:90:50:99:D8:DA:DF:C5:B1:D2:09:B7:37:CB:E2:C1:8C:FB:2C:10:C0:FF:0B:CF:0D:32:86:FC:1A:A2

Alias name: digicertglobalrootg2
Certificate fingerprints:
  MD5:  E4:A6:8A:C8:54:AC:52:42:46:0A:FD:72:48:1B:2A:44
  SHA1: DF:3C:24:F9:BF:D6:66:76:1B:26:80:73:FE:06:D1:CC:8D:4F:82:A4
  SHA256:
 CB:3C:CB:B7:60:31:E5:E0:13:8F:8D:D3:9A:23:F9:DE:47:FF:C3:5E:43:C1:14:4C:EA:27:D4:6A:5A:B1:CB:5F
```

```
Alias name: valicertclass2ca
Certificate fingerprints:
  MD5:  A9:23:75:9B:BA:49:36:6E:31:C2:DB:F2:E7:66:BA:87
  SHA1: 31:7A:2A:D0:7F:2B:33:5E:F5:A1:C3:4E:4B:57:E8:B7:D8:F1:FC:A6
  SHA256:
 58:D0:17:27:9C:D4:DC:63:AB:DD:B1:96:A6:C9:90:6C:30:C4:E0:87:83:EA:E8:C1:60:99:54:D6:93:55:59:6B

Alias name: geotrustprimaryca
Certificate fingerprints:
  MD5:  02:26:C3:01:5E:08:30:37:43:A9:D0:7D:CF:37:E6:BF
  SHA1: 32:3C:11:8E:1B:F7:B8:B6:52:54:E2:E2:10:0D:D6:02:90:37:F0:96
  SHA256:
 37:D5:10:06:C5:12:EA:AB:62:64:21:F1:EC:8C:92:01:3F:C5:F8:2A:E9:8E:E5:33:EB:46:19:B8:DE:B4:D0:6C

Alias name: netlockaranyclassgoldfotanusitvany
Certificate fingerprints:
  MD5:  C5:A1:B7:FF:73:DD:D6:D7:34:32:18:DF:FC:3C:AD:88
  SHA1: 06:08:3F:59:3F:15:A1:04:A0:69:A4:6B:A9:03:D0:06:B7:97:09:91
  SHA256:
 6C:61:DA:C3:A2:DE:F0:31:50:6B:E0:36:D2:A6:FE:40:19:94:FB:D1:3D:F9:C8:D4:66:59:92:74:C4:46:EC:98

Alias name: geotrustglobalca
Certificate fingerprints:
  MD5:  F7:75:AB:29:FB:51:4E:B7:77:5E:FF:05:3C:99:8E:F5
  SHA1: DE:28:F4:A4:FF:E5:B9:2F:A3:C5:03:D1:A3:49:A7:F9:96:2A:82:12
  SHA256:
 FF:85:6A:2D:25:1D:CD:88:D3:66:56:F4:50:12:67:98:CF:AB:AA:DE:40:79:9C:72:2D:E4:D2:B5:DB:36:A7:3A

Alias name: oistewisekeyglobalrootgbca
Certificate fingerprints:
  MD5:  A4:EB:B9:61:28:2E:B7:2F:98:B0:35:26:90:99:51:1D
  SHA1: 0F:F9:40:76:18:D3:D7:6A:4B:98:F0:A8:35:9E:0C:FD:27:AC:CC:ED
  SHA256:
 6B:9C:08:E8:6E:B0:F7:67:CF:AD:65:CD:98:B6:21:49:E5:49:4A:67:F5:84:5E:7B:D1:ED:01:9F:27:B8:6B:D6

Alias name: certumtrustednetworkca2
Certificate fingerprints:
  MD5:  6D:46:9E:D9:25:6D:08:23:5B:5E:74:7D:1E:27:DB:F2
  SHA1: D3:DD:48:3E:2B:BF:4C:05:E8:AF:10:F5:FA:76:26:CF:D3:DC:30:92
  SHA256:
 B6:76:F2:ED:DA:E8:77:5C:D3:6C:B0:F6:3C:D1:D4:60:39:61:F4:9E:62:65:BA:01:3A:2F:03:07:B6:D0:B8:04

Alias name: starfieldservicesrootcertificateauthorityg2
Certificate fingerprints:
  MD5:  17:35:74:AF:7B:61:1C:EB:F4:F9:3C:E2:EE:40:F9:A2
  SHA1: 92:5A:8F:8D:2C:6D:04:E0:66:5F:59:6A:FF:22:D8:63:E8:25:6F:3F
  SHA256:
 56:8D:69:05:A2:C8:87:08:A4:B3:02:51:90:ED:CF:ED:B1:97:4A:60:6A:13:C6:E5:29:0F:CB:2A:E6:3E:DA:B5

Alias name: comodorsacertificationauthority
Certificate fingerprints:
  MD5:  1B:31:B0:71:40:36:CC:14:36:91:AD:C4:3E:FD:EC:18
  SHA1: AF:E5:D2:44:A8:D1:19:42:30:FF:47:9F:E2:F8:97:BB:CD:7A:8C:B4
  SHA256:
 52:F0:E1:C4:E5:8E:C6:29:29:1B:60:31:7F:07:46:71:B8:5D:7E:A8:0D:5B:07:27:34:63:53:4B:32:B4:02:34

Alias name: comodoaaaca
Certificate fingerprints:
  MD5:  49:79:04:B0:EB:87:19:AC:47:B0:BC:11:51:9B:74:D0
  SHA1: D1:EB:23:A4:6D:17:D6:8F:D9:25:64:C2:F1:F1:60:17:64:D8:E3:49
  SHA256:
 D7:A7:A0:FB:5D:7E:27:31:D7:71:E9:48:4E:BC:DE:F7:1D:5F:0C:3E:0A:29:48:78:2B:C8:3E:E0:EA:69:9E:F4

Alias name: identrustpublicsectorrootca1
Certificate fingerprints:
```

```
  MD5:  37:06:A5:B0:FC:89:9D:BA:F4:6B:8C:1A:64:CD:D5:BA
  SHA1: BA:29:41:60:77:98:3F:F4:F3:EF:F2:31:05:3B:2E:EA:6D:4D:45:FD
  SHA256:
 30:D0:89:5A:9A:44:8A:26:20:91:63:55:22:D1:F5:20:10:B5:86:7A:CA:E1:2C:78:EF:95:8F:D4:F4:38:9F:2F

Alias name: certplusclass2primaryca
Certificate fingerprints:
  MD5:  88:2C:8C:52:B8:A2:3C:F3:F7:BB:03:EA:AE:AC:42:0B
  SHA1: 74:20:74:41:72:9C:DD:92:EC:79:31:D8:23:10:8D:C2:81:92:E2:BB
  SHA256:
 0F:99:3C:8A:EF:97:BA:AF:56:87:14:0E:D5:9A:D1:82:1B:B4:AF:AC:F0:AA:9A:58:B5:D5:7A:33:8A:3A:FB:CB

Alias name: ttelesecglobalrootclass2ca
Certificate fingerprints:
  MD5:  2B:9B:9E:E4:7B:6C:1F:00:72:1A:CC:C1:77:79:DF:6A
  SHA1: 59:0D:2D:7D:88:4F:40:2E:61:7E:A5:62:32:17:65:CF:17:D8:94:E9
  SHA256:
 91:E2:F5:78:8D:58:10:EB:A7:BA:58:73:7D:E1:54:8A:8E:CA:CD:01:45:98:BC:0B:14:3E:04:1B:17:05:25:52

Alias name: accvraiz1
Certificate fingerprints:
  MD5:  D0:A0:5A:EE:05:B6:09:94:21:A1:7D:F1:B2:29:82:02
  SHA1: 93:05:7A:88:15:C6:4F:CE:88:2F:FA:91:16:52:28:78:BC:53:64:17
  SHA256:
 9A:6E:C0:12:E1:A7:DA:9D:BE:34:19:4D:47:8A:D7:C0:DB:18:22:FB:07:1D:F1:29:81:49:6E:D1:04:38:41:13

Alias name: digicerthighassuranceevrootca
Certificate fingerprints:
  MD5:  D4:74:DE:57:5C:39:B2:D3:9C:85:83:C5:C0:65:49:8A
  SHA1: 5F:B7:EE:06:33:E2:59:DB:AD:0C:4C:9A:E6:D3:8F:1A:61:C7:DC:25
  SHA256:
 74:31:E5:F4:C3:C1:CE:46:90:77:4F:0B:61:E0:54:40:88:3B:A9:A0:1E:D0:0B:A6:AB:D7:80:6E:D3:B1:18:CF

Alias name: amzninternalinfoseccag3
Certificate fingerprints:
  MD5:  E9:34:94:02:BA:BB:31:6B:22:E6:2B:A9:C4:F0:26:04
  SHA1: B9:B1:CA:38:F7:BF:9C:D2:D4:95:E7:B6:5E:75:32:9B:A8:78:2E:F6
  SHA256:
 81:03:0B:C7:E2:54:DA:7B:F8:B7:45:DB:DD:41:15:89:B5:A3:81:86:FB:4B:29:77:1F:84:0A:18:D9:67:6D:68

Alias name: cia-crt-g3-02-ca
Certificate fingerprints:
  MD5:  FD:B9:23:FD:D3:EB:2D:3E:57:EF:56:FF:DB:D3:E4:B9
  SHA1: 96:4A:BB:A7:BD:DA:FC:97:34:C0:0A:2D:F0:05:98:F7:E6:C6:6F:09
  SHA256:
 93:F1:72:FB:BA:43:31:5C:06:EE:0F:9F:04:89:B8:F6:88:BC:75:15:3C:BE:B4:80:AC:A7:14:3A:F6:FC:4A:C1

Alias name: entrustrootcertificationauthorityec1
Certificate fingerprints:
  MD5:  B6:7E:1D:F0:58:C5:49:6C:24:3B:3D:ED:98:18:ED:BC
  SHA1: 20:D8:06:40:DF:9B:25:F5:12:25:3A:11:EA:F7:59:8A:EB:14:B5:47
  SHA256:
 02:ED:0E:B2:8C:14:DA:45:16:5C:56:67:91:70:0D:64:51:D7:FB:56:F0:B2:AB:1D:3B:8E:B0:70:E5:6E:DF:F5

Alias name: securitycommunicationrootca
Certificate fingerprints:
  MD5:  F1:BC:63:6A:54:E0:B5:27:F5:CD:E7:1A:E3:4D:6E:4A
  SHA1: 36:B1:2B:49:F9:81:9E:D7:4C:9E:BC:38:0F:C6:56:8F:5D:AC:B2:F7
  SHA256:
 E7:5E:72:ED:9F:56:0E:EC:6E:B4:80:00:73:A4:3F:C3:AD:19:19:5A:39:22:82:01:78:95:97:4A:99:02:6B:6C

Alias name: globalsignca
Certificate fingerprints:
  MD5:  3E:45:52:15:09:51:92:E1:B7:5D:37:9F:B1:87:29:8A
  SHA1: B1:BC:96:8B:D4:F4:9D:62:2A:A8:9A:81:F2:15:01:52:A4:1D:82:9C
```

```
  SHA256:
 EB:D4:10:40:E4:BB:3E:C7:42:C9:E3:81:D3:1E:F2:A4:1A:48:B6:68:5C:96:E7:CE:F3:C1:DF:6C:D4:33:1C:99

Alias name: trustcenterclass2caii
Certificate fingerprints:
  MD5:  CE:78:33:5C:59:78:01:6E:18:EA:B9:36:A0:B9:2E:23
  SHA1: AE:50:83:ED:7C:F4:5C:BC:8F:61:C6:21:FE:68:5D:79:42:21:15:6E
  SHA256:
 E6:B8:F8:76:64:85:F8:07:AE:7F:8D:AC:16:70:46:1F:07:C0:A1:3E:EF:3A:1F:F7:17:53:8D:7A:BA:D3:91:B4

Alias name: camerfirmachambersofcommerceroot
Certificate fingerprints:
  MD5:  B0:01:EE:14:D9:AF:29:18:94:76:8E:F1:69:33:2A:84
  SHA1: 6E:3A:55:A4:19:0C:19:5C:93:84:3C:C0:DB:72:2E:31:30:61:F0:B1
  SHA256:
 0C:25:8A:12:A5:67:4A:EF:25:F2:8B:A7:DC:FA:EC:EE:A3:48:E5:41:E6:F5:CC:4E:E6:3B:71:B3:61:60:6A:C3

Alias name: geotrustprimarycag3
Certificate fingerprints:
  MD5:  B5:E8:34:36:C9:10:44:58:48:70:6D:2E:83:D4:B8:05
  SHA1: 03:9E:ED:B8:0B:E7:A0:3C:69:53:89:3B:20:D2:D9:32:3A:4C:2A:FD
  SHA256:
 B4:78:B8:12:25:0D:F8:78:63:5C:2A:A7:EC:7D:15:5E:AA:62:5E:E8:29:16:E2:CD:29:43:61:88:6C:D1:FB:D4

Alias name: geotrustprimarycag2
Certificate fingerprints:
  MD5:  01:5E:D8:6B:BD:6F:3D:8E:A1:31:F8:12:E0:98:73:6A
  SHA1: 8D:17:84:D5:37:F3:03:7D:EC:70:FE:57:8B:51:9A:99:E6:10:D7:B0
  SHA256:
 5E:DB:7A:C4:3B:82:A0:6A:87:61:E8:D7:BE:49:79:EB:F2:61:1F:7D:D7:9B:F9:1C:1C:6B:56:6A:21:9E:D7:66

Alias name: hongkongpostrootca1
Certificate fingerprints:
  MD5:  A8:0D:6F:39:78:B9:43:6D:77:42:6D:98:5A:CC:23:CA
  SHA1: D6:DA:A8:20:8D:09:D2:15:4D:24:B5:2F:CB:34:6E:B2:58:B2:8A:58
  SHA256:
 F9:E6:7D:33:6C:51:00:2A:C0:54:C6:32:02:2D:66:DD:A2:E7:E3:FF:F1:0A:D0:61:ED:31:D8:BB:B4:10:CF:B2

Alias name: affirmtrustpremiumeccca
Certificate fingerprints:
  MD5:  64:B0:09:55:CF:B1:D5:99:E2:BE:13:AB:A6:5D:EA:4D
  SHA1: B8:23:6B:00:2F:1D:16:86:53:01:55:6C:11:A4:37:CA:EB:FF:C3:BB
  SHA256:
 BD:71:FD:F6:DA:97:E4:CF:62:D1:64:7A:DD:25:81:B0:7D:79:AD:F8:39:7E:B4:EC:BA:9C:5E:84:88:82:14:23

Alias name: hellenicacademicandresearchinstitutionsrootca2015
Certificate fingerprints:
  MD5:  CA:FF:E2:DB:03:D9:CB:4B:E9:0F:AD:84:FD:7B:18:CE
  SHA1: 01:0C:06:95:A6:98:19:14:FF:BF:5F:C6:B0:B6:95:EA:29:E9:12:A6
  SHA256:
 A0:40:92:9A:02:CE:53:B4:AC:F4:F2:FF:C6:98:1C:E4:49:6F:75:5E:6D:45:FE:0B:2A:69:2B:CD:52:52:3F:36
```

Réduction des coûts de messagerie avec Basic Ingest

Basic Ingest vous permet d'envoyer en toute sécurité les données d'appareil versAWSservices pris en charge parActions de règle AWS IoT (p. 401)sans encourircoûts de messagerie. Basic Ingest optimise le flux de données en supprimant l'agent messages de publication/abonnement à partir du chemin d'intégration, il est donc plus rentable.

Pour utiliser Basic Ingest, vous devez envoyer des messages à partir de vos appareils ou applications avec des noms de rubriques qui commencent par `$aws/rules/rule-name` comme trois premiers niveaux, où *`rule-name`* est le nom de votre règle AWS IoT à déclencher.

Vous pouvez utiliser une règle existante avec Basic Ingest en ajoutant simplement le préfixe Basic Ingest (`$aws/rules/rule-name`) à la rubrique du message par lequel vous déclenchez normalement la règle. Par exemple, si vous avez une règle nommée `BuildingManager` qui est déclenchée par des messages avec des sujets tels que `Buildings/Building5/Floor2/Room201/Lights` (`"sql": "SELECT * FROM 'Buildings/#'"`), vous pouvez déclencher la même règle avec Basic Ingest en envoyant un message avec le sujet `$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`.

Sachez que :

- Vos appareils et vos règles ne peuvent pas s'abonner aux rubriques réservées Basic Ingest. Pour plus d'informations, consultez Rubriques réservées (p. 91).
- Si vous avez besoin d'un agent de publication/abonnement pour distribuer des messages à plusieurs abonnés (par exemple, pour envoyer des messages à d'autres appareils, ainsi qu'au moteur de règles), vous devez continuer à utiliser l'agent de messages AWS IoT pour gérer la distribution des messages. Il suffit de publier vos messages dans des rubriques autres que les rubriques Basic Ingest.

Utilisation de Basic Ingest

Assurez-vous que votre appareil ou application utilise une stratégie (p. 270) qui dispose d'autorisations de publication dans `$aws/rules/*`. Vous pouvez également spécifier l'autorisation pour des règles individuelles avec `$aws/rules/rule-name/*` dans la stratégie. Sinon, vos appareils et applications peuvent continuer à utiliser leurs connexions existantes avec AWS IoT Core .

Lorsque le message atteint le moteur de règles, il n'existe pas de différence dans l'exécution ou la gestion des erreurs entre les règles déclenchées à partir de Basic Ingest et celles déclenchées via des abonnements d'agent de messages.

Bien sûr, vous pouvez créer des règles à utiliser avec Basic Ingest. Gardez à l'esprit les points suivants :

- Le préfixe initial d'une rubrique Basic Ingest (`$aws/rules/rule-name`) n'est pas disponible pour la fonction topic(Decimal) (p. 533).
- Si vous définissez une règle qui est déclenchée uniquement avec Basic Ingest, la clause `FROM` est facultative dans le champ `sql` de la définition `rule`. Elle est encore requise si la règle est également déclenchée par d'autres messages qui doivent être envoyés via l'agent de messages (par exemple, parce que ces autres messages doivent être distribués à plusieurs abonnés). Pour plus d'informations, consultez Référence SQL AWS IoT (p. 482).
- Les trois premiers niveaux de la rubrique Basic Ingest (`$aws/rules/rule-name`) ne sont pas comptabilisés dans la limite de longueur de huit segments ou de 256 caractères maximum pour une rubrique. Sinon, les mêmes restrictions que celles documentées dansAWS IoTRestrictions.
- Si un message est reçu avec une rubrique Basic Ingest qui spécifie une règle inactive ou une règle qui n'existe pas, un journal d'erreur est créé dans un journal Amazon CloudWatch Log afin de vous aider pour le débogage. Pour plus d'informations, consultez Entrées de journal du moteur de règles (p. 380). Une métrique `RuleNotFound` est indiquée et vous pouvez créer des alarmes sur cette métrique. Pour de plus amples informations, veuillez consulter Métriques dans Métriques de règle (p. 364).
- Vous pouvez toujours publier avec QoS 1 dans des rubriques Basic Ingest. Vous recevez un PUBACK une fois que le message a été transmis avec succès au moteur de règles. La réception d'un PUBACK ne signifie pas que vos actions de règle se sont terminées avec succès. Vous pouvez configurer une action d'erreur pour gérer les erreurs lors de l'exécution de l'action. Voir Gestion des erreurs (action d'erreur) (p. 452).

Référence SQL AWS IoT

Dans AWS IoT, les règles sont définies à l'aide d'une syntaxe de type SQL. Les instructions SQL se composent de trois types de clauses :

SELECT

Obligatoire. Extrait les informations de la charge utile d'un message entrant et effectue des transformations sur les informations. Les messages à utiliser sont identifiés par leFiltre de rubriques (p. 90)spécifiée dans la clause FROM.

La clause SELECT accepteTypes de données (p. 485),Operators (p. 489),Functions (p. 495),Literals (p. 539),Instructions Case (p. 539),Extensions JSON (p. 540),Modèles de substitution (p. 541),Requêtes d'objets imbriqués (p. 543), etCharges utiles binaires (p. 544).

DE

Message MQTTFiltre de rubriques (p. 90)qui identifie les messages à partir desquels extraire les données. La règle est déclenchée pour chaque message envoyé à une rubrique MQTT qui correspond au filtre de rubrique défini ici. Requis pour les règles qui sont déclenchées par les messages qui transitent par le courtier de messages. Facultatif pour les règles qui sont déclenchées uniquement à l'aide de la fonction Basic Ingest (p. 480).

WHERE

(Facultatif) Ajoute une logique conditionnelle qui détermine si les actions spécifiées par une règle sont exécutées.

La clause WHERE prend en charge Types de données (p. 485), Operators (p. 489), Functions (p. 495), Literals (p. 539), Instructions Case (p. 539), Extensions JSON (p. 540), Modèles de substitution (p. 541) et Requêtes d'objets imbriqués (p. 543).

Un exemple d'instruction SQL se présente sous la forme suivante :

```
SELECT color AS rgb FROM 'topic/subtopic' WHERE temperature > 50
```

Un exemple de message MQTT (également appelé charge entrante) se présente sous la forme suivante :

```
{
    "color":"red",
    "temperature":100
}
```

Si ce message est publié dans la rubrique `'topic/subtopic'`, la règle est déclenchée et l'instruction SQL est évaluée. L'instruction SQL extrait la valeur de la propriété `color` si la propriété `"temperature"` est supérieure à 50. La clause WHERE spécifie la condition `temperature > 50`. Le mot-clé `AS` change le nom de la propriété `"color"` en `"rgb"`. Le résultat (également appelé charge sortante) se présente sous la forme suivante :

```
{
    "rgb":"red"
}
```

Ces données sont alors transférées vers l'action de la règle qui envoie les données pour traitement supplémentaire. Pour plus d'informations sur les actions de règle, consultez Actions de règle AWS IoT (p. 401).

Clause SELECT

La clause SELECT AWS IoT est essentiellement la même que la clause SELECT SQL ANSI, avec certaines différences mineures.

La clause SELECT accepteTypes de données (p. 485),Operators (p. 489),Functions (p. 495),Literals (p. 539),Instructions Case (p. 539),Extensions JSON (p. 540),Modèles de substitution (p. 541),Requêtes d'objets imbriqués (p. 543), etCharges utiles binaires (p. 544).

Vous pouvez utiliser la clause SELECT pour extraire les informations des messages entrants MQTT. Une clause `SELECT *` peut être utilisée pour récupérer toute la charge utile du message entrant. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT * FROM 'topic/subtopic'
Outgoing payload: {"color":"red", "temperature":50}
```

Si la charge utile est un objet JSON, vous pouvez référencer des clés dans l'objet. Votre charge utile sortante contient la paire clé-valeur. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL statement: SELECT color FROM 'topic/subtopic'
Outgoing payload: {"color":"red"}
```

Vous pouvez utiliser le même mot-clé AS pour renommer des clés. Exemples :

```
Incoming payload published on topic 'topic/subtopic':{"color":"red", "temperature":50}
SQL:SELECT color AS my_color FROM 'topic/subtopic'
Outgoing payload: {"my_color":"red"}
```

Vous pouvez sélectionner plusieurs éléments en les séparant par une virgule. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT color as my_color, temperature as fahrenheit FROM 'topic/subtopic'
Outgoing payload: {"my_color":"red","fahrenheit":50}
```

Vous pouvez sélectionner plusieurs éléments comprenant « * » pour ajouter des éléments dans la charge utile entrante. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT *, 15 as speed FROM 'topic/subtopic'
Outgoing payload: {"color":"red", "temperature":50, "speed":15}
```

Vous pouvez utiliser le mot-clé `"VALUE"` pour produire les charges utiles sortantes qui ne sont pas des objets JSON. Avec la version SQL `2015-10-08`, vous ne pouvez sélectionner qu'un seul élément. Avec la version SQL `2016-03-23` ou une version ultérieure, vous pouvez également sélectionner un tableau à afficher en tant qu'objet de niveau supérieur.

Example

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT VALUE color FROM 'topic/subtopic'
Outgoing payload: "red"
```

Vous pouvez utiliser une syntaxe `'.'` pour explorer des objets JSON imbriqués dans la charge utile entrante. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":
{"red":255,"green":0,"blue":0}, "temperature":50}
SQL: SELECT color.red as red_value FROM 'topic/subtopic'
Outgoing payload: {"red_value":255}
```

Pour plus d'informations sur l'utilisation des noms d'objets et de propriétés JSON qui incluent des caractères réservés, tels que des nombres ou le trait d'union (moins), consultezExtensions JSON (p. 540)

Vous pouvez utiliser des fonctions (voir Functions (p. 495)) pour transformer la charge utile entrante. Vous pouvez utiliser des parenthèses pour le regroupement. Exemples :

```
Incoming payload published on topic 'topic/subtopic': {"color":"red", "temperature":50}
SQL: SELECT (temperature - 32) * 5 / 9 AS celsius, upper(color) as my_color FROM 'topic/
subtopic'
Outgoing payload: {"celsius":10,"my_color":"RED"}
```

Clause FROM

La clause FROM abonne votre règle à unetopic (p. 89)ouFiltre de rubriques (p. 90). Vous devez placer la rubrique ou le filtre de rubrique entre guillemets simples ('). La règle est déclenchée pour chaque message envoyé à une rubrique MQTT qui correspond au filtre de rubrique défini ici. Un filtre de rubriques permet de s'abonner à un groupe de rubriques similaires.

Exemple :

Charge utile entrante publiée dans une rubrique `'topic/subtopic'` : `{temperature: 50}`

Charge utile entrante publiée dans une rubrique `'topic/subtopic-2'` : `{temperature: 50}`

SQL : `"SELECT temperature AS t FROM 'topic/subtopic'"`.

`'topic/subtopic'` est abonné à la règle, de sorte que la charge utile entrante soit transmise à la règle. La charge utile sortante, transmise aux actions de règle, est : `{t: 50}`. La règle n'est abonnée à`'topic/subtopic-2'`, donc la règle n'est pas déclenchée pour le message publié sur`'topic/subtopic-2'`.

Exemple de caractère générique # :

Vous pouvez utiliser le caractère générique « # » (multiniveau) pour correspondre à un ou plusieurs éléments de chemin particuliers :

Charge utile entrante publiée dans une rubrique `'topic/subtopic'` : `{temperature: 50}`.

Charge utile entrante publiée dans une rubrique `'topic/subtopic-2'` : `{temperature: 60}`.

Charge utile entrante publiée dans une rubrique `'topic/subtopic-3/details'` : `{temperature: 70}`.

Charge utile entrante publiée dans une rubrique `'topic-2/subtopic-x'` : `{temperature: 80}`.

SQL : `"SELECT temperature AS t FROM 'topic/#'"`.

La règle est abonnée à n'importe quelle rubrique commençant par`'topic'`, il est donc exécuté trois fois, en envoyant des charges utiles sortantes de`{t: 50}`(pour le sujet/sous-thème),`{t: 60}`(pour le sujet/sous-sujet-2), et`{t: 70}`(pour le sujet/sous-sujet-3/détails) à ses actions. N'étant pas abonnée à `'topic-2/subtopic-x'`, la règle n'est pas déclenchée pour le message `{temperature: 80}`.

Exemple de caractère générique + :

Vous pouvez utiliser le caractère générique « + » (un seul niveau) pour correspondre à tout élément de chemin particulier :

Charge utile entrante publiée dans une rubrique `'topic/subtopic'` : `{temperature: 50}`.

Charge utile entrante publiée dans une rubrique `'topic/subtopic-2'` : `{temperature: 60}`.

Charge utile entrante publiée dans une rubrique `'topic/subtopic-3/details'` : `{temperature: 70}`.

Charge utile entrante publiée dans une rubrique `'topic-2/subtopic-x'` : `{temperature: 80}`.

SQL : `"SELECT temperature AS t FROM 'topic/+'"`.

La règle est abonnée à toutes les rubriques avec deux éléments de chemin où le premier élément est `'topic'`. La règle est exécutée pour les messages envoyés à`'topic/subtopic'`and`'topic/subtopic-2'`, mais pas`'topic/subtopic-3/details'`(il a plus de niveaux que le filtre de rubrique) ou`'topic-2/subtopic-x'`(il ne commence pas par`topic`).

Clause WHERE

La clause WHERE détermine si les actions spécifiées par une règle sont exécutées. Si la clause WHERE évalue sur true, les actions de règle sont exécutées. Sinon, les actions de règle ne sont pas exécutées.

La clause WHERE prend en charge Types de données (p. 485), Operators (p. 489), Functions (p. 495), Literals (p. 539), Instructions Case (p. 539), Extensions JSON (p. 540), Modèles de substitution (p. 541) et Requêtes d'objets imbriqués (p. 543).

Exemple :

Charge utile entrante publiée dans `topic/subtopic` : `{"color":"red", "temperature":40}`.

SQL : `SELECT color AS my_color FROM 'topic/subtopic' WHERE temperature > 50 AND color <> 'red'`.

Dans ce cas, la règle sera déclenchée mais les actions spécifiées par cette règle ne seront pas exécutées. Il n'y aura aucune charge utile sortante.

Vous pouvez utiliser des fonctions et des opérateurs dans une clause WHERE. Toutefois, vous ne pouvez pas faire référence à des alias créés avec le mot-clé AS dans la clause SELECT. (La clause WHERE est évaluée en premier pour déterminer si la clause SELECT doit être évaluée.)

Types de données

Le moteur de règles AWS IoT prend en charge tous les types de données JSON.

Types de données pris en charge

Type	Signification
`Int`	Un. discret`Int`. 34 chiffres maximum.
`Decimal`	Une valeur `Decimal` avec une précision de 34 chiffres, avec une magnitude non nulle de 1E-999 et une magnitude maximale de 9.999… E999. **Note** Certaines fonctions renvoient`Decimal`Avec double précision plutôt qu'avec une précision de 34 chiffres.

Type	Signification
	Avec SQL V2 (2016-03-23), les valeurs numériques qui sont des nombres entiers, tels que`10.0`, sont traitées en tant que`Int`Valeur (`10`) au lieu de l'`Decimal`Valeur (`10.0`). Pour traiter de manière fiable les valeurs numériques de nombre entier en tant que`Decimal`, utilisez SQL V1 (2015-10-08) pour l'instruction de requête de règle.
`Boolean`	`True` ou `False`.
`String`	Une chaîne UTF-8.
`Array`	Une série de valeurs qui ne sont pas nécessairement du même type.
`Object`	Une valeur JSON composée d'une clé et d'une valeur. Les clés doivent être des chaînes. Les valeurs peuvent être de n'importe quel type.
`Null`	`Null` comme défini par JSON. C'est une valeur réelle qui représente l'absence d'une valeur. Vous pouvez créer une valeur `Null` en utilisant le mot-clé `Null` dans votre instruction SQL. Par exemple: `"SELECT NULL AS n FROM 'topic/subtopic'"`

`Undefined`	Ce n'est pas une valeur. Non représenté dans JSON, sauf en omettant la valeur. Par exemple, dans l'objet `{"foo": null}`, la clé « foo » renvoie NULL, mais la clé « bar » renvoie `Undefined`. En interne, le langage SQL traite `Undefined` comme une valeur, mais il ne peut pas être représenté dans JSON, donc quand il est sérialisé au format JSON, les résultats sont `Undefined`.
	`{"foo":null, "bar":undefined}`
	est sérialisé au format JSON comme suit :
	`{"foo":null}`
	De même, `Undefined` est converti en chaîne vide lorsqu'il est sérialisé par lui-même. Les fonctions appelées avec des arguments non valides (par exemple, des types incorrects, un nombre d'arguments incorrect, etc.) renvoient `Undefined`.

Conversions

Le tableau suivant répertorie les résultats lorsqu'une valeur d'un type est convertie dans un autre type (lorsqu'une valeur d'un type incorrect est transmise à une fonction). Par exemple, si la fonction de valeur

absolue « abs » (qui prévoit une valeur `Int` ou `Decimal`) reçoit une valeur `String`, elle tente de convertir la valeur `String` en `Decimal`, en respectant ces règles. Dans ce cas, « abs("-5.123") » est traité comme « abs(-5.123) ».

Note

Il n'y a aucune tentative de conversion en `Array`, `Object`, `Null` ou `Undefined`.

En valeur décimale

Type d'argument	Résultat
`Int`	Un chiffre `Decimal` sans virgule décimale.
`Decimal`	La valeur source.
`Boolean`	`Undefined`. (Vous pouvez explicitement utiliser la fonction cast pour transformer true = 1.0, false = 0.0.)
`String`	Le moteur SQL tente d'analyser la chaîne en tant que `Decimal`. AWS IoT tente d'analyser les chaînes correspondant à l'expression régulière :`^-?\d+(\.\d+)?((?i)E-?\d+)?$`. « 0 », « -1.2 », « 5E-12 » sont des exemples de chaînes qui sont automatiquement converties en valeurs `Decimal`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Null`.
Non défini	`Undefined`.

En Entier

Type d'argument	Résultat
`Int`	La valeur source.
`Decimal`	La valeur source arrondie à la valeur `Int` la plus proche.
`Boolean`	`Undefined`. (Vous pouvez explicitement utiliser la fonction cast pour transformer true = 1.0, false = 0.0.)
`String`	Le moteur SQL tente d'analyser la chaîne en tant que `Decimal`. AWS IoT tente d'analyser les chaînes correspondant à l'expression régulière :`^-?\d+(\.\d+)?((?i)E-?\d+)?$`. « 0 », « -1.2 », « 5E-12 » sont des exemples de chaînes qui sont automatiquement converties en valeurs `Decimal`. AWS IoT tente de convertir la valeur `String` en `Decimal`, puis de tronquer les chiffres après la virgule de la valeur `Decimal` pour obtenir une valeur `Int`.

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

En valeur booléenne

Type d'argument	Résultat
`Int`	`Undefined`. (Vous pouvez explicitement utiliser la méthode`cast`Fonction pour transformer 0 = False, any_nonzero_value = True.)
`Decimal`	`Undefined`. (Vous pouvez explicitement utiliser la fonction cast pour transformer 0 = False, any_nonzero_value = True.)
`Boolean`	La valeur d'origine.
`String`	« true »=True et « false »=False (insensible à la casse). Les autres valeurs de chaînes sont : `Undefined`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

En chaîne

Type d'argument	Résultat
`Int`	Une représentation de chaîne de la valeur `Int` en notation standard.
`Decimal`	Une chaîne représentant la valeur `Decimal`, probablement en notation scientifique.
`Boolean`	« true » ou « false ». Tout en minuscules.
`String`	La valeur d'origine.
Array	Le `Array` sérialisé au format JSON. La chaîne résultante est une liste de valeurs séparées par des virgules, délimitées par des crochets. Une valeur `String` est indiquée entre guillemets. Pas les valeurs `Decimal`, `Int`, `Boolean` et `Null`.
Objet	L'objet sérialisé au format JSON. La chaîne résultante est une liste de paires clé-valeur séparées par des virgules, qui commence et se

Type d'argument	Résultat
	termine par des accolades. Une valeur `String` est indiquée entre guillemets. Pas les valeurs `Decimal`, `Int`, `Boolean` et `Null`.
Null	`Undefined`.
Non défini	Non défini.

Operators

Les opérateurs suivants peuvent être utilisés dans les clauses SELECT et WHERE.

Opérateur AND

Il renvoie une valeur `Boolean`. Il effectue une opération AND logique. Il renvoie la valeur true si les opérandes gauche et droit sont vrais. Sinon, il renvoie la valeur « false ». Des opérandes `Boolean` ou de chaînes « true » ou « false » sensibles à la casse sont requis.

Syntaxe : *`expression`* `AND` *`expression`*.

Opérateur AND

Opérande gauche	Opérande droit	Sortie
`Boolean`	`Boolean`	`Boolean`. Vrai si les deux opérandes sont vrais. Sinon, la valeur renvoyée est Faux.
`String/Boolean`	`String/Boolean`	Si toutes les chaînes sont « true » ou « false » (insensibles à la casse), elles sont converties en valeurs `Boolean` et traitées normalement en tant que valeurs *`boolean`* `AND` *`boolean`*.
Autre valeur	Autre valeur	`Undefined`.

Opérateur OR

Il renvoie une valeur `Boolean`. Il effectue une opération OR logique. Il renvoie la valeur true si les opérandes gauche ou droit sont vrais. Sinon, il renvoie la valeur « false ». Des opérandes `Boolean` ou de chaînes « true » ou « false » sensibles à la casse sont requis.

Syntaxe : *`expression`* `OR` *`expression`*.

Opérateur OR

Opérande gauche	Opérande droit	Sortie
`Boolean`	`Boolean`	`Boolean`. Vrai si l'un des deux opérandes est vrai. Sinon, la valeur renvoyée est Faux. Sinon, la valeur renvoyée est Faux.
`String/Boolean`	`String/Boolean`	Si toutes les chaînes sont « true » ou « false » (insensibles à la casse), elles sont converties en valeurs booléennes et traitées normalement en tant que valeurs *`boolean`* `OR` *`boolean`*.
Autre valeur	Autre valeur	`Undefined`.

Opérateur NOT

Il renvoie une valeur `Boolean`. Il effectue une opération NOT logique. Il renvoie true si l'opérande est faux. Sinon, la valeur renvoyée est true. Un opérande `Boolean` ou un opérande de chaîne « true » ou « false » insensible à la casse est requis.

Syntaxe : `NOT` *`expression`*.

Opérateur NOT

Opérande	Sortie
`Boolean`	`Boolean`. Vrai si l'opérande est Faux. Sinon, la valeur renvoyée est Vrai.
`String`	Si une chaîne est « true » ou « false » (insensible à la casse), elle est convertie dans la valeur booléenne correspondante, et la valeur opposée est renvoyée.
Autre valeur	`Undefined`.

> opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si l'opérande gauche est supérieur à l'opérande droit. Les deux opérandes sont convertis en valeur `Decimal`, puis comparés.

Syntaxe : *`expression`* `>` *`expression`*.

> opérateur

Opérande gauche	Opérande droit	Sortie
`Int/Decimal`	`Int/Decimal`	`Boolean`. Vrai si l'opérande gauche est supérieur à l'opérande droit. Sinon, la valeur renvoyée est Faux.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes peuvent être converties en valeurs `Decimal`, puis en valeurs `Boolean`. Il renvoie la valeur true si l'opérande gauche est supérieur à l'opérande droit. Sinon, la valeur renvoyée est Faux.
Autre valeur	`Undefined`.	`Undefined`.

>= opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si l'opérande gauche est supérieur ou égal à l'opérande droit. Les deux opérandes sont convertis en valeur `Decimal`, puis comparés.

Syntaxe : *`expression`* `>=` *`expression`*.

>= opérateur

Opérande gauche	Opérande droit	Sortie
`Int/Decimal`	`Int/Decimal`	`Boolean`. Vrai si l'opérande gauche est supérieur ou égal à l'opérande droit. Sinon, la valeur renvoyée est Faux.

Opérande gauche	Opérande droit	Sortie
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes peuvent être converties en valeurs `Decimal`, puis en valeurs `Boolean`. Il renvoie la valeur true si l'opérande gauche est supérieur ou égal à l'opérande droit. Sinon, la valeur renvoyée est Faux.
Autre valeur	`Undefined`.	`Undefined`.

< opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si l'opérande gauche est inférieur à l'opérande droit. Les deux opérandes sont convertis en valeur `Decimal`, puis comparés.

Syntaxe : *`expression`* `<` *`expression`*.

< opérateur

Opérande gauche	Opérande droit	Sortie
`Int/Decimal`	`Int/Decimal`	`Boolean`. Vrai si l'opérande gauche est inférieur à l'opérande droit. Sinon, la valeur renvoyée est Faux.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes peuvent être converties en valeurs `Decimal`, puis en valeurs `Boolean`. Il renvoie la valeur true si l'opérande gauche est inférieur à l'opérande droit. Sinon, la valeur renvoyée est Faux.
Autre valeur	`Undefined`	`Undefined`

<= opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si l'opérande gauche est inférieur ou égal à l'opérande droit. Les deux opérandes sont convertis en valeur `Decimal`, puis comparés.

Syntaxe: *`expression`* `<=` *`expression`*.

<= opérateur

Opérande gauche	Opérande droit	Sortie
`Int/Decimal`	`Int/Decimal`	`Boolean`. Vrai si l'opérande gauche est inférieur ou égal à l'opérande droit. Sinon, la valeur renvoyée est Faux.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes peuvent être converties en valeurs `Decimal`, puis en valeurs `Boolean`. Il renvoie la valeur true si l'opérande gauche est inférieur ou égal à l'opérande droit. Sinon, la valeur renvoyée est Faux.
Autre valeur	`Undefined`	`Undefined`

<> opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si les opérandes gauche et droit ne sont pas égaux. Sinon, la valeur renvoyée est false.

Syntaxe : *`expression`* `<>` *`expression`*.

<> opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Vrai si l'opérande gauche est différent de l'opérande droit. Sinon, la valeur renvoyée est Faux.
`Decimal`	`Decimal`	Vrai si l'opérande gauche est différent de l'opérande droit. Sinon, la valeur renvoyée est Faux. `Int` est convertie en valeur `Decimal` avant d'être comparée.
`String`	`String`	Vrai si l'opérande gauche est différent de l'opérande droit. Sinon, la valeur renvoyée est Faux.
Array	Array	Vrai si les éléments de chaque opérande sont différents et ne sont pas dans le même ordre. Sinon, la valeur renvoyée est Faux
Objet	Objet	Vrai si les clés et les valeurs de chaque opérande sont différentes. Sinon, la valeur renvoyée est Faux. L'ordre des clés/valeurs n'est pas important.
Null	Null	Faux.
N'importe quelle valeur	`Undefined`	Non défini.
`Undefined`	N'importe quelle valeur	Non défini.
Type non compatible	Type non compatible	Vrai.

= opérateur

Il renvoie une valeur `Boolean`. Il renvoie la valeur true si les opérandes gauche et droit sont égaux. Sinon, la valeur renvoyée est false.

Syntaxe : *`expression`* `=` *`expression`*.

= opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Vrai si l'opérande gauche est identique à l'opérande droit. Sinon, la valeur renvoyée est Faux.
`Decimal`	`Decimal`	Vrai si l'opérande gauche est identique à l'opérande droit. Sinon, la valeur renvoyée est Faux. `Int` est convertie en valeur `Decimal` avant d'être comparée.
`String`	`String`	Vrai si l'opérande gauche est identique à l'opérande droit. Sinon, la valeur renvoyée est Faux.
Array	Array	Vrai si les éléments de chaque opérande sont égaux et sont dans le même ordre. Sinon, la valeur renvoyée est Faux.

Opérande gauche	Opérande droit	Sortie
Objet	Objet	Vrai si les clés et valeurs de chaque opérande sont identiques. Sinon, la valeur renvoyée est Faux. L'ordre des clés/valeurs n'est pas important.
N'importe quelle valeur	`Undefined`	`Undefined`.
`Undefined`	N'importe quelle valeur	`Undefined`.
Type non compatible	Type non compatible	Faux.

+ opérateur

Le signe « + » est un opérateur surchargé. Il peut être utilisé pour l'addition ou la concaténation de chaines.

Syntaxe : *`expression`* `+` *`expression`*.

+ opérateur

Opérande gauche	Opérande droit	Sortie
`String`	N'importe quelle valeur	Il convertit l'opérande droit en chaîne et l'ajoute à la fin de l'opérande gauche.
N'importe quelle valeur	`String`	Il convertit l'opérande gauche en chaîne et ajoute l'opérande droit à la fin de l'opérande gauche converti.
`Int`	`Int`	Valeur `Int`. Il ajoute les opérandes ensemble.
`Int/Decimal`	`Int/Decimal`	Valeur `Decimal`. Il ajoute les opérandes ensemble.
Autre valeur	Autre valeur	`Undefined`.

- opérateur

Il soustrait l'opérande droit de l'opérande gauche.

Syntaxe : *`expression`* `-` *`expression`*.

- opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Valeur `Int`. Il soustrait l'opérande droit de l'opérande gauche.
`Int/Decimal`	`Int/Decimal`	Valeur `Decimal`. Il soustrait l'opérande droit de l'opérande gauche.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes sont correctement converties en valeurs décimales, une valeur `Decimal` est renvoyée. Il soustrait l'opérande droit de l'opérande gauche. Sinon, la valeur renvoyée est `Undefined`.

Opérande gauche	Opérande droit	Sortie
Autre valeur	Autre valeur	`Undefined`.
Autre valeur	Autre valeur	`Undefined`.

* opérateur

Il multiplie l'opérande gauche par l'opérande droit.

Syntaxe : *`expression`* `*` *`expression`*.

* opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Valeur `Int`. Il multiplie l'opérande gauche par l'opérande droit.
`Int/Decimal`	`Int/Decimal`	Valeur `Decimal`. Il multiplie l'opérande gauche par l'opérande droit.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes sont correctement converties en valeurs décimales, une valeur `Decimal` est renvoyée. Il multiplie l'opérande gauche par l'opérande droit. Sinon, la valeur renvoyée est `Undefined`.
Autre valeur	Autre valeur	`Undefined`.

/ opérateur

Il divise l'opérande gauche par l'opérande droit.

Syntaxe : *`expression`* `/` *`expression`*.

/ opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Valeur `Int`. Il divise l'opérande gauche par l'opérande droit.
`Int/Decimal`	`Int/Decimal`	Valeur `Decimal`. Il divise l'opérande gauche par l'opérande droit.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes sont correctement converties en valeurs décimales, une valeur `Decimal` est renvoyée. Il divise l'opérande gauche par l'opérande droit. Sinon, la valeur renvoyée est `Undefined`.
Autre valeur	Autre valeur	`Undefined`.

% opérateur

Il renvoie le reste résultant de la division de l'opérande gauche par l'opérande droit.

Syntaxe : *`expression`* `%` *`expression`*.

% opérateur

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	Valeur `Int`. Il renvoie le reste résultant de la division de l'opérande gauche par l'opérande droit.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes sont correctement converties en valeurs décimales, une valeur `Decimal` est renvoyée. Il renvoie le reste résultant de la division de l'opérande gauche par l'opérande droit. Sinon la valeur `Undefined` est renvoyée.
Autre valeur	Autre valeur	`Undefined`.

Functions

Vous pouvez utiliser les fonctions intégrées suivantes dans les clauses SELECT ou WHERE de vos expressions SQL.

abs(Decimal)

Il renvoie la valeur absolue d'un nombre. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Par exemple, `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`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

accountid()

Renvoie l'ID du compte qui possède la règle comme une valeur `String`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`accountid()` = "123456789012"

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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `acos(0)` = 1,5707963267948966

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le cosinus inverse de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined`.
`Decimal`	`Decimal` (avec double précision), le cosinus inverse de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined`.
`Boolean`	`Undefined`.
`String`	`Decimal`, 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`.
Array	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `asin(0)` = 0,0

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

Type d'argument	Résultat
Array	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `atan(0)` = 0,0

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), la tangente inverse de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined`.
`Decimal`	`Decimal` (avec double précision), la tangente inverse de l'argument. Des résultats imaginaires sont retournés sous la forme `Undefined`.
`Boolean`	`Undefined`.
`String`	`Decimal`, 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`.
Array	`Undefined`.
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 sans le sens contraire des aiguilles d'une montre (moitié supérieure du plan, y > 0) et négatif pour les angles dans le sens des aiguilles d'une montre (moitié inférieure du plan, y < 0). Les arguments `Decimal` sont arrondis pour une meilleure précision avant l'application de la fonction. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `atan2(1, 0)` = 1,5707963267948966

Type d'argument	Type d'argument	Résultat
`Int/Decimal`	`Int/Decimal`	`Decimal` (avec doubl et le point (x, y) spéci
`Int/Decimal/String`	`Int/Decimal/String`	`Decimal`, la tangente chaîne ne peut pas êt `Undefined`.
Autre valeur	Autre valeur	`Undefined`.

aws_lambda(functionArn, inputJson)

Appelle la fonction Lambda spécifiée en passant`inputJson`À la fonction Lambda et renvoie les données JSON générées par la fonction Lambda.

Arguments

Argument	Description
`functionArn`	L'ARN de la fonction Lambda à appeler. La fonction Lambda doit renvoyer des données JSON.
`inputJson`	Données JSON en entrée transmises à la fonction Lambda. Pour transmettre des requêtes et littéraux d'objet imbriqués, vous devez utiliser SQL 2016-03-23.

Vous devez accorderAWS IoT `lambda:InvokeFunction`Autorisations pour appeler la fonction Lambda spécifiée. L'exemple suivant montre comment accorder l'autorisation `lambda:InvokeFunction` à l'aide de l'AWS CLI :

```
aws lambda add-permission --function-name "function_name"
--region "region"
--principal iot.amazonaws.com
--source-arn arn:aws:iot:us-east-1:account_id:rule/rule_name
--source-account "account_id"
--statement-id "unique_id"
--action "lambda:InvokeFunction"
```

Les arguments de la commande add-permission sont les suivants :

--function-name

Nom de la fonction Lambda. Vous ajoutez une nouvelle autorisation pour mettre à jour la stratégie de ressources de la fonction.

--région

La . Région AWS de votre compte.

--principal

Mandataire qui obtient l'autorisation. Cela devrait être`iot.amazonaws.com`pour permettreAWS IoTautorisation d'appeler une fonction Lambda.

--source-arn

ARN de la règle. Vous pouvez utiliser la stratégie`get-topic-rule` AWS CLIpour obtenir l'ARN d'une règle.

--source-account

La . Compte AWS où la règle est définie.

--statement-id

Identifiant unique de l'instruction.

--action

Action Lambda que vous souhaitez autoriser dans cette instruction. Pour permettreAWS IoTPour appeler une fonction Lambda, spécifiez`lambda:InvokeFunction`.

Important

Si vous ajoutez une autorisation pour unAWS IoTprincipal sans fournir l'ARN source, tout Compte AWS qui crée une règle avec votre action Lambda peut déclencher des règles pour appeler votre fonction Lambda à partir d'AWS IoT. Pour de plus amples informations, veuillez consulterModèle d'autorisation Lambda.

Soit une charge utile de message JSON comme suit :

```
{
    "attribute1": 21,
    "attribute2": "value"
}
```

La `.aws_lambda`peut être utilisée pour appeler la fonction Lambda, comme suit.

```
SELECT
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function",
 {"payload":attribute1}) as output FROM 'topic-filter'
```

Si vous souhaitez transmettre l'intégralité de la charge utile des messages MQTT, vous pouvez spécifier la charge utile JSON en utilisant « * », comme l'exemple suivant.

```
SELECT
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function", *) as output
 FROM 'topic-filter'
```

`payload.inner.element`sélectionne les données à partir d'un message publié sur le sujet « sujet/sous-sujet ».

`some.value`sélectionne les données à partir de la sortie générée par la fonction Lambda.

Note

Le moteur de règles limite la durée d'exécution des fonctions Lambda. Les appels de fonction Lambda provenant de règles doivent être terminés en moins de 2 000 millisecondes.

bitand(Int, Int)

Il effectue une opération AND au niveau du bit sur des représentations binaires des deux arguments `Int`(-convertis). Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `bitand(13, 5)` = 5

Type d'argument	Type d'argument	Résultat
`Int`	`Int`	`Int`, une opération Al arguments.
`Int/Decimal`	`Int/Decimal`	`Int`, une opération Al arguments. Tous les r valeur `Int` inférieure ne peut pas être conv `Undefined`.
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, une opération Al arguments. Toutes les décimales et arrondie plus proche. Si la con `Undefined`.
Autre valeur	Autre valeur	`Undefined`.

bitor(Int, Int)

Il effectue une opération OR au niveau du bit des représentations binaires des deux arguments. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `bitor(8, 5)` = 13

Type d'argument	Type d'argument	Résultat
`Int`	`Int`	`Int`, une opération O arguments.
`Int/Decimal`	`Int/Decimal`	`Int`, une opération O arguments. Tous les r la valeur `Int` inférieur échoue, le résultat est
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, une opération O arguments. Toutes les décimales et arrondie plus proche. Si la con `Undefined`.
Autre valeur	Autre valeur	`Undefined`.

bitxor(Int, Int)

Il effectue une opération XOR au niveau du bit sur des représentations binaires des deux arguments `Int`(-convertis). Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple:`bitor(13, 5)` = 8

Type d'argument	Type d'argument	Résultat
`Int`	`Int`	`Int`, une opération X arguments.

Type d'argument	Type d'argument	Résultat
`Int/Decimal`	`Int/Decimal`	`Int`, une opération X arguments. Les nomb `Int` inférieure la plus
`Int/Decimal/String`	`Int/Decimal/String`	`Int`, un XOR au nive Les chaînes sont con la valeur `Int` inférieu échoue, le résultat es
Autre valeur	Autre valeur	`Undefined`.

bitnot(Int)

Il effectue une opération NOT au niveau du bit sur des représentations binaires de l'argument `Int`(-converti). Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `bitnot(13)` = 2

Type d'argument	Résultat
`Int`	`Int`, une opération NOT au niveau du bit de l'argument.
`Decimal`	`Int`, une opération NOT au niveau du bit de l'argument. La valeur `Decimal` est arrondie à la valeur `Int` inférieure la plus proche.
`String`	`Int`, une opération NOT au niveau du bit de l'argument. Les chaînes sont converties en valeurs décimales et arrondies à la valeur `Int` inférieure la plus proche. Si une conversion échoue, le résultat est `Undefined`.
Autre valeur	Autre valeur.

cast()

Convertit une valeur d'un type de données en un autre. La conversion se comporte principalement comme les conversions standard, avec en outre la capacité de convertir des chiffres vers/depuis des valeurs booléennes. Si AWS IoT ne peut pas déterminer comment convertir un type vers un autre, le résultat est `Undefined`. Prise en charge par SQL 2015-10-08 et versions ultérieures. Format : cast(*`valeur`* as *`type`*).

Exemple :

```
cast(true as Int)  = 1
```

Les mots-clés suivants peuvent apparaître après « as » lors de l'appel de `cast` :

Pour SQL versions 2015-10-08 et 2016-03-23

Mot clé	Résultat
`String`	Il convertit une valeur en `String`.
Nvarchar	Il convertit une valeur en `String`.

Mot clé	Résultat
Text	Il convertit une valeur en `String`.
Ntext	Il convertit une valeur en `String`.
varchar	Il convertit une valeur en `String`.
`Int`	Il convertit une valeur en `Int`.
Integer	Il convertit une valeur en `Int`.
Double	Il convertit une valeur en`Decimal`(avec double précision).

En outre, pour SQL version 2016-03-23

Mot clé	Résultat
`Decimal`	Il convertit une valeur en `Decimal`.
Booléen	Il convertit une valeur en `Boolean`.
`Boolean`	Il convertit une valeur en `Boolean`.

Règles de conversion de types :

Conversion en décimal

Type d'argument	Résultat
`Int`	Un chiffre `Decimal` sans virgule décimale.
`Decimal`	La valeur source. **Note** Avec SQL V2 (2016-03-23), les valeurs numériques qui sont des nombres entiers, tels que`10.0`, renvoie une`Int`Valeur (`10`) au lieu de l'`Decimal`Valeur (`10.0`). Pour lancer de manière fiable des valeurs numériques de nombre entier en tant que`Decimal`, utilisez SQL V1 (2015-10-08) pour l'instruction de requête de règle.
`Boolean`	true = 1.0, false = 0.0.
`String`	Tente d'analyser la chaîne en tant que `Decimal`. AWS IoT tente d'analyser les chaînes correspondant à l'expression regex : ^-?\d+(\.\d+)?((?i)E-?\d+)?$. « 0 », « -1.2 », « 5E-12 » sont des exemples de chaînes qui sont automatiquement converties en valeurs décimales.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.

Type d'argument	Résultat
Non défini	`Undefined`.

Conversion en entier

Type d'argument	Résultat
`Int`	La valeur source.
`Decimal`	La valeur source arrondie à la valeur `Int` inférieure la plus proche.
`Boolean`	true = 1.0, false = 0.0.
`String`	Tente d'analyser la chaîne en tant que `Decimal`. AWS IoT tente d'analyser les chaînes correspondant à l'expression regex : ^-?\d+(\.\d+)?((?i)E-?\d+)?$. « 0 », « -1.2 », « 5E-12 » sont des exemples de chaînes qui sont automatiquement converties en valeurs décimales. AWS IoT tente de convertir la chaîne en valeur `Decimal`, puis de l'arrondir à la valeur `Int` inférieure la plus proche.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

Conversion en valeur **`Boolean`**

Type d'argument	Résultat
`Int`	0 = False, any_nonzero_value = True.
`Decimal`	0 = False, any_nonzero_value = True.
`Boolean`	La valeur source.
`String`	« true »=True et « false »=False (insensible à la casse). Autres valeurs de chaînes = `Undefined`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

Conversion en chaîne

Type d'argument	Résultat
`Int`	Une représentation de chaîne de la valeur `Int` en notation standard.

Type d'argument	Résultat
`Decimal`	Une chaîne représentant la valeur `Decimal`, probablement en notation scientifique.
`Boolean`	« true » ou « false », tout en minuscules.
`String`	« true »=True et « false »=False (insensible à la casse). Autres valeurs de chaînes = `Undefined`.
Array	Le tableau sérialisé au format JSON. La chaîne résultante consiste en une liste séparée par des virgules et délimitée par des crochets. `String` est entre guillemets, à l'inverse de `Decimal`, `Int` et `Boolean`.
Objet	L'objet sérialisé au format JSON. La chaîne JSON est une liste de paires clé-valeur séparées par des virgules, délimitées par des accolades. `String` est entre guillemets, à l'inverse de `Decimal`, `Int`, `Boolean` et `Null`.
Null	`Undefined`.
Non défini	`Undefined`.

ceil(Decimal)

Arrondit la valeur `Decimal` donnée à la valeur `Int` supérieure la plus proche. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`ceil(1.2)` = 2

`ceil(-1.2)` = -1

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

chr(String)

Renvoie le caractère ASCII qui correspond à l'argument `Int` donné. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`chr(65)` = "A".

`chr(49)` = "1".

Type d'argument	Résultat
`Int`	Le caractère correspondant à la valeur ASCII spécifiée. Si l'argument n'est pas une valeur ASCII valide, le résultat est `Undefined`.
`Decimal`	Le caractère correspondant à la valeur ASCII spécifiée. L'argument `Decimal` est arrondi à la valeur `Int` inférieure la plus proche. Si l'argument n'est pas une valeur ASCII valide, le résultat est `Undefined`.
`Boolean`	`Undefined`.
`String`	Si la valeur `String` peut être convertie en valeur `Decimal`, elle est arrondie à la valeur `Int` inférieure la plus proche. Si l'argument n'est pas une valeur ASCII valide, le résultat est `Undefined`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Autre valeur	`Undefined`.

clientid()

Retourne l'ID du client MQTT en envoyant le message, ou une valeur `n/a` si le message n'a pas été pas envoyé via MQTT. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`clientid()` = "123456789012"

concat()

Concatène des tableaux ou des chaînes. Cette fonction accepte n'importe quel nombre d'arguments et renvoie une valeur `String` ou `Array`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`concat()` = `Undefined`.

`concat(1)` = "1".

`concat([1, 2, 3], 4)` = [1, 2, 3, 4].

`concat([1, 2, 3], "hello")` = [1, 2, 3, "bonjour"]

`concat("con", "cat")` = "concat"

`concat(1, "hello")` = "bonjour1"

`concat("he","is","man")` = "heisman"

`concat([1, 2, 3], "hello", [4, 5, 6])` = [1, 2, 3, "bonjour", 4, 5, 6]

Nombre d'arguments	Résultat
0	`Undefined`.
1	L'argument est renvoyé non modifié.
2+	Si un argument est une valeur `Array`, le résultat est un seul tableau contenant l'ensemble des arguments. Si aucun argument n'est une valeur de tableau, et qu'un argument au moins est une valeur `String`, le résultat est la concaténation des représentations de `String` de tous les arguments. Des arguments sont convertis en chaînes à l'aide des conversions standard répertoriées plus haut. .

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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`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`.
Array	`Undefined`.
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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

décoder (valeur, DecodingScheme)

Utilisation de l'`decode`pour décoder une valeur encodée. Si la chaîne décodée est un document JSON, un objet adressable est renvoyé. Sinon, la chaîne décodée est renvoyée comme chaîne. La fonction renvoie NULL si la chaîne ne peut pas être décodée.

Pris en charge par SQL 2016-03-23 et versions ultérieures.

valeur

Une valeur de chaîne ou une des expressions valides, telles que définies dans laRéférence SQL AWS IoT (p. 482), qui renvoient une chaîne.

Schéma de décodage

Chaîne littérale qui représente le schéma utilisé pour décoder la valeur. Actuellement, seul `'base64'` est pris en charge.

Example

Dans cet exemple, la charge utile du message inclut une valeur encodée.

```
{
    encoded_temp: "eyAidGVtcGVyYXR1cmUiOiAzMyB9Cg=="
}
```

La .`decode`dans cette instruction SQL, décode la valeur dans la charge utile du message.

```
SELECT decode(encoded_temp,"base64").temperature AS temp from 'topic/subtopic'
```

Décodage des`encoded_temp`entraîne le document JSON valide suivant, ce qui permet à l'instruction SELECT de lire la valeur de température.

```
{ "temperature": 33 }
```

Le résultat de l'instruction SELECT dans cet exemple est illustré ici.

```
{ "temp": 33 }
```

Si la valeur décodée n'était pas un document JSON valide, la valeur décodée serait renvoyée sous forme de chaîne.

encode(value, encodingScheme)

Utilisez la fonction `encode` pour encoder la charge utile, qui peut être constituée de données non-JSON, dans sa représentation de chaîne basée sur le schéma d'encodage. Pris en charge par SQL 2016-03-23 et versions ultérieures.

valeur

Une des expressions valides, telles que définies dans la Référence SQL AWS IoT (p. 482). Vous pouvez spécifier * pour encoder la charge utile dans son ensemble, qu'elle soit ou non au format JSON. Si vous fournissez une expression, le résultat de l'évaluation est converti en une chaîne avant d'être codé.

encodingScheme

Chaîne littérale qui représente le schéma de codage à utiliser. Actuellement, seul `'base64'` est pris en charge.

endswith(String, String)

Renvoie une valeur `Boolean` indiquant si le premier argument `String` se termine par le deuxième argument `String`. Si l'un des arguments est `Null` ou `Undefined`, le résultat a la valeur `Undefined`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Par exemple : `endswith("cat","at")` = true.

Type d'argument 1	Type d'argument 2	Résultat
`String`	`String`	Vrai si le premier argu argument. Sinon, la v
Autre valeur	Autre valeur	Les deux arguments s règles de conversion se termine dans le se renvoyée est Faux. Si `Undefined`, le résult

exp(Decimal)

Renvoie la valeur augmentée vers l'argument `Decimal`. Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple : `exp(1)` = e.

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), argument puissance e.
`Decimal`	`Decimal` (avec double précision), argument puissance e.
`String`	`Decimal` (avec double précision), argument puissance e. Si la valeur `String` ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined`.
Autre valeur	`Undefined`.

floor(Decimal)

Arrondit la valeur `Decimal` donnée à la valeur `Int` inférieure la plus proche. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`floor(1.2)` = 1

`floor(-1.2)` = -2

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

get

Extrait une valeur à partir d'un type de collection (tableau, chaîne, objet). Aucune conversion n'est appliquée au premier argument. Une conversion s'applique comme documenté dans le tableau au deuxième argument. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`get(["a", "b", "c"], 1)` = "b"

`get({"a":"b"}, "a")` = "b"

`get("abc", 1)` = "b"

Type d'argument 1	Type d'argument 2	Résultat
Array	Tout type (converti valeur `Int`)	L'élément à l'index de fourni par le deuxième

Type d'argument 1	Type d'argument 2	Résultat
		conversion échoue, le est en dehors des lim array.length), le résult
Chaîne	Tout type (converti valeur `Int`)	Le caractère est à l'in fournie par le deuxièm conversion échoue, le est en dehors des lim string.length), le résul
Objet	`String` (aucune conversion appliquée)	La valeur stockée dar correspondant à la clé argument.
Autre valeur	N'importe quelle valeur	`Undefined`.

get_dynamodb(tableName, partitionKeyName, partitionKeyValue, sortKeyName, sortKeyValue, roleArn)

Récupère des données d'une table DynamoDB.`get_dynamodb()`vous permet d'interroger une table DynamoDB pendant l'évaluation d'une règle. Vous pouvez filtrer ou augmenter les charges utiles des messages à l'aide des données extraites de DynamoDB. Pris en charge par SQL 2016-03-23 et versions ultérieures.

`get_dynamodb()` accepte les paramètres suivants :

tableName

Nom de la table DynamoDB à interroger.

partitionKeyName

Nom de la clé de partition. Pour de plus amples informations, veuillez consulterClés DynamoDB.

partitionKeyValue

Valeur de la clé de partition utilisée pour identifier un enregistrement. Pour de plus amples informations, veuillez consulterClés DynamoDB.

sortKeyName

(Facultatif) Nom de la clé de tri. Ce paramètre n'est requis que si la table DynamoDB interrogée utilise une clé composite. Pour de plus amples informations, veuillez consulterClés DynamoDB.

sortKeyValue

(Facultatif) Valeur de la clé de tri. Ce paramètre n'est requis que si la table DynamoDB interrogée utilise une clé composite. Pour de plus amples informations, veuillez consulterClés DynamoDB.

roleArn

ARN d'un rôle IAM qui accorde l'accès à la table DynamoDB. Le moteur de règles assume ce rôle pour accéder à la table DynamoDB en votre nom. Évitez d'utiliser un rôle trop permissif. Accordez au rôle uniquement les autorisations requises par la règle. Voici un exemple de stratégie qui accorde l'accès à une table DynamoDB.

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

```
        {
            "Effect": "Allow",
            "Action": "dynamodb:GetItem",
            "Resource": "arn:aws:dynamodb:aws-region:account-id:table/table-name"
        }
    ]
}}
```

Comme exemple de la façon dont vous pouvez utiliserget_dynamodb(), supposons que vous disposez d'une table DynamoDB qui contient l'ID d'appareil et les informations d'emplacement de tous vos appareils connectés àAWS IoT. L'instruction SELECT suivante utilise la fonction `get_dynamodb()` pour récupérer l'emplacement de l'ID d'appareil spécifié :

```
SELECT *, get_dynamodb("InServiceDevices", "deviceId", id,
"arn:aws:iam::12345678910:role/getdynamo").location AS location FROM 'some/
topic'
```

Note

- Vous pouvez appeler `get_dynamodb()` une fois au maximum par instruction SQL. L'appel de `get_dynamodb()` plusieurs fois dans une même instruction SQL entraîne la fin de la règle sans invoquer aucune action.
- Si `get_dynamodb()` renvoie plus de 8 Ko de données, l'action de la règle ne peut pas être invoquée.

get_secret (secretId, SecretType, key, roleArn)

Récupère la valeur du`SecretString`ou`SecretBinary`de la version actuelle d'un secret dansAWS Secrets Manager. Pour de plus amples informations sur la création et la gestion des secrets, veuillez consulterCreateSecret,UpdateSecret, etPutSecretValue.

`get_secret()` accepte les paramètres suivants :

secretId

Chaîne : Amazon Resource Name (ARN) ou le nom convivial du secret à récupérer.

SecretType

Chaîne : Le type secret. Valeurs valides : `SecretString` | `SecretBinary`.

SecretString

- Pour les secrets que vous créez en tant qu'objets JSON à l'aide des API, la propriétéAWS CLI, ou leAWS Secrets ManagerConsole :
  - Si vous spécifiez une valeur pour la valeur`key`, cette fonction renvoie la valeur de la clé spécifiée.
  - Si vous ne spécifiez aucune valeur pour la valeur`key`, cette fonction renvoie l'ensemble de l'objet JSON.
- Pour les secrets que vous créez en tant qu'objets non-JSON à l'aide des API ou duAWS CLI :
  - Si vous spécifiez une valeur pour la valeur`key`, cette fonction échoue avec une exception.
  - Si vous ne spécifiez aucune valeur pour la valeur`key`, cette fonction renvoie le contenu du secret.

SecretBinary

- Si vous spécifiez une valeur pour la valeur`key`, cette fonction échoue avec une exception.
- Si vous ne spécifiez aucune valeur pour la valeur`key`, cette fonction renvoie la valeur secrète sous forme de chaîne UTF-8 encodée en base64.

key

(Facultatif) Chaîne : Le nom de la clé à l'intérieur d'un objet JSON stocké dans le`SecretString`champ d'un secret. Utilisez cette valeur lorsque vous souhaitez récupérer uniquement la valeur d'une clé stockée dans un secret au lieu de l'objet JSON entier.

Si vous spécifiez une valeur pour ce paramètre et que le secret ne contient pas d'objet JSON dans son`SecretString`, cette fonction échoue avec une exception.

roleArn

Chaîne : Un ARN de rôle avec`secretsmanager:GetSecretValue`and`secretsmanager:DescribeSecret`Autorisations.

Note

Cette fonction renvoie toujours la version actuelle du secret (la version avec l'option`AWSCURRENT`Balise). La .AWS IoTLe moteur de règles met en cache chaque secret pendant 15 minutes. Par conséquent, le moteur de règles peut prendre jusqu'à 15 minutes pour mettre à jour un secret. Cela signifie que si vous récupérez un secret jusqu'à 15 minutes après une mise à jour avecAWS Secrets Manager, cette fonction peut renvoyer l'ancienne version. Cette fonction n'est pas mesurée et est libre d'utiliser, maisAWS Secrets ManagerDes frais supplémentaires seront facturés. En raison du mécanisme de mise en cache secret, le moteur de règles appelle occasionnellementAWS Secrets Manager. Étant donné que le moteur de règles est un service entièrement distribué, vous pouvez voir plusieurs appels d'API Secrets Manager depuis le moteur de règles au cours de la fenêtre de mise en cache de 15 minutes.

Exemples :

Vous pouvez utiliser la stratégie`get_secret`dans un en-tête d'authentification dans une action de règle HTTPS, comme dans l'exemple d'authentification par clé API suivant.

```
"API_KEY": "${get_secret('API_KEY', 'SecretString', 'API_KEY_VALUE',
 'arn:aws:iam::12345678910:role/getsecret')}"
```

Pour de plus amples informations sur l'action de la règle HTTPS, veuillez consulterthe section called “HTTPS” (p. 418).

get_thing_shadow(thingName, shadowName, roleARN)

Renvoie le shadow spécifié de l'objet spécifié. Pris en charge par SQL 2016-03-23 et versions ultérieures.

thingName

Chaîne : Nom de la chose dont vous voulez récupérer l'ombre.

shadowName

(Facultatif) Chaîne : Nom du shadow. Ce paramètre est requis uniquement quand vous référencez des shadows nommés.

roleArn

Chaîne : Un ARN de rôle avec`iot:GetThingShadow`Autorisation.

Exemples :

Lorsqu'elle est utilisée avec un shadow nommé, fournissez le paramètre `shadowName`.

```
SELECT * from 'topic/subtopic'
WHERE
    get_thing_shadow("MyThing","MyThingShadow","arn:aws:iam::123456789012:role/
AllowsThingShadowAccess")
    .state.reported.alarm = 'ON'
```

Lorsqu'elle est utilisée avec un shadow non nommé, omettez le paramètre `shadowName`.

```
SELECT * from 'topic/subtopic'
WHERE
    get_thing_shadow("MyThing","arn:aws:iam::123456789012:role/AllowsThingShadowAccess")
    .state.reported.alarm = 'ON'
```

Fonctions de hachage

AWS IoT fournit les fonctions de hachage suivantes :

- md2
- md5
- sha1
- sha224
- sha256
- sha384
- sha512

Toutes les fonctions de hachage prévoit un argument de type chaîne. Le résultat est la valeur hachée de cette chaîne. Les conversions de chaîne standard s'appliquent aux arguments non-chaîne. Toutes les fonctions de hachage sont prises en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`md2("hello")` = "a9046c73e00331af68917d3804f70655"

`md5("hello")` = "5d41402abc4b2a76b9719d911017c592"

indexof(String, String)

Renvoie le premier index (de base 0) du deuxième argument comme une sous-chaîne dans le premier argument. Les deux arguments doivent être des chaînes. Les arguments qui ne sont pas des chaînes sont soumis aux règles de conversion de chaînes standard. Cette fonction ne s'applique pas aux tableaux, uniquement aux chaînes. Pris en charge par SQL 2016-03-23 et versions ultérieures.

Exemples :

`indexof("abcd", "bc")` = 1

isNull()

Retourne la valeur true si la valeur de l'argument est `Null`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`isNull(5)` = false.

`isNull(Null)` = vrai.

Type d'argument	Résultat
`Int`	false
`Decimal`	false
`Boolean`	false
`String`	false
`Array`	false
`Object`	false
`Null`	vrai
`Undefined`	false

isUndefined()

Retourne la valeur true si l'argument est `Undefined`. Pris en charge par SQL 2016-03-23 et versions ultérieures.

Exemples :

`isUndefined(5)` = false.

`isUndefined(floor([1,2,3]))` = vrai.

Type d'argument	Résultat
`Int`	false
`Decimal`	false
`Boolean`	false
`String`	false
`Array`	false
`Object`	false
`Null`	false
`Undefined`	true

length(String)

Renvoie le nombre de caractères dans la chaîne fournie. Les règles de conversion standard s'appliquent aux arguments non-`String`. Pris en charge par SQL 2016-03-23 et versions ultérieures.

Exemples :

`length("hi")` = 2

`length(false)` = 5

ln(Decimal)

Renvoie le logarithme naturel de l'argument Les arguments `Decimal` sont arrondis pour une meilleure prévision avant l'application de la fonction. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `ln(e)` = 1.

Type d'argument	Résultat
`Int`	`Decimal` (avec double précision), le logarithme naturel de l'argument.
`Decimal`	`Decimal` (avec double précision), le logarithme naturel de l'argument.
`Boolean`	`Undefined`.
`String`	`Decimal` (avec double précision), le logarithme naturel de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined`.
Array	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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` ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.

Type d'argument	Résultat
Non défini	`Undefined`.

lower(String)

Renvoie la version en minuscules de la valeur de `String` donnée. Les arguments non-chaîne sont convertis en chaînes à l'aide des règles de conversion standard. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`lower("HELLO")` = "bonjour".

`lower(["HELLO"])` = "[\"bonjour\"]".

lpad(String, Int)

Renvoie l'argument `String`, complété à gauche par le nombre d'espaces spécifié par le deuxième argument. L'argument `Int` doit être compris entre 0 et 1000. Si la valeur fournie se situe en dehors de cette plage valide, l'argument est défini sur la valeur valide la plus proche (0 ou 1 000). Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`lpad("hello", 2)` = `" hello"`.

`lpad(1, 3)` = `" 1"`

Type d'argument 1	Type d'argument 2	Résultat
`String`	`Int`	`String`, l'argument S un nombre d'espaces
`String`	`Decimal`	L'argument `Decimal` la plus proche, et l'arg gauche par le nombre
`String`	`String`	Le deuxième argumen qui est arrondie à la v l'argument `String` es d'espaces spécifié. Si être converti en une v
Autre valeur	`Int/Decimal/String`	La première valeur es l'aide des conversions appliquée sur cette va convertie, le résultat e
N'importe quelle valeur	Autre valeur	`Undefined`.

ltrim(String)

Supprime tous les espaces de début (tabulations et espaces) de la valeur `String` fournie. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`Ltrim(" h i ")` = "bonjour".

Type d'argument	Résultat
`Int`	La représentation `String` de `Int` avec tous les espaces de début supprimés.
`Decimal`	La représentation `String` de `Decimal` avec tous les espaces de début supprimés.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false ») avec tous les espaces de début supprimés.
`String`	L'argument avec tous les espaces de début supprimés.
Array	La représentation `String` de `Array` (à l'aide des règles de conversion standard) avec tous les espaces de début supprimés.
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard) avec tous les espaces de début supprimés.
Null	`Undefined`.
Non défini	`Undefined`.

machinelearning_predict(modelId)

Utilisation de l'`machinelearning_predict`Pour faire des prévisions en utilisant les données d'un message MQTT basé sur un modèle Amazon Machine Learning (Amazon ML). Prise en charge par SQL 2015-10-08 et versions ultérieures. Les arguments de la fonction `machinelearning_predict` sont :

modelId

L'ID du modèle sur lequel doit être réalisée la prévision. Le point de terminaison en temps réel du modèle doit être activé.

roleArn

Le rôle IAM qui a une stratégie avec`machinelearning:Predict`and`machinelearning:GetMLModel`et permet d'accéder au modèle par rapport auquel la prévision doit être réalisée.

record

Les données à transmettre dans l'API Amazon ML Predict. Elles doivent être représentées sous la forme d'un objet JSON à couche unique. Si l'enregistrement est un objet JSON multiniveau, il est mis à plat en sérialisant ses valeurs. Par exemple, le code JSON suivant :

```
{ "key1": {"innerKey1": "value1"}, "key2": 0}
```

deviendrait :

```
{ "key1": "{\"innerKey1\": \"value1\"}", "key2": 0}
```

La fonction renvoie un objet JSON dans les champs suivants :

predictedLabel

Classification de l'entrée basée sur le modèle.

détails

Contient les attributs suivants :

PredictiveModelType

Type de modèle. Les valeurs valides sont REGRESSION, BINARY, MULTICLASS.

Algorithm

L'algorithme utilisé par Amazon ML pour faire des prévisions. La valeur doit être SGD.

predictedScores

Contient le score de classification brut correspondant à chaque étiquette.

predictedValue

Valeur prévue par Amazon ML.

mod(Decimal, Decimal)

Renvoie le reste résultant de la division du premier argument par le deuxième argument. Équivalent à remainder(Decimal, Decimal) (p. 524). Vous pouvez également utiliser « % » comme opérateur infixe pour la même fonctionnalité modulo. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `mod(8, 3)` = 2.

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	`Int`, les premier et de vous voulez exécuter
`Int/Decimal`	`Int/Decimal`	`Decimal`, le premier pour lesquels vous vo Modulo.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes résultat est le premier argument. Sinon la va
Autre valeur	Autre valeur	`Undefined`.

nanvl(AnyValue, AnyValue)

Renvoie le premier argument s'il s'agit d'une valeur `Decimal` valide. Sinon, le deuxième argument est renvoyé. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `Nanvl(8, 3)` = 8.

Type d'argument 1	Type d'argument 2	Sortie
Non défini	N'importe quelle valeur	Le deuxième argumer
Null	N'importe quelle valeur	Le deuxième argumer
`Decimal` (NaN)	N'importe quelle valeur	Le deuxième argumer

Type d'argument 1	Type d'argument 2	Sortie
`Decimal` (non-NaN)	N'importe quelle valeur	Le premier argument.
Autre valeur	N'importe quelle valeur	Le premier argument.

newuuid()

Retourne un UUID aléatoire de 16 octets. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `newuuid()` = `123a4567-b89c-12d3-e456-789012345000`

numbytes(String)

Renvoie le nombre d'octets dans l'encodage UTF-8 de la chaîne fournie. Les règles de conversion standard s'appliquent aux arguments non-`String`. Pris en charge par SQL 2016-03-23 et versions ultérieures.

Exemples :

`numbytes("hi")` = 2

`numbytes("€")` = 3

parse_time (String, Long [, String])

Utilisez la fonction `parse_time` pour mettre en forme un horodatage dans un format date/heure lisible par l'utilisateur. Pris en charge par SQL 2016-03-23 et versions ultérieures. Pour convertir une chaîne d'horodatage en millisecondes, consulteztime_to_epoch (String, String) (p. 532).

La .`parse_time`La fonction attend les arguments suivants :

pattern

(String) Modèle date/heure suivant la valeurISO 8601format standard. Plus précisément, la fonction prend en chargeFormats Joda-Time.

timestamp

(Long) Heure à formater en millisecondes depuis l'époque Unix. Voir la fonction timestamp() (p. 533).

timezone

(String) Fuseau horaire de la date/heure mise en forme. La valeur par défaut est « UTC ». La fonction prend en charge les fuseaux horaires Joda-Time. Cet argument est facultatif.

Exemples :

Lorsque ce message est publié dans la rubrique « A/B », la charge utile `{"ts": "1970.01.01 AD at 21:46:40 CST"}` est envoyée au compartiment S3 :

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT parse_time("yyyy.MM.dd G 'at' HH:mm:ss z", 100000000, "America/
Belize" ) as ts FROM 'A/B'",

        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "s3": {
```

```
                    "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                    "bucketName": "BUCKET_NAME",
                    "key": "KEY_NAME"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

Lorsque ce message est publié dans la rubrique « A/B », une charge utile similaire à `{"ts": "2017.06.09 AD at 17:19:46 UTC"}` (mais avec la date et l'heure du moment) est envoyée au compartiment S3 :

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT parse_time("yyyy.MM.dd G 'at' HH:mm:ss z", timestamp() ) as ts FROM
 'A/B'",
        "awsIotSqlVersion": "2016-03-23",
        "ruleDisabled": false,
        "actions": [
            {
                "s3": {
                    "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                    "bucketName": "BUCKET_NAME",
                    "key": "KEY_NAME"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

`parse_time()` peut également servir de modèle de substitution. Par exemple, lorsque ce message est publié dans la rubrique « A/B », la charge utile est envoyée au compartiment S3 avec la clé = « 2017 » :

```
{
    "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
    "topicRulePayload": {
        "sql": "SELECT * FROM 'A/B'",
        "awsIotSqlVersion": "2016-03-23",
        "ruleDisabled": false,
        "actions": [
            {
                "s3": {
                    "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
                    "bucketName": BUCKET_NAME,
                    "key": "${parse_time("yyyy", timestamp(), "UTC")}"
                }
            }
        ],
        "ruleName": "RULE_NAME"
    }
}
```

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. Prise en charge par SQL 2015-10-08 et versions ultérieures. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `power(2, 5)` = 32.0.

Type d'argument 1	Type d'argument 2	Sortie
`Int/Decimal`	`Int/Decimal`	Une valeur `Decimal` argument renvoyé à l
`Int/Decimal/String`	`Int/Decimal/String`	Une valeur `Decimal` argument renvoyé à l Toutes les chaînes so valeur `String` échou résultat est `Undefine`
Autre valeur	Autre valeur	`Undefined`.

principal()

Renvoie le mandataire utilisé par le périphérique pour l'authentification, en fonction de la façon dont le message de déclenchement a été publié. Le tableau suivant décrit le mandataire renvoyé pour chaque méthode et protocole de publication.

Méthode de publication du message	Protocole	Type d'informations d
Client MQTT	MQTT	Certificat d'appareil X
Client MQTT de la console AWS IoT	MQTT	Utilisateur ou rôle IAM
AWS CLI	HTTP	Utilisateur ou rôle IAM
SDK pour les appareils AWS IoT	MQTT	Certificat d'appareil X
SDK pour les appareils AWS IoT	MQTT via WebSocket	Utilisateur ou rôle IAM

Les exemples suivants illustrent les différents types de valeur qui`principal()`peut retourner :

- Empreinte du certificat X.509 : `ba67293af50bf2506f5f93469686da660c7c844e7b3950bfb16813e0d31e9373`
- ID de rôle IAM et nom de session :`ABCD1EFG3HIJK2LMNOP5:my-session-name`
- renvoie un ID utilisateur :`ABCD1EFG3HIJK2LMNOP5`

rand()

Renvoie une valeur pseudo aléatoire, uniformément distribuée en double entre 0,0 et 1,0. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`rand()` = 0.8231909191640703

regexp_matches(String, String)

Renvoie la valeur true si la chaîne (le premier argument) contient un élément correspondant à l'expression régulière (le deuxième argument).

Exemple :

`regexp_matches("aaaa", "a{2,}")` = vrai.

`regexp_matches("aaaa", "b")` = false.

Premier argument :

Type d'argument	Résultat
`Int`	La représentation `String` de la valeur `Int`.
`Decimal`	La représentation `String` de la valeur `Decimal`.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false »).
`String`	La valeur `String`.
Array	La représentation `String` de la valeur `Array` (à l'aide des règles de conversion standard).
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard).
Null	`Undefined`.
Non défini	`Undefined`.

Deuxième argument :

Il doit s'agir d'une expression regex valide. Les types non-chaîne sont convertis en valeurs `String` à l'aide des règles de conversion standard. Selon le type, la chaîne résultante peut ne pas être une expression régulière valide. Si l'argument (converti) n'est pas un regex valide, le résultat est `Undefined`.

regexp_replace(String, String, String)

Remplace toutes les occurrences du deuxième argument (expression régulière) figurant dans le premier argument par le troisième argument. Fait référence aux groupes de capture avec « $ ». Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`regexp_replace("abcd", "bc", "x")` = "axd".

`regexp_replace("abcd", "b(.*)d", "$1")` = "ac".

Premier argument :

Type d'argument	Résultat
`Int`	La représentation `String` de la valeur `Int`.
`Decimal`	La représentation `String` de la valeur `Decimal`.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false »).
`String`	La valeur source.

Type d'argument	Résultat
Array	La représentation `String` de la valeur `Array` (à l'aide des règles de conversion standard).
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard).
Null	`Undefined`.
Non défini	`Undefined`.

Deuxième argument :

Il doit s'agir d'une expression regex valide. Les types non-chaîne sont convertis en valeurs `String` à l'aide des règles de conversion standard. Selon le type, la chaîne résultante peut ne pas être une expression régulière valide. Si l'argument (converti) n'est pas une expression regex valide, le résultat est `Undefined`.

Troisième argument :

Il doit s'agir d'une chaîne de remplacement regex valide. (Peut faire référence à d'autres groupes de capture.) Les types non-chaîne sont convertis en valeurs `String` à l'aide des règles de conversion standard. Si l'argument (converti) n'est pas une chaîne de remplacement regex valide, le résultat est `Undefined`.

regexp_substr(String, String)

Recherche la première correspondance du deuxième paramètre (regex) dans le premier paramètre. Fait référence aux groupes de capture avec « $ ». Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`regexp_substr("hihihello", "hi")` = "bonjour"

`regexp_substr("hihihello", "(hi)*")` = "hihi"

Premier argument :

Type d'argument	Résultat
`Int`	La représentation `String` de la valeur `Int`.
`Decimal`	La représentation `String` de la valeur `Decimal`.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false »).
`String`	L'argument `String`.
Array	La représentation `String` de la valeur `Array` (à l'aide des règles de conversion standard).
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard).
Null	`Undefined`.
Non défini	`Undefined`.

Deuxième argument :

Il doit s'agir d'une expression regex valide. Les types non-chaîne sont convertis en valeurs `String` à l'aide des règles de conversion standard. Selon le type, la chaîne résultante peut ne pas être une expression régulière valide. Si l'argument (converti) n'est pas une expression regex valide, le résultat est `Undefined`.

Troisième argument :

Il doit s'agir d'une chaîne de remplacement regex valide. (Peut faire référence à d'autres groupes de capture.) Les types non-chaîne sont convertis en valeurs `String` à l'aide des règles de conversion standard. Si l'argument n'est pas une chaîne de remplacement regex valide, le résultat est `Undefined`.

remainder(Decimal, Decimal)

Renvoie le reste résultant de la division du premier argument par le deuxième argument. Équivalent à mod(Decimal, Decimal) (p. 518). Vous pouvez également utiliser « % » comme opérateur infixe pour la même fonctionnalité modulo. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `remainder(8, 3)` = 2.

Opérande gauche	Opérande droit	Sortie
`Int`	`Int`	`Int`, les premier et de vous voulez exécuter
`Int/Decimal`	`Int/Decimal`	`Decimal`, le premier pour lesquels vous vo Modulo.
`String/Int/Decimal`	`String/Int/Decimal`	Si toutes les chaînes résultat est le premier argument. Sinon la va
Autre valeur	Autre valeur	`Undefined`.

replace(String, String, String)

Remplace toutes les occurrences du deuxième argument par le troisième argument dans le premier argument. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`replace("abcd", "bc", "x")` = `"axd"`.

`replace("abcdabcd", "b", "x")` = `"axcdaxcd"`.

Tous les arguments

Type d'argument	Résultat
`Int`	La représentation `String` de la valeur `Int`.
`Decimal`	La représentation `String` de la valeur `Decimal`.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false »).
`String`	La valeur source.
Array	La représentation `String` de la valeur `Array` (à l'aide des règles de conversion standard).

Type d'argument	Résultat
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard).
Null	`Undefined`.
Non défini	`Undefined`.

rpad(String, Int)

Renvoie l'argument chaîne, complété à droite par le nombre d'espaces spécifié dans le deuxième argument. L'argument `Int` doit être compris entre 0 et 1000. Si la valeur fournie se situe en dehors de cette plage valide, l'argument est défini sur la valeur valide la plus proche (0 ou 1 000). Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`rpad("hello", 2)` = `"hello "`.

`rpad(1, 3)` = `"1 "`.

Type d'argument 1	Type d'argument 2	Résultat
`String`	`Int`	L'argument `String` est complété à droite par un nombre d'espaces égal à la valeur `Int` fournie.
`String`	`Decimal`	L'argument `Decimal` est arrondi à la valeur `Int` inférieure la plus proche, et la chaîne est complétée à droite par un nombre

Type d'argument 1	Type d'argument 2	Résultat
		d'espaces égal à la valeur `Int` fournie.
`String`	`String`	Le deuxième argument est converti en une valeur `Decimal`, qui est arrondie à la valeur `Int` inférieure la plus proche. L'argument `String` est complété à droite par un nombre d'espaces égal à la valeur `Int` fournie.

Type d'argument 1	Type d'argument 2	Résultat
Autre valeur	`Int`/`Decimal`/`String`	La première valeur est convertie en une valeur `String` à l'aide des conversions standard, puis la fonction RPAD est appliquée sur cette valeur `String`. Si elle ne peut pas être convertie, le résultat est `Undefined`.
N'importe quelle 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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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.

Type d'argument	Résultat
`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`.

rtrim(String)

Supprime tous les espaces de fin (tabulations et espaces) de la valeur `String` fournie. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`rtrim(" h i ")` = " sa lut "

Type d'argument	Résultat
`Int`	La représentation `String` de la valeur `Int`.
`Decimal`	La représentation `String` de la valeur `Decimal`.
`Boolean`	La représentation `String` de la valeur booléenne (« true » ou « false »).
Array	La représentation `String` de la valeur `Array` (à l'aide des règles de conversion standard).
Objet	La représentation `String` de l'objet (à l'aide des règles de conversion standard).
Null	`Undefined`.
Non défini	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`sign(-7)` = -1.

`sign(0)` = 0.

`sign(13)` = 1.

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

Type d'argument	Résultat
`Decimal`	`Int`, le signe de la valeur `Decimal`.
`String`	`Int`, le signe de la valeur `Decimal`. La chaîne est convertie en une valeur `Decimal`, et le signe de la valeur `Decimal` est renvoyée. Si la valeur `String` ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined`. Prise en charge par SQL 2015-10-08 et versions ultérieures.
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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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` (avec double précision), le sinus de l'argument. Si la chaîne ne peut pas être convertie en une valeur `Decimal`, le résultat est `Undefined`.
Array	`Undefined`.
Objet	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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.

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

substring (String, Int [, Int])

Prévoit un argument `String` suivi par une ou deux valeurs `Int`. Pour un argument `String` et un seul argument `Int`, cette fonction renvoie la sous-chaîne de l'argument `String` fourni provenant de l'index (de base 0, inclus) `Int` fourni à la fin de l'argument `String`. Pour un argument `String` et deux arguments `Int`, cette fonction renvoie la sous-chaîne de l'argument `String` fourni provenant du premier argument d'index `Int` (de base 0, inclus) dans le deuxième argument d'index `Int` (de base 0, inclus). Les index qui sont inférieurs à zéro sont définis sur zéro. Les index qui sont supérieurs à la longueur de `String` sont définis sur la longueur de `String`. Pour la version des trois arguments, si le premier index est supérieur (ou égale) au deuxième index, le résultat et vide `String`.

Si les arguments fournis ne sont pas (*`Chaîne`*, *`Entier`*) ou (*`Chaîne`*, *`Entier`*, *`Entier`*), les conversions standard sont appliquées aux arguments pour tenter de les convertir dans les types corrects. Si les types ne peuvent pas être convertis, le résultat de la fonction est `Undefined`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`substring("012345", 0)` = "012345".

`substring("012345", 2)` = "2345".

`substring("012345", 2.745)` = "2345".

`substring(123, 2)` = "3".

`substring("012345", -1)` = "012345".

`substring(true, 1.2)` = "true".

`substring(false, -2.411E247)` = "false".

`substring("012345", 1, 3)` = "12".

`substring("012345", -50, 50)` = "012345".

`substring("012345", 3, 1)` = "".

sql_version()

Renvoie la version SQL spécifiée dans cette règle. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`sql_version()` = "2016-03-23"

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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `sqrt(9)` = 3,0.

Type d'argument	Résultat
`Int`	La racine carrée de l'argument.
`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`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

startswith(String, String)

Renvoie une valeur `Boolean` si le premier argument de type chaîne commence par le deuxième argument de type chaîne. Si l'un des arguments est `Null` ou `Undefined`, le résultat a la valeur `Undefined`. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`startswith("ranger","ran")` = true

Type d'argument 1	Type d'argument 2	Résultat
`String`	`String`	Si la première chaîne
Autre valeur	Autre valeur	Les deux arguments s règles de conversion la première chaîne co l'un des arguments es la valeur `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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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`.
Array	`Undefined`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`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. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `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`.
Objet	`Undefined`.
Null	`Undefined`.
Non défini	`Undefined`.

time_to_epoch (String, String)

Utilisation de l'`time_to_epoch`pour convertir une chaîne d'horodatage en un nombre de millisecondes dans l'heure Unix. Pris en charge par SQL 2016-03-23 et versions ultérieures. Pour convertir

des millisecondes en une chaîne d'horodatage formatée, consultezparse_time (String, Long [, String]) (p. 519).

La `.time_to_epoch`La fonction attend les arguments suivants :

timestamp

(String) Chaîne d'horodatage à convertir en millisecondes depuis l'époque Unix. Si la chaîne d'horodatage ne spécifie pas de fuseau horaire, la fonction utilise le fuseau horaire UTC.

pattern

(String) Modèle date/heure suivant la valeurISO 8601format standard. Plus précisément, la fonction prend en chargeFormats de temps JDK11.

Exemples :

`time_to_epoch("2020-04-03 09:45:18 UTC+01:00", "yyyy-MM-dd HH:mm:ss VV")`= 1585903518000

`time_to_epoch("18 December 2015", "dd MMMM yyyy")`= 1450396800000

`time_to_epoch("2007-12-03 10:15:30.592 America/Los_Angeles", "yyyy-MM-dd HH:mm:ss.SSS z")`= 1196705730592

timestamp()

Il renvoie l'horodatage actuel en millisecondes à partir de 00:00:00 UTC (temps universel coordonné), jeudi 1er janvier 1970, comme observé par le moteur des règles AWS IoT. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple: `timestamp()` = `1481825251155`

topic(Decimal)

Il renvoie la rubrique vers laquelle le message qui a déclenché la règle a été envoyé. Si aucun paramètre n'est indiqué, la rubrique entière est renvoyée. Le paramètre `Decimal` est utilisé pour spécifier un segment de rubrique spécifique, avec le chiffre 1 désignant le premier segment. Pour la rubrique `foo/bar/baz`, topic(1) renvoie `foo`, topic(2) renvoie `bar`, et ainsi de suite. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`topic()` = "things/myThings/thingOne"

`topic(1)` = "things"

Lorsque Basic Ingest (p. 480) est utilisé, le préfixe initial de la rubrique (`$aws/rules/`*`rule-name`*) n'est pas disponible pour la fonction topic(). Prenons l'exemple de la rubrique suivante :

`$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`

`topic()` = "Buildings/Building5/Floor2/Room201/Lights"

`topic(3)` = "Floor2"

traceid()

Renvoie l'ID de suivi (UUID) du message MQTT ou une valeur `Undefined` si le message n'a pas été pas envoyé via MQTT. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`traceid()` = "12345678-1234-1234-1234-123456789012"

transform (chaîne, objet, tableau)

Retourne un tableau d'objets qui contient le résultat de la transformation spécifiée de l'objet`Object`sur la stratégie`Array` Paramètre .

Pris en charge par SQL 2016-03-23 et versions ultérieures.

Chaîne

Le mode de transformation à utiliser. Reportez-vous au tableau suivant pour connaître les modes de transformation pris en charge et la façon dont ils créent le`Result`à partir des`Object`and`Array`Paramètres.

Objet

Objet qui contient les attributs à appliquer à chaque élément de la propriété`Array`.

Array

Tableau d'objets dans lesquels les attributs de`Object`sont appliquées.

Chaque objet de ce tableau correspond à un objet dans la réponse de la fonction. Chaque objet de la réponse de la fonction contient les attributs présents dans l'objet d'origine et les attributs fournis par`Object`tel que déterminé par le mode de transformation spécifié dans`String`.

`String` paramètre	**`Object`** paramètre	**`Array`** paramètre	Résultat
`enrichArray`	Objet	Tableau d'objets	Un tableau d'objets dans lequel chaque objet contient les attributs d'un élément de la`Array`et les attributs du paramètre`Object` Paramètre .
Toute autre valeur	N'importe quelle valeur	N'importe quelle valeur	Non défini

Note

Le tableau retourné par cette fonction est limité à 128 KiB.

Exemple 1 de fonction de transformation

Cet exemple montre commenttransform()produit un seul tableau d'objets à partir d'un objet de données et d'un tableau.

Dans cet exemple, le message suivant est publié dans la rubrique MQTT`A/B`.

```
{
    "attributes": {
        "data1": 1,
        "data2": 2
    },
```

```
    "values": [
        {
            "a": 3
        },
        {
            "b": 4
        },
        {
            "c": 5
        }
    ]
}
```

Cette instruction SQL pour une action de règle de rubrique utilise la méthodetransform()avec une fonction`String`Valeur de`enrichArray`. Dans cet exemple,`Object`est le`attributes`de la charge utile du message et`Array`est le`values`, qui contient trois objets.

```
select value transform("enrichArray", attributes, values) from 'A/B'
```

Lors de la réception de la charge utile du message, l'instruction SQL est évaluée à la réponse suivante.

```
[
  {
    "a": 3,
    "data1": 1,
    "data2": 2
  },
  {
    "b": 4,
    "data1": 1,
    "data2": 2
  },
  {
    "c": 5,
    "data1": 1,
    "data2": 2
  }
]
```

Exemple 2 de la fonction Transform

Cet exemple montre commenttransform()peut utiliser des valeurs littérales pour inclure et renommer des attributs individuels de la charge utile du message.

Dans cet exemple, le message suivant est publié dans la rubrique MQTT`A/B`. Voici le même message que celui qui a été utilisé dansthe section called “Exemple 1 de fonction de transformation” (p. 534).

```
{
    "attributes": {
        "data1": 1,
        "data2": 2
    },
    "values": [
        {
            "a": 3
        },
        {
            "b": 4
        },
        {
```

```
            "c": 5
        }
    ]
}
```

Cette instruction SQL pour une action de règle de rubrique utilise la méthodetransform()avec une fonction`String`Valeur de`enrichArray`. La `.Object`dans letransform()a un attribut unique nommé`key`avec la valeur de`attributes.data1`dans la charge utile du message et`Array`est le`values`, qui contient les trois mêmes objets utilisés dans l'exemple précédent.

```
select value transform("enrichArray", {"key": attributes.data1}, values) from 'A/B'
```

Lors de la réception de la charge utile du message, cette instruction SQL est évaluée à la réponse suivante. Notez comment la stratégie`data1`est nommée`key`Dans la réponse.

```
[
  {
    "a": 3,
    "key": 1
  },
  {
    "b": 4,
    "key": 1
  },
  {
    "c": 5,
    "key": 1
  }
]
```

Exemple 3 de la fonction Transform

Cet exemple montre commenttransform()peut être utilisée dans les clauses SELECT imbriquées pour sélectionner plusieurs attributs et créer de nouveaux objets pour un traitement ultérieur.

Dans cet exemple, le message suivant est publié dans la rubrique MQTT`A/B`.

```
{
  "data1": "example",
  "data2": {
    "a": "first attribute",
    "b": "second attribute",
    "c": [
      {
        "x": {
          "someInt": 5,
          "someString": "hello"
        },
        "y": true
      },
      {
        "x": {
          "someInt": 10,
          "someString": "world"
        },
        "y": false
      }
    ]
  }
}
```

La .`Object`pour cette fonction de transformation est l'objet retourné par l'instruction SELECT, qui contient leaandbéléments de la`data2`Objet. La .`Array`se compose des deux objets de la propriété`data2.c`dans le message original.

```
select value transform('enrichArray', (select a, b from data2), (select value c from
 data2)) from 'A/B'
```

Avec le message précédent, l'instruction SQL est évaluée à la réponse suivante.

```
[
  {
    "x": {
      "someInt": 5,
      "someString": "hello"
    },
    "y": true,
    "a": "first attribute",
    "b": "second attribute"
  },
  {
    "x": {
      "someInt": 10,
      "someString": "world"
    },
    "y": false,
    "a": "first attribute",
    "b": "second attribute"
  }
]
```

Le tableau retourné dans cette réponse pourrait être utilisé avec des actions de règle de rubrique qui prennent en charge`batchMode`.

trim(String)

Supprime tous les espaces de début et de fin de la valeur `String` fournie. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemple :

`Trim(" hi ")` = "bonjour"

Type d'argument	Résultat
`Int`	La représentation `String` de `Int` avec tous les espaces de début et de fin supprimés.
`Decimal`	La représentation `String` de `Decimal` avec tous les espaces de début et de fin supprimés.
`Boolean`	La représentation `String` de la valeur `Boolean` (« true » ou « false ») avec tous les espaces de début et de fin supprimés.
`String`	L'argument `String` avec tous les espaces de début et de fin supprimés.
Array	La représentation `String` de la valeur `Array` à l'aide des règles de conversion standard.

Type d'argument	Résultat
Objet	La représentation `String` de l'objet à l'aide des règles de conversion standard.
Null	`Undefined`.
Non défini	`Undefined`.

trunc(Decimal, Int)

Tronque le premier argument du nombre de `Decimal`, spécifié par le deuxième argument. Si le deuxième argument est inférieur à zéro, il est défini sur zéro. Si le deuxième argument est supérieur à 34, il est défini sur 34. Les zéros de fin sont supprimés du résultat. Prise en charge par SQL 2015-10-08 et versions ultérieures.

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`	`Int/Decimal`	Le premier argument décrite par le deuxièm s'il ne s'agit pas d'un inférieure la plus proc
`Int/Decimal/String`	`Int/Decimal`	Le premier argument décrite par le deuxièm s'il ne s'agit pas d'un inférieure la plus proc en une valeur `Decima` convertie, le résultat e
Autre valeur		`Undefined`.

upper(String)

Renvoie la version en majuscules de la valeur `String` donnée. Les arguments non-`String` sont convertis en valeurs `String` à l'aide des règles de conversion standard. Prise en charge par SQL 2015-10-08 et versions ultérieures.

Exemples :

`upper("hello")` = "BONJOUR"

`upper(["hello"])` = "[\"BONJOUR\"]"

Literals

Vous pouvez spécifier directement des objets littéraux dans les clauses SELECT et WHERE de votre règle SQL, qui permet de transmettre des informations.

> **Note**
>
> Les littéraux sont disponibles uniquement lors de l'utilisation de SQL 2016-03-23 ou versions ultérieures.

Une syntaxe d'objet JSON est utilisée (paires clé-valeur, séparées par des virgules, où les clés sont des chaînes, et les valeurs des valeurs JSON, entourées d'accolades {}). Exemples :

Charge utile entrante publiée dans une rubrique `topic/subtopic` : `{"lat_long": [47.606,-122.332]}`

Instruction SQL : `SELECT {'latitude': get(lat_long, 0),'longitude':get(lat_long, 1)} as lat_long FROM 'topic/subtopic'`

La charge utile sortante résultante serait : `{"lat_long": {"latitude":47.606,"longitude":-122.332}}`.

Vous pouvez également spécifier des tableaux dans des clauses SELECT et WHERE de votre règle SQL, qui vous permet de regrouper des informations. Une syntaxe JSON est utilisée (éléments séparés par des virgules entre crochets [] pour créer un littéral de tableau). Exemples :

Charge utile entrante publiée dans une rubrique `topic/subtopic` : `{"lat": 47.696, "long": -122.332}`

Instruction SQL : `SELECT [lat,long] as lat_long FROM 'topic/subtopic'`

La charge utile de sortie serait : `{"lat_long": [47.606,-122.332]}`.

Instructions Case

Les instructions case peuvent être utilisées pour l'exécution de branche, comme une instruction switch ou des instructions if/else.

Syntaxe:

```
CASE v WHEN t[1] THEN r[1]
  WHEN t[2] THEN r[2] ...
  WHEN t[n] THEN r[n]
  ELSE r[e] END
```

L'expression `v` est évaluée et fait l'objet d'une comparaison d'égalité avec chaque expression `t[i]`. Si une correspondance est trouvée, l'expression `r[i]` correspondante devient le résultat de l'instruction case. S'il existe plusieurs correspondances possibles, la première est sélectionnée. S'il n'y a aucune correspondance, l'instruction else de `re` est utilisée comme résultat. S'il n'existe aucune correspondance et aucune instruction, le résultat de l'instruction case est `Undefined`. Exemples :

Charge utile entrante publiée dans une rubrique `topic/subtopic` : `{"color":"yellow"}`

Instruction SQL : `SELECT CASE color WHEN 'green' THEN 'go' WHEN 'yellow' THEN 'caution' WHEN 'red' THEN 'stop' ELSE 'you are not at a stop light' END as instructions FROM 'topic/subtopic'`

La charge utile de sortie serait : `{"instructions":"caution"}`.

Les instructions case requièrent au moins une clause WHEN. Une clause ELSE n'est pas obligatoire.

Note

Si v est `Undefined`, le résultat de l'instruction case est `Undefined`.

Extensions JSON

Vous pouvez utiliser les extensions suivantes de la syntaxe SQL ANSI pour faciliter l'utilisation d'objets JSON imbriqués.

"."."."

Cet opérateur accède aux membres dans les objets et les fonctions JSON imbriqués de la même manière qu'au SQL ANSI et à JavaScript. Exemples :

```
SELECT foo.bar AS bar.baz FROM 'topic/subtopic'
```

sélectionne la valeur de la propriété`bar`dans la stratégie`foo`à partir de la charge utile de message suivante envoyée à l'objet`topic/subtopic`.

```
{
  "foo": {
    "bar": "RED",
    "bar1": "GREEN",
    "bar2": "BLUE"
  }
}
```

Si un nom de propriété JSON inclut un trait d'union ou des caractères numériques, la simple notation « point » ne fonctionnera pas. Vous devez plutôt utiliser l'Obtenir une fonction (p. 509)pour extraire la valeur de la propriété.

Dans cet exemple, le message suivant est envoyé à l'`iot/rules`.

```
{
  "mydata": {
    "item2": {
      "0": {
        "my-key": "myValue"
      }
    }
  }
}
```

Normalement, la valeur de`my-key`serait identifié comme dans cette requête.

```
SELECT * from iot/rules WHERE mydata.item2.0.my-key= "myValue"
```

Cependant, parce que le nom de la propriété`my-key`contient un trait d'union et`item2`contient un caractère numérique, leObtenir une fonction (p. 509)doit être utilisé comme le montre la requête suivante.

```
SELECT * from 'iot/rules' WHERE get(get(get(mydata,"item2"),"0"),"my-key") = "myValue"
```

`*` Opérateur

Cela fonctionne de la même manière que le caractère générique `*` dans SQL ANSI. Il est utilisé dans la clause SELECT uniquement et crée un nouvel objet JSON contenant les données du message. Si la

charge utile du message n'est pas au format JSON, * renvoie la charge utile du message entier sous la forme d'octets bruts. Exemples :

```
SELECT * FROM 'topic/subtopic'
```

Appliquer une fonction à une valeur d'attribut

Voici un exemple de charge utile JSON qui peut être publiée par un appareil :

```
{
    "deviceid" : "iot123",
    "temp" : 54.98,
    "humidity" : 32.43,
    "coords" : {
        "latitude" : 47.615694,
        "longitude" : -122.3359976
    }
}
```

L'exemple suivant applique une fonction à une valeur d'attribut dans une charge utile JSON :

```
SELECT temp, md5(deviceid) AS hashed_id FROM topic/#
```

Le résultat de cette requête est l'objet JSON suivant :

```
{
   "temp": 54.98,
   "hashed_id": "e37f81fb397e595c4aeb5645b8cbbbd1"
}
```

Modèles de substitution

Vous pouvez utiliser un modèle de substitution pour alimenter les données JSON renvoyées lorsqu'une règle est déclenchée et que AWS IoT exécute une action. La syntaxe d'un modèle de substitution est `${expression}`, où expression peut être n'importe quelle expression prise en charge par AWS IoT dans les clauses SELECT, les clauses WHERE et Actions de règle AWS IoT (p. 401). Cette expression peut être connectée à un champ d'action d'une règle, ce qui vous permet de configurer dynamiquement une action. En effet, cette fonctionnalité remplace un élément d'information dans une action. Cela inclut les fonctions, les opérateurs et les informations présents dans la charge utile du message d'origine.

> **Important**
>
> Puisqu'une expression dans un modèle de substitution est évaluée séparément de l'instruction « SELECT... », vous ne pouvez pas référencer un alias créé à l'aide de la clause AS. Vous pouvez référencer uniquement les informations présentes dans la charge utile d'origine,functions (p. 495), etOpérateurs (p. 489).

Pour plus d'informations concernant les expressions prises en charge, consultez la page Référence SQL AWS IoT (p. 482).

Les actions de règle suivantes prennent en charge les modèles de substitution. Chaque action prend en charge différents champs qui peuvent être substitués.

- Apache Kafka (p. 403)
- Alarmes CloudWatch (p. 408)
- CloudWatch Logs (p. 410)

- Métriques CloudWatch (p. 411)
- DynamoDB (p. 412)
- DynamoDBv2 (p. 415)
- Elasticsearch (p. 416)
- HTTPS (p. 418)
- IoT Analytics (p. 420)
- IoT Events (p. 422)
- IoT SiteWise (p. 423)
- Kinesis Data Streams (p. 429)
- Kinesis Data Firehose (p. 427)
- Lambda (p. 431)
- Republish (p. 433)
- S3 (p. 434)
- SNS (p. 437)
- SQS (p. 438)
- Step Functions (p. 440)
- Timestream (p. 441)

Les modèles de substitution apparaissent dans les paramètres d'action au sein d'une règle :

```
{
    "sql": "SELECT *, timestamp() AS timestamp FROM 'my/iot/topic'",
    "ruleDisabled": false,
    "actions": [{
        "republish": {
            "topic": "${topic()}/republish",
            "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
        }
    }]
}
```

Si cette règle est déclenchée par le code JSON suivant publié dans `my/iot/topic` :

```
{
    "deviceid": "iot123",
    "temp": 54.98,
    "humidity": 32.43,
    "coords": {
        "latitude": 47.615694,
        "longitude": -122.3359976
    }
}
```

Ensuite, cette règle publie le code JSON suivant dans `my/iot/topic/republish`, que AWS IoT remplace à partir de `${topic()}/republish` :

```
{
    "deviceid": "iot123",
    "temp": 54.98,
    "humidity": 32.43,
    "coords": {
        "latitude": 47.615694,
        "longitude": -122.3359976
```

```
    },
    "timestamp": 1579637878451
}
```

Requêtes d'objets imbriqués

Vous pouvez utiliser des clauses SELECT imbriquées pour interroger les attributs dans les tableaux et les objets JSON internes. Pris en charge par SQL 2016-03-23 et versions ultérieures.

Examinez le message MQTT suivant :

```
{
    "e": [
        { "n": "temperature", "u": "Cel", "t": 1234, "v": 22.5 },
        { "n": "light", "u": "lm", "t": 1235, "v": 135 },
        { "n": "acidity", "u": "pH", "t": 1235, "v": 7 }
    ]
}
```

Example

Vous pouvez convertir des valeurs en un nouveau tableau avec la règle suivante.

```
SELECT (SELECT VALUE n FROM e) as sensors FROM 'my/topic'
```

La règle génère le résultat suivant.

```
{
    "sensors": [
        "temperature",
        "light",
        "acidity"
    ]
}
```

Example

En utilisant le même message MQTT, vous pouvez également interroger une valeur spécifique dans un objet imbriqué avec la règle suivante.

```
SELECT (SELECT v FROM e WHERE n = 'temperature') as temperature FROM 'my/topic'
```

La règle génère le résultat suivant.

```
{
    "temperature": [
        {
            "v": 22.5
        }
    ]
}
```

Example

Vous pouvez également aplatir la sortie avec une règle plus compliquée.

```
SELECT get((SELECT v FROM e WHERE n = 'temperature'), 0).v as temperature FROM 'topic'
```

La règle génère le résultat suivant.

```
{
    "temperature": 22.5
}
```

Utilisation des charges utiles binaires

Lorsque la charge utile du message doit être traitée sous la forme de données binaires brutes (et non comme objet JSON), utilisez l'opérateur« * » pour y faire référence dans une clause `SELECT`. Cela fonctionne pour les charges utiles autres que JSON avec certaines actions de règle, telles que l'action S3.

Pour pouvoir utiliser * pour faire référence à la charge utile du message en tant que données binaires brutes, suivez les règles suivantes :

1. L'instruction SQL et les modèles ne doivent pas faire référence à des noms JSON autres que *.
2. La .`SELECT`doit avoir * comme seul élément ou ne comporter que des fonctions. Consultez l'exemple suivant.

```
SELECT * FROM 'topic/subtopic'
```

```
SELECT encode(*, 'base64') AS data, timestamp() AS ts FROM 'topic/subtopic'
```

Pour les actions de règle qui ne prennent pas en charge l'entrée de charge utile binaire, telles queAction Lambda, vous devez décoder les charges utiles binaires. L'action de la règle Lambda peut recevoir des données binaires si elles sont codées en base64 et stockées dans une charge utile JSON. Pour cela, apportez les modifications suivantes à la règle :

```
SELECT encode(*, 'base64') AS data FROM 'my_topic'
```

Exemples de charge utile binaire

La clause `SELECT` suivante peut être utilisée avec des charges utiles binaires, car elle ne se réfère pas à des noms JSON.

```
SELECT * FROM 'topic/subtopic'
```

La clause `SELECT` suivante ne peut pas être utilisée avec des charges utiles binaires, car elle fait référence à `device_type` dans la clause WHERE.

```
SELECT * FROM 'topic/subtopic' WHERE device_type = 'thermostat'
```

La clause `SELECT` suivante ne peut pas être utilisée avec des charges utiles binaires, car elle ne respecte pas la règle n° 2.

```
SELECT *, timestamp() AS timestamp FROM 'topic/subtopic'
```

Vous pouvez utiliser la clause `SELECT` avec des charges utiles binaires car elle est conforme à la règle n° 1 ou à la règle n° 2.

```
SELECT * FROM 'topic/subtopic' WHERE timestamp() % 12 = 0
```

La règle AWS IoT suivante ne peut pas être utilisée avec des charges utiles binaires, car elle ne respecte pas la règle n° 1.

```
{
    "sql": "SELECT * FROM 'topic/subtopic'"
    "actions": [{
        "republish": {
            "topic":"device/${device_id}"
         }
     }]
}
```

Versions de SQL

Le moteur de règles AWS IoT utilise une syntaxe de type SQL pour sélectionner les données dans des messages MQTT. Les instructions SQL sont interprétées en fonction d'une version SQL spécifiée avec la propriété `awsIotSqlVersion` dans un document JSON qui décrit la règle. Pour plus d'informations sur la structure des documents de règle JSON, consultez Création d'une règle (p. 397). La propriété `awsIotSqlVersion` permet de spécifier la version du moteur de règles SQL AWS IoT que vous voulez utiliser. Lorsqu'une nouvelle version est déployée, vous pouvez continuer à utiliser une version antérieure ou modifier votre règle pour utiliser la nouvelle version. Vos règles actuelles continuent d'utiliser la version avec laquelle elles ont été créées.

L'exemple de code JSON suivant montre comment spécifier la version SQL à l'aide la propriété `awsIotSqlVersion`.

```
{
    "sql": "expression",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [{
        "republish": {
            "topic": "my-mqtt-topic",
            "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
        }
    }]
}
```

AWS IoT prend actuellement en charge les versions suivantes de SQL :

- `2015-10-08`— La version de SQL d'origine créée le 08-10-2015.
- `2016-03-23`— La version de SQL créée le 23-03-2016 ;
- `beta`— La version bêta de SQM la plus récente. L'utilisation de cette version peut entraîner des modifications de décomposition dans vos règles.

Nouveautés de la version 2016-03-23 du moteur de règles SQL

- Correctifs pour sélectionner les objets JSON imbriqués.
- Correctifs pour les requêtes de tableaux.
- Prise en charge des requêtes inter-objet. Pour plus d'informations, consultez Requêtes d'objets imbriqués (p. 543).
- Prise en charge de la génération en sortie d'un tableau comme un objet de niveau supérieur.

- Ajout de la fonction `encode(value, encodingScheme)`, qui peut être appliquée sur les données de format JSON et non-JSON. Pour plus d'informations, consultez la fonction d'encodage (p. 508).

Générer un `Array` en sortie comme un objet de niveau supérieur

Cette fonction permet à une règle de retourner un tableau comme un objet de niveau supérieur. Par exemple, avec le message MQTT suivant :

```
{
    "a": {"b":"c"},
    "arr":[1,2,3,4]
}
```

Et la règle suivante :

```
SELECT VALUE arr FROM 'topic'
```

La règle génère le résultat suivant.

```
[1,2,3,4]
```

Service AWS IoT Device Shadow

Le service AWS IoT Device Shadow ajoute des shadows aux objets d'objet AWS IoT. Les shadows peuvent mettre l'état d'un appareil à la disposition d'applications et d'autres services, que l'appareil soit connecté ou non à AWS IoT. Les objets d'objet AWS IoT peuvent avoir plusieurs shadows nommés afin que votre solution IoT ait plus d'options pour connecter vos appareils à d'autres applications et services.

Les objets d'objet AWS IoT n'ont pas de shadow nommé tant qu'ils n'ont pas été créés explicitement. Toutefois, un shadow classique non nommé est créé pour un objet quand l'objet est créé. Les shadows peuvent être créés, mis à jour et supprimés à l'aide de la console AWS IoT. Les appareils, les autres clients Web et les services peuvent créer, mettre à jour et supprimer des shadows à l'aide de MQTT et deRubriques réservées MQTT (p. 103), HTTP en utilisant leAPI REST Device Shadow (p. 573), et leAWS CLIpourAWS IoT. Parce que les ombres sont stockées parAWSDans le cloud, ils peuvent collecter et rapporter les données d'état de l'appareil à partir d'applications et d'autres services cloud, que l'appareil soit connecté ou non.

Rubriques

Utilisation des shadows

Les shadows fournissent un magasin de données fiable pour que les appareils, les applications et d'autres services cloud partagent des données. Ils permettent aux appareils, aux applications et aux autres services cloud de se connecter et déconnecter sans perdre l'état d'un appareil.

Pendant que les appareils, les applications et les autres services cloud sont connectés à AWS IoT, ils peuvent accéder à l'état actuel d'un appareil et le contrôler via ses shadows. Par exemple, une application peut demander une modification de l'état d'un appareil en mettant à jour un shadow. AWS IoT publie un message indiquant la modification apportée à l'appareil. L'appareil reçoit ce message, met à jour son état pour qu'il corresponde et publie un message avec son état mis à jour. Le service Device Shadow reflète cet état mis à jour dans le shadow correspondant. L'application peut s'abonner à la mise à jour du shadow ou interroger le shadow pour obtenir son état actuel.

Lorsqu'un appareil passe hors connexion, une application peut continuer à communiquer avec AWS IoT et avec les shadows de l'appareil. Lorsque l'appareil se reconnecte, il reçoit l'état actuel de ses shadows pour pouvoir mettre à jour son état afin que ce dernier corresponde à celui de ses shadows, et pour pouvoir publier un message avec son état mis à jour. De même, lorsqu'une application passe hors connexion et

que l'état de l'appareil change alors qu'elle est hors connexion, l'appareil conserve le shadow mis à jour afin que l'application puisse interroger les shadows pour connaître son état actuel lorsqu'elle se reconnecte.

Choix d'utilisation de shadows nommés ou non nommés

Le service Device Shadow prend en charge les shadows classiques nommés et non nommés, tels qu'ils étaient utilisés par le passé. Un objet d'objet peut avoir plusieurs shadows nommés, mais pas plus d'un shadow classique non nommé. Un objet d'objet peut avoir des shadows nommés et non nommés en même temps. Toutefois, l'API utilisée pour accéder à chacun d'eux est légèrement différente. Il pourrait donc être plus efficace de décider quel type de shadow fonctionnerait le mieux pour votre solution et d'utiliser ce type uniquement. Pour de plus amples informations sur l'API permettant d'accéder aux shadows, veuillez consulter Rubriques de shadow (p. 103).

Avec les shadows nommés, vous pouvez créer différentes vues de l'état d'un objet d'objet. Par exemple, vous pouvez diviser un objet d'objet avec de nombreuses propriétés en shadows avec des groupes logiques de propriétés, chacun identifié par son nom de shadow. Vous pouvez également limiter l'accès aux propriétés en les regroupant dans différents shadows et en utilisant des stratégies pour contrôler l'accès. Les ombres nommées, cependant, ne prennent pas en chargeIndexation de parc (p. 758).

Les shadows classiques non nommés sont plus simples, mais un peu plus limités que les shadows nommés. Chaque objet d'objet AWS IoT peut avoir un seul shadow non nommé. Si vous prévoyez que votre solution IoT aura un besoin limité de données de shadow, c'est peut-être ainsi que vous souhaitez commencer à utiliser les shadows. Toutefois, si vous pensez ajouter des shadows supplémentaires à l'avenir, envisagez d'utiliser des shadows nommés dès le début.

Accès aux shadows

Chaque shadow a une rubrique MQTT (p. 103) réservée et une URL HTTP (p. 573) qui prend en charge les actions `get`, `update` et `delete` sur le shadow.

Les shadows utilisent des documents de shadow JSON (p. 585) pour stocker et récupérer des données. Un document shadow contient une propriété d'état qui décrit les aspects suivants de l'état de l'appareil :

- `desired`

  Les applications spécifient les états souhaités des propriétés de l'appareil en mettant à jour l'objet `desired`.
- `reported`

  Les appareils rapportent leur état actuel dans l'objet `reported`.
- `delta`

  AWS IoT rapporte les différences entre l'état souhaité et l'état rapporté dans l'objet `delta`.

Les données stockées dans un shadow sont déterminées par la propriété d'état du corps du message de l'action de mise à jour. Les actions de mise à jour suivantes peuvent modifier les valeurs d'un objet de données existant, ainsi qu'ajouter et supprimer des clés et d'autres éléments dans l'objet d'état du shadow. Pour de plus amples informations sur l'accès aux shadows, veuillez consulter Utilisation des shadows sur les appareils (p. 551) et Utilisation des shadows dans les applications et les services (p. 554).

> Important
>
> L'autorisation d'effectuer des demandes de mise à jour doit être limitée aux applications et aux appareils approuvés. Cela empêche la propriété d'état du shadow d'être modifiée de manière

inattendue. Sinon, les appareils et les applications qui utilisent le shadow doivent être conçus de manière à s'attendre à ce que les clés figurant dans la propriété d'état changent.

Utilisation des shadows sur les appareils, dans les applications et dans d'autres services cloud

L'utilisation de shadows sur les appareils, dans les applications et dans d'autres services cloud nécessite une cohérence et une coordination entre tous ces éléments. Le service AWS IoT Device Shadow stocke l'état du shadow, envoie des messages lorsque cet état change et répond aux messages qui changent son état. Les appareils, applications et autres services cloud de votre solution IoT doivent gérer leur état et le maintenir cohérent avec l'état du shadow de l'appareil.

Les données d'état du shadow sont dynamiques et peuvent être modifiées par les appareils, les applications et les autres services cloud dotés de l'autorisation d'accéder au shadow. Pour cette raison, il est important de considérer comment chaque appareil, application et autre service cloud interagira avec le shadow. Exemples :

- Lesappareils doivent écrire uniquement dans la propriété `reported` de l'état du shadow lors de la communication des données d'état au shadow.
- Les applications et autres services cloud doivent écrire uniquement dans la propriété `desired` lors de la communication des demandes de changement d'état à l'appareil via le shadow.

> **Important**
>
> Les données contenues dans un objet de données de shadow sont indépendantes de celles des autres shadows et des autres propriétés de l'objet d'objet, telles que les attributs d'un objet et le contenu des messages MQTT que l'appareil d'un objet d'objet peut publier. Toutefois, un appareil peut rapporter les mêmes données dans différents shadows et différentes rubriques MQTT, si nécessaire.
> Un appareil qui prend en charge plusieurs shadows doit maintenir la cohérence des données qu'il rapporte dans les différents shadows.

Ordre des messages

Il n'existe aucune garantie que les messages émanant du service AWS IoT parviendront au dispositif dans un ordre spécifique. Le scénario suivant montre ce qui se passe dans ce cas.

Document d'état initial :

```
{
  "state": {
    "reported": {
      "color": "blue"
    }
  },
  "version": 9,
  "timestamp": 123456776
}
```

Mise à jour 1 :

```
{
  "state": {
    "desired": {
      "color": "RED"
```

```
    }
  },
  "version": 10,
  "timestamp": 123456777
}
```

Mise à jour 2 :

```
{
  "state": {
    "desired": {
      "color": "GREEN"
    }
  },
  "version": 11,
  "timestamp": 123456778
}
```

Document d'état final :

```
{
  "state": {
    "reported": {
      "color": "GREEN"
    }
  },
  "version": 12,
  "timestamp": 123456779
}
```

Il en résulte deux messages delta :

```
{
  "state": {
    "color": "RED"
  },
  "version": 11,
  "timestamp": 123456778
}
```

```
{
  "state": {
    "color": "GREEN"
  },
  "version": 12,
  "timestamp": 123456779
}
```

L'appareil peut recevoir ces messages dans le désordre. Etant donné que l'état de ces messages est cumulé, un dispositif peut ignorer en toute sécurité tous les messages qui contiennent un numéro de version plus ancien que celui qu'il suit. Si le dispositif reçoit le delta pour la version 12 avant la version 11, il peut en toute sécurité ignorer le message concernant la version 11.

Suppression des messages de shadow

Pour réduire la taille des messages de shadow envoyés à votre appareil, définissez une règle qui sélectionne uniquement les champs dont votre appareil a besoin, puis republie le message dans une rubrique MQTT que votre appareil écoute.

La règle est spécifiée dans JSON et doit ressembler à ceci :

```
{
  "sql": "SELECT state, version FROM '$aws/things/+/shadow/update/delta'",
  "ruleDisabled": false,
  "actions": [
    {
      "republish": {
        "topic": "${topic(3)}/delta",
        "roleArn": "arn:aws:iam:123456789012:role/my-iot-role"
      }
    }
  ]
}
```

L'instruction SELECT détermine quels champs du message seront republiés dans la rubrique spécifiée. Un caractère générique « + » est utilisé pour apparier tous les noms de shadow. La règle spécifie que tous les messages correspondants doivent être republiés dans la rubrique spécifiée. Dans ce cas, la fonction `"topic()"` est utilisée pour indiquer la rubrique dans laquelle republier. `topic(3)` correspond à la valeur du nom de l'objet dans la rubrique d'origine. Pour de plus amples informations sur la création de règles, veuillez consulter Règles pour AWS IoT (p. 394).

Utilisation des shadows sur les appareils

Cette section décrit les communications d'appareils avec des shadows à l'aide des messages MQTT, la méthode préférée qui permet aux appareils de communiquer avec le service AWS IoT Device Shadow.

Les communications de shadow émulent un modèle de demande/réponse à l'aide du modèle de communication de publication/abonnement de MQTT. Chaque action de shadow se compose d'une rubrique de demande, d'une rubrique de réponse réussie (`accepted`) et d'une rubrique de réponse d'erreur (`rejected`).

Si vous souhaitez que les applications et les services soient en mesure de déterminer si un appareil est connecté, veuillez consulter Détection d'un appareil connecté (p. 555).

> **Important**
>
> Comme MQTT utilise un modèle de communication de publication/abonnement, vous devez vous abonner aux rubriques de réponseAvantvous publiez une rubrique de demande. Dans le cas contraire, vous risquez de ne pas recevoir la réponse à la demande que vous publiez.
> Si vous utilisez un AWS IoT Device SDK (p. 1140)pour appeler les API du service Device Shadow, cela est géré pour vous.

Les exemples de cette section utilisent une forme abrégée de la rubrique où l'objet*`ShadowTopicPrefix`*peut faire référence à une ombre nommée ou sans nom, comme décrit dans ce tableau.

Les shadows peuvent être nommés ou non (classique). Les rubriques utilisées par chacun d'eux ne diffèrent que par le préfixe de rubrique. Ce tableau indique le préfixe de rubrique utilisé par chaque type de shadow.

Valeur ***`ShadowTopicPrefix`***	Type de shadow
`$aws/things/`*`thingName`*`/shadow`	Shadow non nommé (classique)

Valeur *ShadowTopicPrefix*	Type de shadow
`$aws/things/`*thingName*`/shadow/name/`*shadowName*	Shadow nommé

> **Important**
>
> Assurez-vous que l'utilisation des shadows par votre application ou service est cohérente et prise en charge par les implémentations correspondantes sur vos appareils. Par exemple, prenez en compte la façon dont les shadows sont créés, mis à jour et supprimés. Prenez également en compte la manière dont les mises à jour sont gérées sur l'appareil et dans les applications ou services qui accèdent à l'appareil via un shadow. Votre conception doit indiquer clairement comment l'état de l'appareil est mis à jour et rapporté, et comment vos applications et services interagissent avec l'appareil et ses shadows.

Pour créer une rubrique complète, sélectionnez *ShadowTopicPrefix* pour le type de shadow auquel vous souhaitez faire référence, remplacez *thingName*, et *shadowName* le cas échéant, par leurs valeurs correspondantes, puis ajoutez cela au stub de rubrique comme indiqué dans le tableau suivant. N'oubliez pas que les rubriques sont sensibles à la casse.

Veuillez consulter Rubriques de shadow (p. 103) pour de plus amples informations sur les rubriques réservées pour les shadows.

Initialisation de l'appareil lors de la première connexion à AWS IoT

Une fois qu'un appareil s'est enregistré auprès d'AWS IoT, il doit s'abonner à ces messages MQTT pour les shadows qu'il prend en charge.

Sujet	Signification	Action qu'un appareil doit effectuer lors de la réception de cette rubrique
ShadowTopicPrefix`/delete/accepted`	La demande `delete` a été acceptée et AWS IoT a supprimé le shadow.	Actions nécessaires pour accompagner la suppression du shadow, telles que l'arrêt de la publication des mises à jour.
ShadowTopicPrefix`/delete/rejected`	La demande `delete` a été rejetée par AWS IoT et le shadow n'a pas été supprimé. Le corps du message contient les informations d'erreur.	Répondre au message d'erreur dans le corps du message.
ShadowTopicPrefix`/get/accepted`	La demande `get` a été acceptée par AWS IoT, et le corps du message contient le document shadow actuel.	Actions nécessaires pour traiter le document d'état dans le corps du message.
ShadowTopicPrefix`/get/rejected`	La demande `get` a été rejetée par AWS IoT, et le corps du message contient les informations d'erreur.	Répondre au message d'erreur dans le corps du message.
ShadowTopicPrefix`/update/accepted`	La demande `update` a été acceptée par AWS IoT, et le	Confirmer que les données mises à jour dans le corps du

Sujet	Signification	Action qu'un appareil doit effectuer lors de la réception de cette rubrique
	corps du message contient le document shadow actuel.	message correspondent à l'état de l'appareil.
ShadowTopicPrefix/update/rejected	La demande `update` a été rejetée par AWS IoT, et le corps du message contient les informations d'erreur.	Répondre au message d'erreur dans le corps du message.
ShadowTopicPrefix/update/delta	Le document shadow a été mis à jour par une demande adressée à AWS IoT, et le corps du message contient les modifications demandées.	Mettre à jour l'état de l'appareil pour qu'il corresponde à l'état souhaité dans le corps du message.
ShadowTopicPrefix/update/documents	Une mise à jour du shadow a été récemment réalisée et le corps du message contient le document shadow actuel.	Confirmer que l'état mis à jour dans le corps du message correspond à l'état de l'appareil.

Après s'être abonné aux messages du tableau précédent pour chaque shadow, l'appareil doit tester si les shadows qu'il prend en charge ont déjà été créés en publiant une rubrique `/get` sur chaque shadow. Si un message `/get/accepted` est reçu, le corps du message contient le document shadow que l'appareil peut utiliser pour initialiser son état. Si un message `/get/rejected` est reçu, le shadow doit être créé en publiant un message `/update` avec l'état actuel de l'appareil.

Traitement des messages lorsque l'appareil est connecté à AWS IoT

Pendant qu'un appareil est connecté àAWS IoT, il peut recevoir/update/deltaet doit conserver l'état de l'appareil correspondant aux modifications apportées dans ses shadows par :

1. Lisant tous les messages /update/delta reçus et en synchronisant l'état de l'appareil pour qu'il y corresponde.
2. Publiant un message /update avec un corps de message `reported` doté de l'état actuel de l'appareil, chaque fois que l'état de l'appareil change.

Quand un appareil est connecté, il doit publier ces messages lorsque cela est indiqué.

Indication	Sujet	Charge utile
L'état de l'appareil a changé.	*ShadowTopicPrefix*/update	Un document shadow avec la propriété `reported`.
L'appareil peut ne pas être synchronisé avec le shadow.	*ShadowTopicPrefix*/get	(empty)
Une action sur l'appareil indique qu'un shadow ne sera plus pris en charge par l'appareil, par exemple lors de la suppression ou du remplacement de l'appareil	*ShadowTopicPrefix*/delete	(empty)

Traitement des messages lorsque l'appareil se reconnecte à AWS IoT

Lorsqu'un appareil avec un ou plusieurs shadows se connecte à AWS IoT, il doit synchroniser son état avec celui de tous les shadows qu'il prend en charge en :

1. Lisant tous les messages /update/delta reçus et en synchronisant l'état de l'appareil pour qu'il y corresponde.
2. Publiant un message /update avec un corps de message `reported` doté de l'état actuel de l'appareil.

Utilisation des shadows dans les applications et les services

Cette section décrit comment une application ou un service interagit avec le service AWS IoT Device Shadow. Cet exemple suppose que l'application ou le service interagit uniquement avec le shadow et, via le shadow, avec l'appareil. Cet exemple n'inclut aucune action de gestion, telle que la création ou la suppression de shadows.

Cet exemple utilise l'API REST du service AWS IoT Device Shadow pour interagir avec les shadows. Contrairement à l'exemple utilisé dans Utilisation des shadows sur les appareils (p. 551), qui utilise un modèle de communication de publication/abonnement, cet exemple utilise le modèle de communication demande/réponse de l'API REST. Cela signifie que l'application ou le service doit faire une demande avant de pouvoir recevoir une réponse de AWS IoT. Un inconvénient de ce modèle, toutefois, est qu'il ne prend pas en charge les notifications. Si votre application ou service nécessite des notifications rapides des changements d'état de l'appareil, prenez en compte les protocoles MQTT ou MQTT via WSS, qui prennent en charge le modèle de communication de publication/abonnement, comme décrit dans Utilisation des shadows sur les appareils (p. 551).

> **Important**
>
> Assurez-vous que l'utilisation des shadows par votre application ou service est cohérente et prise en charge par les implémentations correspondantes sur vos appareils. Prenez en compte, par exemple, la façon dont les shadows sont créés, mis à jour et supprimés, et la manière dont les mises à jour sont gérées sur l'appareil et dans les applications ou services qui accèdent au shadow. Votre conception doit indiquer clairement comment l'état de l'appareil est mis à jour et rapporté, et comment vos applications et services interagissent avec l'appareil et ses shadows.

L'URL de l'API REST pour un shadow nommé est :

```
https://endpoint:8443/things/thingName/shadow?name=shadowName
```

et pour un shadow non nommé :

```
https://endpoint:8443/things/thingName/shadow
```

où :

endpoint

Point de terminaison renvoyé par la commande CLI :

```
aws iot describe-endpoint --endpoint-type IOT:Data-ATS
```

thingName

Nom de l'objet d'objet auquel le shadow appartient

shadowName

Nom du shadow nommé. Ce paramètre n'est pas utilisé avec des shadows non nommés.

Initialisation de l'application ou du service lors de la connexion à AWS IoT

Lorsque l'application se connecte pour la première fois à AWS IoT, elle doit envoyer une demande HTTP GET aux URL des shadows qu'elle utilise pour obtenir l'état actuel de ces shadows. Cela lui permet de synchroniser l'application ou le service avec le shadow.

Traitement des changements d'état lorsque l'application ou le service sont connectés à AWS IoT

Lorsque l'application ou le service sont connectés à AWS IoT, ils peuvent interroger périodiquement l'état actuel en envoyant une demande HTTP GET sur les URL des shadows qu'ils utilisent.

Lorsqu'un utilisateur final interagit avec l'application ou le service pour modifier l'état de l'appareil, l'application ou le service peuvent envoyer une demande HTTP POST aux URL des shadows qu'ils utilisent pour mettre à jour l'état `desired` du shadow. Cette demande renvoie la modification qui a été acceptée, mais vous devrez peut-être interroger le shadow en effectuant des demandes HTTP GET jusqu'à ce que l'appareil ait mis à jour le shadow avec son nouvel état.

Détection d'un appareil connecté

Pour déterminer si un appareil est actuellement connecté, incluez une propriété `connected` dans le document shadow et utilisez un message MQTT Last Will and Testament (LWT) pour définir la propriété `connected` sur `false` si un appareil est déconnecté en raison d'une erreur.

Note

Les messages MQTT LWT envoyés à des rubriques réservées AWS IoT (rubriques commençant par $) sont ignorés par le service AWS IoT Device Shadow. Toutefois, ils sont traités par les clients abonnés et par le moteur de règles AWS IoT. Vous devrez donc créer un message LWT à envoyer à une rubrique non réservée et une règle qui republiera le message MQTT LWT en tant que message de mise à jour de shadow vers la rubrique de mise à jour réservée du shadow, *`ShadowTopicPrefix`*`/update`.

Pour envoyer un message LWT au service Device Shadow

1. Créez une règle qui republie le message MQTT LWT sur la rubrique réservée. L'exemple suivant est une règle à l'écoute d'un message sur la rubrique `my/things/myLightBulb/update` et qui le republie dans `$aws/things/myLightBulb/shadow/update`.

```
{
    "rule": {
    "ruleDisabled": false,
    "sql": "SELECT * FROM 'my/things/myLightBulb/update'",
    "description": "Turn my/things/ into $aws/things/",
    "actions": [{
        "republish": {
            "topic": "$$aws/things/myLightBulb/shadow/update",
```

```
            "roleArn": "arn:aws:iam:123456789012:role/aws_iot_republish"
        }
    }]
    }
}
```

2. Lorsque l'appareil se connecte à AWS IoT, il enregistre un message LWT dans une rubrique non réservée pour que la règle de republication le reconnaisse. Dans cet exemple, cette rubrique est `my/things/myLightBulb/update` et elle définit la propriété connectée sur `false`.

```
{
    "state": {
        "reported": {
            "connected":"false"
        }
    }
}
```

3. Après la connexion, l'appareil publie un message sur sa rubrique de mise à jour de shadow, `$aws/things/myLightBulb/shadow/update`, pour rapporter son état actuel, ce qui inclut la définition de sa propriété `connected` sur `true`.

```
{
     "state": {
        "reported": {
            "connected":"true"
        }
    }
}
```

4. Avant que l'appareil ne se déconnecte gracieusement, il publie un message sur sa rubrique de mise à jour de shadow, `$aws/things/myLightBulb/shadow/update`, pour rapporter son état le plus récent, ce qui inclut la définition de sa propriété `connected` sur `false`.

```
{
    "state": {
        "reported": {
            "connected":"false"
        }
    }
}
```

5. Si l'appareil se déconnecte en raison d'une erreur, l'agent de messages AWS IoT publie le message LWT de l'appareil au nom de l'appareil. La règle de republication détecte ce message et publie le message de mise à jour de shadow pour mettre à jour la propriété `connected` du shadow de l'appareil.

Simulation des communications du service Device Shadow

Cette rubrique indique comment le service Device Shadow agit en tant qu'intermédiaire et permet aux appareils et aux applications d'utiliser un shadow pour mettre à jour, stocker et récupérer l'état d'un appareil.

Pour illustrer l'interaction décrite dans cette rubrique et pour l'explorer plus loin, vous aurez besoin d'un Compte AWS et un système sur lequel vous pouvez exécuter l'AWS CLI. Si vous ne les avez pas, vous pouvez toujours voir l'interaction dans les exemples de code.

Dans cet exemple, la console AWS IoT représente l'appareil. L'interface AWS CLI représente l'application ou le service qui accède à l'appareil par l'intermédiaire du shadow. L'interface AWS CLI est très similaire à l'API qu'une application peut utiliser pour communiquer avec AWS IoT. L'appareil dans cet exemple est une ampoule intelligente et l'application affiche l'état de cette ampoule et peut le modifier.

Configuration de la simulation

Ces procédures initialisent la simulation en ouvrant la console AWS IoT, qui simule votre appareil, et la fenêtre de ligne de commande qui simule votre application.

Pour configurer votre environnement de simulation

1. Création d'un Compte AWS Si vous en possédez déjà un pour cette simulation, vous pouvez ignorer cette étape.

   Vous aurez besoin d'un Compte AWS Pour exécuter vous-même les exemples de cette rubrique. Si vous n'avez pas de Compte AWS , créez-en un, comme décrit dansConfigurer votre Compte AWS (p. 19).
2. Ouvrez la console AWS IoT et, dans le menu de gauche, choisissez Test pour ouvrir le client MQTT.
3. Dans une autre fenêtre, ouvrez une fenêtre de terminal sur un système où l'interface AWS CLI est installée.

Deux fenêtres doivent être ouvertes : une avec la console AWS IoT sur la page Test et l'autre avec une invite de ligne de commande.

Initialisation de l'appareil

Dans cette simulation, nous allons travailler avec un objet d'objet nommé, mySimulatedThing, et son shadow nommé, simShadow1.

Créer un objet chose et son ombre nommée

Pour créer un objet d'objet, dans laAWS IoTConsole :

1. ChoisissezGérer, puis choisissezObjets.
2. Cliquez surCréersi les choses sont répertoriées sinon cliquez surEnregistrer un objet unique>Pour créer un objet uniqueAWS IoTUn objet.
3. Saisissez le nom`mySimulatedThing`, laissez les autres paramètres par défaut, puis cliquez surSuivant.
4. Utilisez la création d'un certificat en un clic pour générer les certificats qui authentifieront la connexion du périphérique àAWS IoT. Cliquez surActivatepour activer le certificat.
5. Vous pouvez attacher la stratégie`My_IoT_Policy`qui donnerait à l'appareil l'autorisation de publier et de s'abonner aux rubriques réservées MQTT. Pour des étapes plus détaillées sur la création d'unAWS IoTet comment créer cette stratégie, voirCréer un objet d'objet (p. 37).

Par défaut, chaqueAWS IoTobjet chose a une seule ombre classique. Vous pouvez créer une ombre en publiant une demande de mise à jour dans la rubrique`$aws/things/mySimulatedThing/shadow/name/simShadow1/update`comme décrit ci-dessous, Alternativement, dans leAWS IoTConsole, choisissez votre objet truc dans la liste des objets affichés, puis choisissezShadows. ChoisissezAjouter une shadow, entrez le nom`simShadow1`, puis choisissezCréerPour ajouter l'shadow nommé.

Abonnez-vous et publiez sur des sujets MQTT réservés

Dans la console, abonnez-vous aux rubriques shadow MQTT réservées. Ces rubriques sont les réponses aux actions `get`, `update` et `delete`, afin que votre appareil soit prêt à recevoir les réponses après avoir publié une action.

Pour vous abonner à une rubrique MQTT dans le client MQTT

1. DansClient MQTT, choisissezS'abonner à la rubrique.
2. Saisissez la`get`,`update`, et`delete`sujets auxquels vous souhaitez vous abonner. Copiez une rubrique à la fois à partir de la liste suivante, collez-la dans leFiltre de rubriques, puis cliquez surS'abonner. Vous devez voir ce qui suit s'afficher sousSubscriptions.

   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/delete/accepted`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/delete/rejected`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/delta`
   - `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/documents`

   À ce stade, votre appareil simulé est prêt à recevoir les rubriques telles qu'elles sont publiées par AWS IoT.

Pour publier dans une rubrique MQTT dans laClient MQTT

Une fois qu'un appareil s'est initialisé et s'est abonné aux rubriques de réponse, il doit exécuter une requête portant sur les shadows qu'il prend en charge. Cette simulation prend en charge un seul shadow, le shadow qui prend en charge un objet d'objet nommé, mySimulatedThing, nommé simShadow1.

Pour obtenir l'état actuel du shadow à partir du client MQTT

1. Dans le client MQTT, choisissez Publier dans une rubrique.
2. UNDERPublier, entrez la rubrique suivante et supprimez tout contenu de la fenêtre de corps de message ci-dessous où vous avez entré la rubrique à obtenir. Vous pouvez alors choisirPublier dans la rubriquePour publier la demande.`$aws/things/mySimulatedThing/shadow/name/simShadow1/get`.

   Si vous n'avez pas créé l'shadow nommé,`simShadow1`, vous recevez un message dans le`$aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected`et la rubrique`code`est`404`Comme dans cet exemple car l'shadow n'a pas été créé, nous le créerons ensuite.

   ```
   {
     "code": 404,
     "message": "No shadow exists with name: 'simShadow1'"
   }
   ```

Pour créer un shadow avec l'état actuel de l'appareil

1. DansClient MQTT, choisissezPublier dans une rubriqueet entrez cette rubrique :

   ```
   $aws/things/mySimulatedThing/shadow/name/simShadow1/update
   ```

2. Dans la fenêtre de corps de message ci-dessous où vous avez entré la rubrique, entrez ce document shadow pour montrer que l'appareil rapporte son ID et sa couleur actuelle en valeurs RVB. ChoisissezPublierPour publier la demande.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
    }
  },
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Si vous recevez un message dans la rubrique :

- `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted` : Cela signifie que l'shadow a été créé et que le corps du message contient le document shadow actuel.
- `$aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected` : passez en revue l'erreur dans le corps du message.
- `$aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted` : L'shadow existe déjà et le corps du message a l'état actuel de shadow, comme dans cet exemple. Avec cela, vous pouvez définir votre appareil ou confirmer qu'il correspond à l'état du shadow.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
    }
  },
  "metadata": {
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        }
      ]
    }
  },
  "version": 3,
  "timestamp": 1591140517,
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
```

```
}
```

Envoi d'une mise à jour à partir de l'application

Cette section utilise l'interface AWS CLI pour montrer comment une application peut interagir avec un shadow.

Pour obtenir l'état actuel du shadow à l'aide de l'interface AWS CLI

Sur la ligne de commande, entrez la commande ci-dessous.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /dev/
stdout
```

Sur les plates-formes Windows, vous pouvez également utiliser`con`Au lieu de`/dev/stdout`.

```
aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 con
```

Comme le shadow existe et a été initialisé par l'appareil pour refléter son état actuel, il doit renvoyer le document shadow suivant.

```
{
  "state": {
    "reported": {
      "ID": "SmartLamp21",
      "ColorRGB": [
        128,
        128,
        128
      ]
    }
  },
  "metadata": {
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        },
        {
          "timestamp": 1591140517
        }
      ]
    }
  },
  "version": 3,
  "timestamp": 1591141111
}
```

L'application peut utiliser cette réponse pour initialiser sa représentation de l'état de l'appareil.

Si l'application met à jour l'état, par exemple lorsqu'un utilisateur final change la couleur de notre ampoule intelligente en jaune, l'application enverra une commande update-thing-shadow. Cette commande correspond à l'API REST `UpdateThingShadow`.

Pour mettre à jour un shadow à partir d'une application

Sur la ligne de commande, entrez la commande ci-dessous.

AWS CLI v2.x

```
aws iot-data update-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
 \
    --cli-binary-format raw-in-base64-out \
    --payload '{"state":{"desired":{"ColorRGB":[255,255,0]}},"clientToken":"21b21b21-
bfd2-4279-8c65-e2f697ff4fab"}' /dev/stdout
```

AWS CLI v1.x

```
aws iot-data update-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
 \
    --payload '{"state":{"desired":{"ColorRGB":[255,255,0]}},"clientToken":"21b21b21-
bfd2-4279-8c65-e2f697ff4fab"}' /dev/stdout
```

Si elle réussit, cette commande doit renvoyer le document shadow suivant.

```
{
  "state": {
    "desired": {
      "ColorRGB": [
        255,
        255,
        0
      ]
    }
  },
  "metadata": {
    "desired": {
      "ColorRGB": [
        {
          "timestamp": 1591141596
        },
        {
          "timestamp": 1591141596
        },
        {
          "timestamp": 1591141596
        }
      ]
    }
  },
  "version": 4,
  "timestamp": 1591141596,
  "clientToken": "21b21b21-bfd2-4279-8c65-e2f697ff4fab"
}
```

Réponse à une mise à jour sur l'appareil

En revenant au client MQTT dans la console AWS, vous devriez voir les messages publiés par AWS IoT pour refléter la commande de mise à jour émise dans la section précédente.

Pour afficher les messages de mise à jour dans le client MQTT

Dans le client MQTT, choisissez $aws/things/mySimulatedThing/shadow/name/simShadow1/update/delta dans la colonne Abonnements. Si le nom de la rubrique est tronqué, vous pouvez faire une pause

dessus pour voir le nom complet. Dans le journal des rubriques de cette rubrique, vous devriez voir un`/delta`Message similaire à celui-ci.

```
{
  "version": 4,
  "timestamp": 1591141596,
  "state": {
    "ColorRGB": [
      255,
      255,
      0
    ]
  },
  "metadata": {
    "ColorRGB": [
      {
        "timestamp": 1591141596
      },
      {
        "timestamp": 1591141596
      },
      {
        "timestamp": 1591141596
      }
    ]
  },
  "clientToken": "21b21b21-bfd2-4279-8c65-e2f697ff4fab"
}
```

Votre appareil traite le contenu de ce message pour définir l'état de l'appareil, afin qu'il corresponde à l'état `desired` dans le message.

Une fois que l'appareil a mis à jour l'état pour qu'il corresponde à l'état `desired` dans le message, il doit renvoyer le nouvel état rapporté à AWS IoT en publiant un message de mise à jour. Cette procédure simule cela dans le client MQTT.

Pour mettre à jour le shadow à partir de l'appareil

1. Dans le client MQTT, choisissez Publier dans une rubrique.
2. Dans la fenêtre de corps de message, dans le champ de rubrique au-dessus de la fenêtre de corps de message, entrez la rubrique de l'shadow suivie de la rubrique`/updated`' :`$aws/things/mySimulatedThing/shadow/name/simShadow1/update`Dans le corps du message, entrez ce document shadow mis à jour, qui décrit l'état actuel de l'appareil. Cliquez surPublierPour publier l'état de l'appareil mis à jour.

   ```
   {
     "state": {
       "reported": {
         "ColorRGB": [255,255,0]
         }
     },
     "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
   }
   ```

   Si le message a été reçu avec succès par AWS IoT, vous devriez voir une nouvelle réponse dans le journal de messages $aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted dans le client MQTT avec l'état actuel du shadow, comme dans cet exemple.

   ```
   {
     "state": {
   ```

```
    "reported": {
      "ColorRGB": [
        255,
        255,
        0
      ]
    }
  },
  "metadata": {
    "reported": {
      "ColorRGB": [
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        }
      ]
    }
  },
  "version": 5,
  "timestamp": 1591142747,
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
}
```

Une mise à jour réussie de l'état rapporté de l'appareil entraîne également l'envoi par AWS IoT d'une description complète de l'état du shadow dans un message à la rubrique , telle que ce corps de message obtenu à partir de la mise à jour du shadow effectuée par l'appareil dans le cadre de la procédure précédente.

```
{
  "previous": {
    "state": {
      "desired": {
        "ColorRGB": [
          255,
          255,
          0
        ]
      },
      "reported": {
        "ID": "SmartLamp21",
        "ColorRGB": [
          128,
          128,
          128
        ]
      }
    },
    "metadata": {
      "desired": {
        "ColorRGB": [
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
```

```
          }
        ]
      },
      "reported": {
        "ID": {
          "timestamp": 1591140517
        },
        "ColorRGB": [
          {
            "timestamp": 1591140517
          },
          {
            "timestamp": 1591140517
          },
          {
            "timestamp": 1591140517
          }
        ]
      }
    },
    "version": 4
  },
  "current": {
    "state": {
      "desired": {
        "ColorRGB": [
          255,
          255,
          0
        ]
      },
      "reported": {
        "ID": "SmartLamp21",
        "ColorRGB": [
          255,
          255,
          0
        ]
      }
    },
    "metadata": {
      "desired": {
        "ColorRGB": [
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          },
          {
            "timestamp": 1591141596
          }
        ]
      },
      "reported": {
        "ID": {
          "timestamp": 1591140517
        },
        "ColorRGB": [
          {
            "timestamp": 1591142747
          },
          {
            "timestamp": 1591142747
          },
          {
```

```
            "timestamp": 1591142747
          }
        ]
      }
    },
    "version": 5
  },
  "timestamp": 1591142747,
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"
}
```

Observation de la mise à jour dans l'application

L'application peut désormais interroger le shadow pour obtenir l'état actuel, tel qu'il a été rapporté par l'appareil.

Pour obtenir l'état actuel du shadow à l'aide de l'interface AWS CLI

1. Sur la ligne de commande, entrez la commande ci-dessous.

   ```
   aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1 /
   dev/stdout
   ```

   Sur les plates-formes Windows, vous pouvez également utiliser `con` Au lieu de `/dev/stdout`.

   ```
   aws iot-data get-thing-shadow --thing-name mySimulatedThing --shadow-name simShadow1
    con
   ```

2. Comme le shadow vient d'être mis à jour par l'appareil pour refléter son état actuel, il doit renvoyer le document shadow suivant.

   ```
   {
     "state": {
       "desired": {
         "ColorRGB": [
           255,
           255,
           0
         ]
       },
       "reported": {
         "ID": "SmartLamp21",
         "ColorRGB": [
           255,
           255,
           0
         ]
       }
     },
     "metadata": {
       "desired": {
         "ColorRGB": [
           {
             "timestamp": 1591141596
           },
           {
             "timestamp": 1591141596
           },
           {
             "timestamp": 1591141596
           }
   ```

```
      ]
    },
    "reported": {
      "ID": {
        "timestamp": 1591140517
      },
      "ColorRGB": [
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        },
        {
          "timestamp": 1591142747
        }
      ]
    }
  },
  "version": 5,
  "timestamp": 1591143269
}
```

Au-delà de la simulation

Expérimentez avec l'interaction entreAWS CLI(représentant l'application) et la console (représentant l'appareil) pour modéliser votre solution IoT.

Interaction avec les shadows

Cette rubrique décrit les messages associés à chacune des trois méthodes fournies par AWS IoT qui permettent de travailler avec les shadows. Il s'agit notamment des méthodes suivantes :

`UPDATE`

Crée un shadow s'il n'existe pas ou met à jour le contenu d'un shadow existant avec les informations d'état fournies dans le corps de message. AWS IoT enregistre un horodatage avec chaque mise à jour pour indiquer quand l'état a été mis à jour pour la dernière fois. Lorsque l'état du shadow change, AWS IoT envoie des messages `/delta` à tous les abonnés MQTT avec la différence entre les états `desired` et `reported`. Les appareils ou les applications qui reçoivent un message `/delta` peuvent effectuer des actions en fonction de cette différence. Par exemple, un appareil peut mettre à jour son état à l'état souhaité, ou une application peut mettre à jour son interface utilisateur pour refléter le changement d'état de l'appareil.

`GET`

Récupère un document shadow actuel qui contient l'état complet du shadow, y compris les métadonnées.

`DELETE`

Supprime le shadow et tout son contenu. Vous ne pouvez pas restaurer un shadow supprimé, mais vous pouvez créer un nouveau shadow portant le même nom. Notez que la suppression d'une ombre ne réinitialise pas son numéro de version à 0.

Support du protocole

AWS IoT prend en charge MQTT et une API REST via les protocoles HTTPS pour interagir avec les shadows. AWS IoT fournit un ensemble de rubriques de demande et de réponse réservées pour les actions MQTT de publication et d'abonnement. Les appareils et les applications doivent s'abonner aux rubriques de réponse avant de publier une rubrique de demande pour obtenir des informations sur la façon dont AWS IoT a traité la demande. Pour plus d'informations, consultez Rubriques MQTT de Device Shadow (p. 577) et API REST Device Shadow (p. 573).

Demande d'état et génération de rapport d'état

Lorsque vous concevez une solution IoT à l'aide d'AWS IoT et de shadows, vous devez déterminer les applications et appareils qui demanderont les modifications et ceux qui les implémenteront. Généralement, un appareil implémente les modifications et les rapporte au shadow, et les applications et services y répondent et demandent les modifications dans le shadow. Votre solution peut être différente, mais les exemples de cette rubrique supposent que l'application ou le service client demande les modifications dans le shadow et que l'appareil effectue ces modifications et les rapporte en retour au shadow.

Mise à jour d'un shadow

Votre application ou service peut mettre à jour l'état d'un shadow à l'aide de l'API UpdateThingShadow (p. 574) ou en publiant dans la rubrique /update (p. 579). Les mises à jour concernent uniquement les champs spécifiés dans la demande.

Mise à jour d'un shadow lorsqu'un client demande un changement d'état

Quand un client demande un changement d'état dans un shadow à l'aide du protocole MQTT

1. Le client doit disposer d'un document shadow actuel pour pouvoir identifier les propriétés à modifier. Veuillez consulter l'action /get pour voir comment obtenir le document shadow actuel.
2. Le client s'abonne aux rubriques MQTT suivantes :

   - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/update/rejected`
   - `$aws/things/thingName/shadow/name/shadowName/update/delta`
   - `$aws/things/thingName/shadow/name/shadowName/update/documents`
3. Le client publie une rubrique de demande `$aws/things/thingName/shadow/name/shadowName/update` avec un document d'état qui contient l'état souhaité du shadow. Seules les propriétés à modifier doivent être incluses dans ce document. Ceci est un exemple de document avec l'état souhaité.

   ```
   {
     "state": {
       "desired": {
         "color": {
           "r": 10
         },
         "engine": "ON"
       }
     }
   }
   ```
4. Si la demande de mise à jour est valide, AWS IoT met à jour l'état souhaité dans le shadow et publie des messages sur les rubriques suivantes :

- `$aws/things/thingName/shadow/name/shadowName/update/accepted`
- `$aws/things/thingName/shadow/name/shadowName/update/delta`

Le message `/update/accepted` contient un document shadow /document d'état de la réponse accepté (p. 586) et le message `/update/delta` contient un document shadow /documents d'état de la réponse delta (p. 587).

5. Si la demande de mise à jour n'est pas valide, AWS IoT publie un message avec la rubrique `$aws/things/thingName/shadow/name/shadowName/update/rejected` avec un document shadow Document de réponse d'erreur (p. 589) qui décrit l'erreur.

Quand un client demande un changement d'état dans un shadow à l'aide de l'API

1. Le client appelle l'API `UpdateThingShadow (p. 574)` avec un document d'état Document d'état de demande (p. 586) comme corps de message.
2. Si la demande était valide, AWS IoT renvoie un code de réponse de succès HTTP et un document shadow /document d'état de la réponse accepté (p. 586) comme corps de message de réponse.

   AWS IoT publiera également un message MQTT dans la rubrique `$aws/things/thingName/shadow/name/shadowName/update/delta` avec un document shadow /documents d'état de la réponse delta (p. 587) pour tous les appareils ou clients qui s'y abonnent.
3. Si la demande n'était pas valide, AWS IoT renvoie un code de réponse d'erreur HTTP et un Document de réponse d'erreur (p. 589) comme corps de message de réponse.

Lorsque l'appareil reçoit l'état `/desired` sur la rubrique `/update/delta`, il effectue les modifications souhaitées sur l'appareil. Il envoie ensuite un message à la rubrique `/update` pour rapporter son état actuel au shadow.

Mise à jour d'un shadow lorsqu'un appareil rapporte son état actuel

Quand un appareil rapporte son état actuel au shadow à l'aide du protocole MQTT

1. L'appareil doit s'abonner aux rubriques MQTT suivantes avant de mettre à jour le shadow :

   - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/update/rejected`
   - `$aws/things/thingName/shadow/name/shadowName/update/delta`
   - `$aws/things/thingName/shadow/name/shadowName/update/documents`
2. L'appareil rapporte son état actuel en publiant un message dans la rubrique `$aws/things/thingName/shadow/name/shadowName/update` qui rapporte l'état actuel, comme dans cet exemple.

```
{
    "state": {
        "reported" : {
            "color" : { "r" : 10 },
            "engine" : "ON"
        }
    }
}
```

3. Si AWS IoT accepte la mise à jour, il publie un message adressé aux rubriques `$aws/things/thingName/shadow/name/shadowName/update/accepted` avec un document shadow /document d'état de la réponse accepté (p. 586).
4. Si la demande de mise à jour n'est pas valide, AWS IoT publie un message avec la rubrique `$aws/things/thingName/shadow/name/shadowName/update/rejected` avec un document shadow Document de réponse d'erreur (p. 589) qui décrit l'erreur.

Quand un appareil rapporte son état actuel au shadow à l'aide de l'API

1. L'appareil appelle l'API `UpdateThingShadow (p. 574)` avec un document d'état Document d'état de demande (p. 586) comme corps de message.
2. Si la demande était valide, AWS IoT met à jour le shadow et renvoie un code de réponse de succès HTTP avec un document shadow /document d'état de la réponse accepté (p. 586) comme corps de message de réponse.

   AWS IoT publiera également un message MQTT dans la rubrique `$aws/things/thingName/shadow/name/shadowName/update/delta` avec un document shadow /documents d'état de la réponse delta (p. 587) pour tous les appareils ou clients qui s'y abonnent.
3. Si la demande n'était pas valide, AWS IoT renvoie un code de réponse d'erreur HTTP et un Document de réponse d'erreur (p. 589) comme corps de message de réponse.

Verrouillage optimiste

Vous pouvez utiliser la version du document d'état pour vous assurer que vous mettez à jour la version la plus récente d'un document shadow d'appareil. Lorsque vous fournissez une version dans une demande de mise à jour, le service rejette la demande avec un code de réponse de conflit HTTP 409 si la version actuelle du document d'état ne correspond pas à la version fournie.

Exemples :

Document initial :

```
{
  "state": {
    "desired": {
      "colors": [
        "RED",
        "GREEN",
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

Mise à jour : (la version ne correspond pas ; cette demande est rejetée)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 9
}
```

Résultat:

```
{
  "code": 409,
  "message": "Version conflict",
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"
}
```

Mise à jour : (la version correspond ; cette demande est acceptée)

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 10
}
```

État final :

```
{
  "state": {
    "desired": {
      "colors": [
        "BLUE"
      ]
    }
  },
  "version": 11
}
```

Récupération d'un document Shadow

Vous pouvez récupérer un document shadow à l'aide de l'API GetThingShadow (p. 573) ou en vous abonnant et en publiant dans la rubrique /get (p. 578). Ceci récupère un document shadow complet, y compris tout delta entre les états `desired` et `reported`. La procédure pour cette tâche est la même que l'appareil ou un client effectue la demande.

Pour récupérer un document shadow à l'aide du protocole MQTT

1. L'appareil ou le client doit s'abonner à ces rubriques MQTT avant de mettre à jour le shadow :
   - `$aws/things/thingName/shadow/name/shadowName/get/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/get/rejected`
2. L'appareil ou le client publie un message dans la rubrique `$aws/things/thingName/shadow/name/shadowName/get` avec un corps de message vide.
3. Si la demande réussit, AWS IoT publie un message dans la rubrique `$aws/things/thingName/shadow/name/shadowName/get/accepted` avec un /document d'état de la réponse accepté (p. 586) dans le corps de message.
4. Si la demande n'était pas valide, AWS IoT publie un message dans la rubrique `$aws/things/thingName/shadow/name/shadowName/get/rejected` avec un Document de réponse d'erreur (p. 589) dans le corps du message.

Pour récupérer un document shadow à l'aide d'une API REST

1. L'appareil ou le client appelle l'API `GetThingShadow (p. 573)` avec un corps de message vide.
2. Si la demande est valide, AWS IoT renvoie un code de réponse de succès HTTP avec un document shadow /document d'état de la réponse accepté (p. 586) comme corps de message de réponse.
3. Si la demande n'est pas valide, AWS IoT renvoie un code de réponse d'erreur HTTP et un Document de réponse d'erreur (p. 589) comme corps de message de réponse.

Suppression de données shadow

Il existe deux façons de supprimer des données shadow : vous pouvez supprimer les propriétés spécifiques dans le document shadow et vous pouvez supprimer complètement le shadow.

- Pour supprimer des propriétés spécifiques d'un shadow, mettez à jour le shadow. Toutefois, définissez la valeur des propriétés à supprimer sur `null`. Les champs dotés d'une valeur `null` sont supprimés du document shadow.
- Pour supprimer le shadow entier, utilisez l'API DeleteThingShadow (p. 575) ou publiez dans la rubrique /supprimer (p. 583).

Notez que la suppression d'une ombre ne réinitialise pas son numéro de version à 0.

Suppression d'une propriété dans un document shadow

Pour supprimer une propriété dans un shadow à l'aide du protocole MQTT

1. L'appareil ou le client doit disposer d'un document shadow actuel pour pouvoir identifier les propriétés à modifier. Veuillez consulter Récupération d'un document Shadow (p. 570) pour obtenir des informations sur la façon d'obtenir le document shadow actuel.
2. L'appareil ou le client s'abonne aux rubriques MQTT suivantes :
   - `$aws/things/thingName/shadow/name/shadowName/update/accepted`
   - `$aws/things/thingName/shadow/name/shadowName/update/rejected`
3. L'appareil ou le client publie une rubrique de demande `$aws/things/thingName/shadow/name/shadowName/update` avec un document d'état qui attribue des valeurs `null` aux propriétés du shadow à supprimer. Seules les propriétés à modifier doivent être incluses dans ce document. Voici un exemple de document qui supprime la propriété `engine`.

   ```
   {
     "state": {
       "desired": {
         "engine": null
       }
     }
   }
   ```
4. Si la demande de mise à jour est valide, AWS IoT supprime les propriétés spécifiées dans le shadow et publie un message avec la rubrique `$aws/things/thingName/shadow/name/shadowName/update/accepted` avec un document shadow /document d'état de la réponse accepté (p. 586) dans le corps du message.
5. Si la demande de mise à jour n'est pas valide, AWS IoT publie un message avec la rubrique `$aws/things/thingName/shadow/name/shadowName/update/rejected` avec un document shadow Document de réponse d'erreur (p. 589) qui décrit l'erreur.

Pour supprimer une propriété d'un shadow à l'aide de l'API REST

1. L'appareil ou le client appelle l'API `UpdateThingShadow (p. 574)` avec un Document d'état de demande (p. 586) qui attribue des valeurs `null` aux propriétés du shadow à supprimer. Incluez uniquement les propriétés que vous souhaitez supprimer dans le document. Voici un exemple de document qui supprime la propriété `engine`.

```
{
  "state": {
    "desired": {
      "engine": null
    }
  }
}
```

2. Si la demande était valide, AWS IoT renvoie un code de réponse de succès HTTP et un document shadow /document d'état de la réponse accepté (p. 586) comme corps de message de réponse.
3. Si la demande n'était pas valide, AWS IoT renvoie un code de réponse d'erreur HTTP et un Document de réponse d'erreur (p. 589) comme corps de message de réponse.

Suppression d'un shadow

Note

La définition de l'état de shadow de l'appareil sur `null` ne supprime pas le shadow. La version de shadow sera incrémentée lors de la prochaine mise à jour.
La suppression d'un shadow d'appareil ne supprime pas l'objet d'objet. La suppression d'un objet d'objet ne supprime pas le shadow d'appareil correspondant.
La suppression d'une ombre ne réinitialise pas son numéro de version à 0.

Pour supprimer un shadow à l'aide du protocole MQTT

1. L'appareil ou le client s'abonne aux rubriques MQTT suivantes :
   - `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/delete/accepted`
   - `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/delete/rejected`
2. L'appareil ou le client publie un `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/delete` avec un tampon de messages vide.
3. Si la demande de suppression est valide, AWS IoT supprime le shadow et publie un message avec la rubrique `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/delete/accepted` et un document shadow /document d'état de la réponse accepté (p. 586) abrégé dans le corps du message. Voici un exemple du message de suppression accepté :

```
{
  "version": 4,
  "timestamp": 1591057529
}
```

4. Si la demande de mise à jour n'est pas valide, AWS IoT publie un message avec la rubrique `$aws/things/`*`thingName`*`/shadow/name/`*`shadowName`*`/delete/rejected` avec un document shadow Document de réponse d'erreur (p. 589) qui décrit l'erreur.

Pour supprimer un shadow à l'aide de l'API REST

1. L'appareil ou le client appelle l'API `DeleteThingShadow (p. 575)` avec un tampon de messages vide.

2. Si la demande était valide, AWS IoT renvoie un code de réponse de succès HTTP, un /document d'état de la réponse accepté (p. 586) et un document shadow /document d'état de la réponse accepté (p. 586) abrégé dans le corps du message. Voici un exemple du message de suppression accepté :

```
{
  "version": 4,
  "timestamp": 1591057529
}
```

3. Si la demande n'était pas valide, AWS IoT renvoie un code de réponse d'erreur HTTP et un Document de réponse d'erreur (p. 589) comme corps de message de réponse.

API REST Device Shadow

Un shadow expose l'URI suivante pour mettre à jour les informations d'état :

```
https://endpoint:8443/things/thingName/shadow
```

Le point de terminaison est spécifique à votre Compte AWS . Pour rechercher votre point de terminaison, vous pouvez :

- Utilisation de l'describe-endpointdepuis laAWS CLI.
- Utilisation de l'AWS IoTparamètres de la console. DansParamètres, le point de terminaison est répertorié sousPoint de terminaison personnalisé
- Utilisation de l'AWS IoTdétails de la console. OuvrirGérer. UNDERGérer, choisissezObjetset, dans la liste des choses, sélectionnez et ouvrez une chose. Dans le navigation de gauche de la page de détails Thing, choisissezInteragiret affichez l'URI du point de terminaison dans laHTTPSde la page.

Le format du point de terminaison est le suivant :

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

L'API REST Shadow suit les mêmes protocoles HTTPS/mappages de ports que ceux décrits dans Protocoles de communication de périphérique (p. 79).

Actions d'API

- GetThingShadow (p. 573)
- UpdateThingShadow (p. 574)
- DeleteThingShadow (p. 575)
- ListNamedShadowsForThing (p. 576)

GetThingShadow

Obtient le shadow de l'objet spécifié.

Le document d'état de réponse comprend le delta entre les états `desired` et `reported`.

Request

La demande comprend les en-têtes HTTP standard, plus l'URI suivante :

```
HTTP GET https://endpoint:8443/things/thingName/shadow?name=shadowName
Request body: (none)
```

Le paramètre de requête `name` n'est pas requis pour les shadows non nommés (classiques).

Response

En cas de réussite, la réponse comprend les en-têtes HTTP standard, plus le code et le corps suivants :

```
HTTP 200
Response Body: response state document
```

Pour plus d'informations, consultez Exemple de document d'état de réponse (p. 586).

Authorization

La récupération d'un shadow nécessite une stratégie qui permet au mandataire de réaliser l'action `iot:GetThingShadow`. Le service Device Shadow accepte deux formes d'authentification : Signature Version 4 avec les informations d'identification IAM ou l'authentification mutuelle TLS avec un certificat client.

Voici un exemple de stratégie qui permet à un mandataire de récupérer un shadow d'appareil :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:GetThingShadow",
      "Resource": [
        "arn:aws:iot:region:account:thing/thing"
      ]
    }
  ]
}
```

UpdateThingShadow

Met à jour le shadow de l'objet spécifié.

Les mises à jour concernent uniquement les champs spécifiés dans le document d'état de la demande. Tout champ avec une valeur `null` est supprimé du shadow d'appareil.

Request

La demande comprend les en-têtes HTTP standard, plus l'URI et le corps suivants :

```
HTTP POST https://endpoint:8443/things/thingName/shadow?name=shadowName
Request body: request state document
```

Le paramètre de requête `name` n'est pas requis pour les shadows non nommés (classiques).

Pour plus d'informations, consultez Exemple de document d'état de la demande (p. 586).

Response

En cas de réussite, la réponse comprend les en-têtes HTTP standard, plus le code et le corps suivants :

```
HTTP 200
Response body: response state document
```

Pour plus d'informations, consultez Exemple de document d'état de réponse (p. 586).

Authorization

La mise à jour d'un shadow nécessite une stratégie qui permet au mandataire de réaliser l'action `iot:UpdateThingShadow`. Le service Device Shadow accepte deux formes d'authentification : Signature Version 4 avec les informations d'identification IAM ou l'authentification mutuelle TLS avec un certificat client.

Voici un exemple de stratégie qui permet à un mandataire de mettre à jour un shadow d'appareil :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:UpdateThingShadow",
      "Resource": [
        "arn:aws:iot:region:account:thing/thing"
      ]
    }
  ]
}
```

DeleteThingShadow

Supprime le shadow de l'objet spécifié.

Request

La demande comprend les en-têtes HTTP standard, plus l'URI suivante :

```
HTTP DELETE https://endpoint:8443/things/thingName/shadow?name=shadowName
Request body: (none)
```

Le paramètre de requête `name` n'est pas requis pour les shadows non nommés (classiques).

Response

En cas de réussite, la réponse comprend les en-têtes HTTP standard, plus le code et le corps suivants :

```
HTTP 200
Response body: Empty response state document
```

Notez que la suppression d'une ombre ne réinitialise pas son numéro de version à 0.

Authorization

La suppression d'un shadow d'appareil nécessite une stratégie qui permet au mandataire de réaliser l'action `iot:DeleteThingShadow`. Le service Device Shadow accepte deux formes d'authentification : Signature Version 4 avec les informations d'identification IAM ou l'authentification mutuelle TLS avec un certificat client.

Voici un exemple de stratégie qui permet à un mandataire de supprimer un shadow d'appareil :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:DeleteThingShadow",
      "Resource": [
        "arn:aws:iot:region:account:thing/thing"
      ]
    }
  ]
}
```

ListNamedShadowsForThing

Répertorie les shadows de l'objet spécifié.

Request

La demande comprend les en-têtes HTTP standard, plus l'URI suivante :

```
HTTP GET /api/things/shadow/ListNamedShadowsForThing/thingName?
nextToken=nextToken&pageSize=pageSize
Request body: (none)
```

nextToken

Jeton permettant de récupérer l'ensemble suivant de résultats.

Cette valeur est renvoyée sur les résultats paginés et est utilisée dans l'appel qui renvoie la page suivante.

pageSize

Nombre de noms de shadows à renvoyer dans chaque appel. Voir aussi `nextToken`.

thingName

Nom de l'objet pour lequel répertorier les shadows nommés.

Response

En cas de réussite, la réponse inclut les en-têtes HTTP standard, plus le code de réponse suivant et un Document de réponse de liste de noms de shadows (p. 589)

Note

Le shadow non nommé (classique) n'apparaît pas dans cette liste.

```
HTTP 200
Response body: Shadow name list document
```

Authorization

La liste d'un shadow d'appareil nécessite une stratégie qui permet au mandataire de réaliser`iot:ListNamedShadowsForThing`action. Le service Device Shadow accepte deux formes

d'authentification : Signature Version 4 avec les informations d'identification IAM ou l'authentification mutuelle TLS avec un certificat client.

Voici un exemple de stratégie qui permet à un mandataire de répertorier les shadows nommés d'un objet :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:ListNamedShadowsForThing",
      "Resource": [
        "arn:aws:iot:region:account:thing/thing"
      ]
    }
  ]
}
```

Rubriques MQTT de Device Shadow

Le service Device Shadow utilise des rubriques MQTT réservées pour permettre à des appareils et à des applications d'obtenir, de mettre à jour ou de supprimer les informations d'état d'un appareil (shadow).

La publication et l'abonnement à des rubriques shadow nécessite une autorisation basée sur les rubriques. AWS IoT se réserve le droit d'ajouter de nouvelles rubriques à la structure de rubriques existante. C'est pourquoi nous vous recommandons d'éviter les abonnements de caractère générique aux rubriques shadow. Par exemple, évitez de vous abonner à des filtres de rubrique tels que `$aws/things/thingName/shadow/#`, car le nombre de rubriques qui correspondent à ce filtre peut augmenter lorsqu'AWS IoT introduit de nouvelles rubriques shadow. Pour consulter des messages publiés dans ces rubriques, consultez Interaction avec les shadows (p. 566).

Les shadows peuvent être nommés ou non (classique). Les rubriques utilisées par chacun d'eux ne diffèrent que par le préfixe de rubrique. Ce tableau indique le préfixe de rubrique utilisé par chaque type de shadow.

Valeur *ShadowTopicPrefix*	Type de shadow
`$aws/things/`*thingName*`/shadow`	Shadow non nommé (classique)
`$aws/things/`*thingName*`/shadow/name/`*shadowName*	Shadow nommé

Pour créer une rubrique complète, sélectionnez le *ShadowTopicPrefix* pour le type de shadow auquel vous souhaitez faire référence, remplacez *thingName*, et *shadowName* le cas échéant, par leurs valeurs correspondantes, puis ajoutez cela au stub de rubrique comme indiqué dans les sections suivantes.

Voici les rubriques MQTT utilisés pour interagir avec les shadows.

Rubriques

- /get (p. 578)
- /get/accepted (p. 578)
- /get/rejected (p. 579)
- /update (p. 579)

/get

Publier un message vide dans cette rubrique pour obtenir le shadow d'appareil :

```
ShadowTopicPrefix/get
```

AWS IoT répond en publiant dans /get/accepted (p. 578) ou /get/rejected (p. 579).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get"
      ]
    }
  ]
}
```

/get/accepted

AWS IoT publie un document shadow de réponse dans cette rubrique lors du renvoi du shadow de l'appareil :

```
ShadowTopicPrefix/get/accepted
```

Pour plus d'informations, consultez Documents d'état de la réponse (p. 586).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

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

```
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/get/accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get/accepted"
      ]
    }
  ]
}
```

/get/rejected

AWS IoT publie un document de réponse d'erreur dans cette rubrique lorsqu'il ne peut pas retourner le shadow d'appareil :

```
ShadowTopicPrefix/get/rejected
```

Pour plus d'informations, consultez Document de réponse d'erreur (p. 589).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/get/rejected"
      ]
    },
    {
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/get/rejected"
      ]
    }
  ]
}
```

/update

Publier un document d'état de la demande dans cette rubrique pour mettre à jour l'objet d'appareil :

```
ShadowTopicPrefix/update
```

Le corps du message contient un document d'état de demande partiel (p. 586).

Un client tentant de mettre à jour l'état d'un appareil enverrait un document d'état de demande JSON avec la propriété `desired` comme la suivante :

```
{
  "state": {
    "desired": {
      "color": "red",
      "power": "on"
    }
  }
}
```

Un appareil mettant à jour son shadow enverrait un document d'état de demande JSON avec la propriété `reported`, comme ceci :

```
{
  "state": {
    "reported": {
      "color": "red",
      "power": "on"
    }
  }
}
```

AWS IoT répond en publiant dans /update/accepted (p. 581) ou /update/rejected (p. 583).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update"
      ]
    }
  ]
}
```

/update/delta

AWS IoT publie un document d'état de réponse dans cette rubrique quand il accepte une modification du shadow de l'appareil et que le document d'état de la demande contient des valeurs différentes pour les états `desired` et `reported` :

```
ShadowTopicPrefix/update/delta
```

Le tampon de messages contient un /documents d'état de la réponse delta (p. 587).

Détails du corps de message

- Un message publié dans `update/delta` comprend uniquement les attributs « souhaité » qui diffèrent entre les sections `desired` et `reported`. Il contient tous ces attributs, indépendamment qu'ils aient été contenus dans le message de mise à jour actuel ou qu'ils aient déjà été stockés dans AWS IoT. Les attributs qui ne diffèrent pas entre les sections `desired` et `reported` ne sont pas inclus.
- Si un attribut figure dans la section `reported`, mais qu'il n'a aucun équivalent dans la section `desired`, il n'est pas inclus.
- Si un attribut figure dans la section `desired`, mais qu'il n'a aucun équivalent dans la section `reported`, il n'est pas inclus.
- Si un attribut est supprimé de la section `reported`, mais qu'il existe toujours dans la section `desired`, il est inclus.

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/delta"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/delta"
      ]
    }
  ]
}
```

/update/accepted

AWS IoT publie un document d'état de réponse dans cette rubrique lorsqu'il accepte une modification du shadow d'appareil :

```
ShadowTopicPrefix/update/accepted
```

Le tampon de messages contient un /document d'état de la réponse accepté (p. 586).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/accepted"
      ]
    }
  ]
}
```

/update/documents

AWS IoT publie un document d'état dans cette rubrique chaque fois qu'une mise à jour du shadow est effectuée avec succès :

```
ShadowTopicPrefix/update/documents
```

Le corps du message contient un /documents d'état de la réponse documents (p. 587).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
documents"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/documents"
      ]
    }
```

```
  ]
}
```

/update/rejected

AWS IoT publie un document de réponse à l'erreur dans cette rubrique lorsqu'il rejette une modification du shadow d'appareil :

```
ShadowTopicPrefix/update/rejected
```

Le corps du message contient un Document de réponse d'erreur (p. 589).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/update/
rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update/rejected"
      ]
    }
  ]
}
```

/supprimer

Pour supprimer un shadow d'appareil, publiez un message vide dans la rubrique delete :

```
ShadowTopicPrefix/shadow/delete
```

Le contenu du message est ignoré.

Notez que la suppression d'une ombre ne réinitialise pas son numéro de version à 0.

AWS IoT répond en publiant dans /delete/accepted (p. 584) ou /delete/rejected (p. 584).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete"
      ]
    }
  ]
}
```

/delete/accepted

AWS IoT publie un message dans cette rubrique lors de la suppression d'un shadow d'appareil :

```
ShadowTopicPrefix/delete/accepted
```

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/delete/
accepted"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete/accepted"
      ]
    }
  ]
}
```

/delete/rejected

AWS IoT publie un document de réponse d'erreur dans cette rubrique lorsqu'il ne peut pas supprimer le shadow d'appareil :

```
ShadowTopicPrefix/delete/rejected
```

Le corps du message contient un Document de réponse d'erreur (p. 589).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/delete/
rejected"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete/rejected"
      ]
    }
  ]
}
```

Documents du service Device Shadow

Le service Device Shadow respecte toutes les règles de la spécification JSON. Valeurs, objets et tableaux sont stockés dans le document shadow de l'appareil.

Sommaire

Exemples de documents shadow

Le service Device Shadow utilise les documents suivants dans les opérations UPDATE, GET et DELETE à l'aide de l'API REST (p. 573) ou des messages de publication/abonnement MQTT (p. 577).

Exemples

- Document de réponse de liste de noms de shadows (p. 589)

Document d'état de demande

Un document d'état de demande a le format suivant :

```
{
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer1,
            "attribute2": "string1",
            ...
            "attributeN": boolean1
        }
    },
    "clientToken": "token",
    "version": version
}
```

- `state`— Les mises à jour concernent uniquement les champs spécifiés. Généralement, vous utiliserez la propriété `reported` ou la propriété `desired`, mais pas les deux dans la même demande.
  - `desired`— Les propriétés d'état et les valeurs dont la mise à jour est demandé dans l'appareil.
  - `reported`— Les propriétés d'état et les valeurs signalées par l'appareil.
- `clientToken`— Si utilisé, vous pouvez faire correspondre la requête et la réponse correspondante par le jeton client.
- `version` - Le service Device Shadow traite la mise à jour uniquement si la version spécifiée correspond à la dernière version qu'il a.

Documents d'état de la réponse

Les documents d'état de la réponse ont le format suivant, selon le type de réponse.

/document d'état de la réponse accepté

```
{
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
```

```
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "timestamp": timestamp,
    "clientToken": "token",
    "version": version
}
```

/documents d'état de la réponse delta

```
{
    "state": {
        "attribute1": integer2,
        "attribute2": "string2",
        ...
        "attributeN": boolean2
        }
    },
    "metadata": {
        "attribute1": {
            "timestamp": timestamp
        },
        "attribute2": {
            "timestamp": timestamp
        },
        ...
        "attributeN": {
            "timestamp": timestamp
        }
    },
    "timestamp": timestamp,
    "clientToken": "token",
    "version": version
}
```

/documents d'état de la réponse documents

```
{
  "previous" : {
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer1,
            "attribute2": "string1",
            ...
            "attributeN": boolean1
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
```

```
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        },
        "reported": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "version": version-1
},
"current": {
    "state": {
        "desired": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        },
        "reported": {
            "attribute1": integer2,
            "attribute2": "string2",
            ...
            "attributeN": boolean2
        }
    },
    "metadata": {
        "desired": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        },
        "reported": {
            "attribute1": {
                "timestamp": timestamp
            },
            "attribute2": {
                "timestamp": timestamp
            },
            ...
            "attributeN": {
                "timestamp": timestamp
            }
        }
    },
    "version": version
},
"timestamp": timestamp,
"clientToken": "token"
```

```
}
```

Propriétés du document d'état de réponse

- `previous`— Après une mise à jour réussie, il contient l'`state`de l'objet avant la mise à jour.
- `current`— Après une mise à jour réussie, il contient l'`state`de l'objet après la mise à jour.
- `state`
  - `reported`— Présente uniquement si une chose a rapporté des données dans le`reported`et contient uniquement les champs qui se trouvaient dans le document d'état de la demande.
  - `desired`— Présente uniquement si un périphérique a signalé des données dans le`desired`et contient uniquement les champs qui se trouvaient dans le document d'état de la demande.
  - `delta`— Présente uniquement si le`desired`DONNÉES diffèrent du courant de l'shadow`reported`DONNÉES.
- `metadata`— Contient les horodatages de chaque attribut dans la`desired`and`reported`afin que vous puissiez déterminer quand l'état a été mis à jour.
- `timestamp`— Date et heure Unix auxquelles la réponse a été générée parAWS IoT.
- `clientToken`— Présent uniquement si un jeton client a été utilisé lors de la publication d'un JSON valide dans l'interface`/update`Sujet.
- `version` - Version actuelle du document du shadow de l'appareil partagé dans AWS IoT. Elle est augmentée d'une unité par rapport à la version précédente du document.

Document de réponse d'erreur

Un document de réponse d'erreur a le format suivant :

```
{
    "code": error-code,
    "message": "error-message",
    "timestamp": timestamp,
    "clientToken": "token"
}
```

- `code`— Code de réponse HTTP qui indique le type d'erreur.
- `message`— Message texte qui fournit des informations supplémentaires.
- `timestamp`— Date et heure auxquelles la réponse a été générée parAWS IoT. Cette propriété n'est pas présente dans tous les documents de réponse d'erreur.
- `clientToken`— Présent uniquement si un jeton client a été utilisé dans le message publié.

Pour plus d'informations, consultez Messages d'erreur de Device Shadow (p. 594).

Document de réponse de liste de noms de shadows

Un document de réponse de liste de noms de shadows a le format suivant :

```
{
    "results": [
        "shadowName-1",
        "shadowName-2",
        "shadowName-3",
        "shadowName-n"
    ],
    "nextToken": "nextToken",
    "timestamp": timestamp
```

```
}
```

- `results`— Tableau des noms d'ombres.
- `nextToken`— Valeur de jeton à utiliser dans les requêtes paginées pour obtenir la page suivante dans la séquence. Cette propriété n'est pas présente quand il ne reste plus de noms de shadows à renvoyer.
- `timestamp`— Date et heure auxquelles la réponse a été générée parAWS IoT.

Propriétés du document

Un document de shadow d'appareil possède les propriétés suivantes :

`state`
: `desired`
  : État souhaité de l'appareil. Les applications peuvent écrire dans cette partie du document pour mettre à jour l'état d'un appareil directement sans avoir à s'y connecter.
: `reported`
  : État rapporté de l'appareil. Les appareils écrivent dans cette partie du document pour rapporter leur nouvel état. Les applications lisent cette partie du document pour déterminer le dernier état rapporté de l'appareil.

`metadata`
: Informations sur les données stockées dans la section `state` du document. Il s'agit notamment des horodatages, en heure Unix, de chaque attribut de la section `state`, qui vous permet de déterminer quand ils ont été mis à jour.

  Note

  Les métadonnées ne contribuent pas à la taille du document pour les limites de service ou la tarification. Pour plus d'informations, consultez AWS IoTLimites de service .

`timestamp`
: Indique quand le message a été envoyé par AWS IoT. En utilisant l'horodatage dans le message et les horodatages pour les attributs individuels dans la section `desired` ou `reported`, un appareil peut déterminer l'âge d'une propriété, même si l'appareil n'a pas d'horloge interne.

`clientToken`
: Chaîne unique du dispositif qui permet d'associer les réponses à des demandes dans un environnement MQTT.

`version`
: Version du document. Chaque fois que le document est mis à jour, ce numéro de version est incrémenté. Permet de s'assurer que la version du document en cours de mise à jour est la plus récente.

Pour plus d'informations, consultez Exemples de documents shadow (p. 585).

État Delta

L'état Delta est un type virtuel d'état qui contient l'écart entre les états `desired` et `reported`. Les champs de la section `desired` qui ne figurent pas dans la section `reported` sont inclus dans le delta. Les champs qui figurent dans la section `reported` mais pas dans la section `desired` ne sont pas inclus dans le delta. Le delta contient des métadonnées et ses valeurs sont égales aux métadonnées contenues dans le champ `desired`. Exemples :

```
{
  "state": {
    "desired": {
      "color": "RED",
      "state": "STOP"
    },
    "reported": {
      "color": "GREEN",
      "engine": "ON"
    },
    "delta": {
      "color": "RED",
      "state": "STOP"
    }
  },
  "metadata": {
    "desired": {
      "color": {
        "timestamp": 12345
      },
      "state": {
        "timestamp": 12345
      }
      },
      "reported": {
        "color": {
          "timestamp": 12345
        },
        "engine": {
          "timestamp": 12345
        }
      },
      "delta": {
        "color": {
          "timestamp": 12345
        },
        "state": {
          "timestamp": 12345
        }
      }
    },
    "version": 17,
    "timestamp": 123456789
  }
}
```

Lorsque des objets imbriqués diffèrent, le delta contient le chemin d'accès à la racine.

```
{
  "state": {
    "desired": {
      "lights": {
        "color": {
          "r": 255,
          "g": 255,
          "b": 255
        }
      }
    },
    "reported": {
      "lights": {
        "color": {
          "r": 255,
          "g": 0,
```

```
          "b": 255
        }
      }
    },
    "delta": {
      "lights": {
        "color": {
          "g": 255
        }
      }
    }
  },
  "version": 18,
  "timestamp": 123456789
}
```

Le service Device Shadow calcule le delta en itérant sur chaque champ de l'état `desired` et en le comparant à l'état `reported`.

Les tableaux sont traités comme des valeurs. Si un tableau de la section `desired` ne correspond pas au tableau de la section `reported`, l'intégralité du tableau souhaité est copiée dans le delta.

Documents shadow de gestion des versions

Le service Device Shadow prend en charge la gestion des versions sur chaque message de mise à jour, qu'il s'agisse de la demande ou de la réponse. Cela signifie qu'à chaque mise à jour d'un shadow, la version du document JSON est incrémentée. Cela permet de garantir deux choses :

- Un client peut recevoir une erreur s'il tente de remplacer un shadow par un numéro de version plus ancien. Le client est informé qu'il doit effectuer une resynchronisation avant de mettre à jour un shadow d'appareil.
- Un client peut décider de ne pas agir sur un message reçu si le message est d'une version inférieure à la version stockée par le client.

Un client peut contourner la mise en correspondance des versions en n'incluant pas de version dans le document shadow.

Jetons clients dans les documents shadow

Vous pouvez utiliser un jeton client avec la messagerie MQTT pour vérifier si une demande et une réponse contiennent le même jeton client. Cela garantit que la réponse et la demande sont associées.

> **Note**
>
> La longueur du jeton client ne peut pas dépasser 64 octets. Un jeton client d'une longueur supérieure à 64 octets génère une réponse 400 (Demande erronée) et un message d'erreur ClientToken non valide.

Propriétés de document shadow vides

Les propriétés `reported` et `desired` d'un document shadow peuvent être vides ou omises lorsqu'elles ne s'appliquent pas à l'état actuel du shadow. Par exemple, un document shadow contient une propriété `desired` uniquement s'il a un état souhaité. Voici un exemple valide d'un document d'état sans propriété `desired` :

```
{
```

```
    "reported" : { "temp": 55 }
}
```

La propriété `reported` peut également être vide, par exemple si le shadow n'a pas été mis à jour par l'appareil :

```
{
    "desired" : { "color" : "RED" }
}
```

Si une mise à jour entraîne l'affectation de la valeur null aux propriétés `desired` ou `reported`, elle est supprimée du document. Voici comment supprimer la propriété `desired` en la définissant sur `null`. Vous pouvez le faire lorsqu'un appareil met à jour son état, par exemple.

```
{
    "state": {
        "reported": {
            "color": "red"
        },
        "desired": null
    }
}
```

Un document shadow peut également n'avoir aucune des propriétés `desired` et `reported`, auquel cas le document shadow est vide. Ceci est un exemple de document shadow vide, mais valide.

```
{
}
```

Valeurs de tableau dans les documents shadow

Les shadows prennent en charge les tableaux, mais les traitent comme des valeurs normales, dans la mesure où une mise à jour d'un tableau remplace l'intégralité du tableau. Il n'est pas possible de mettre à jour une partie d'un tableau.

État initial :

```
{
    "desired" : { "colors" : ["RED", "GREEN", "BLUE" ] }
}
```

Mise à jour:

```
{
    "desired" : { "colors" : ["RED"] }
}
```

État final :

```
{
    "desired" : { "colors" : ["RED"] }
}
```

Les tableaux ne peuvent pas avoir de valeurs null. Par exemple, le tableau suivant n'est pas valide et sera rejeté.

```
{
    "desired" : {
        "colors" : [ null, "RED", "GREEN" ]
    }
}
```

Messages d'erreur de Device Shadow

Le service Device Shadow publie un message dans la rubrique d'erreur (via MQTT) en cas d'échec d'une tentative de modifier le document d'état. Ce message est émis uniquement en réponse à une demande de publication dans l'une des`$aws`Rubriques. Si le client met à jour le document à l'aide de l'API REST, il reçoit le code d'erreur HTTP dans le cadre de la réponse, et aucun message d'erreur MQTT n'est émis.

Code d'erreur HTTP	Messages d'erreur
400 (Requête erronée)	• JSON non valide • Nœud requis manquant : état • Le nœud d'état doit être un objet • Le nœud souhaitée doit être un objet • Le nœuds déclaré doit être un objet • Version non valide • ClientToken non valide **Note** Un jeton client d'une longueur supérieure à 64 octets génère cette réponse. • Le code JSON contient trop de niveaux d'imbrication ; le maximum est 6 • L'état contient un nœud non valide
401 (Accès non autorisé)	• Non autorisé
403 (Accès interdit)	• Accès interdit
404 (Introuvable)	• Objet non trouvé • Aucun shadow n'existe avec le nom : *shadowName*
409 (Conflit)	• Conflit de versions
413 (Charge utile trop importante)	• La charge utile dépasse la taille maximale autorisée
415 (Type de support non pris en charge)	• Codage documenté non pris en charge ; l'encodage pris en charge est UTF-8
429 (Nombre de requêtes trop élevé)	• Le service Device Shadow génère ce message d'erreur lorsqu'il y a plus de 10 demandes en vol.
500 (Erreur interne du serveur)	• Échec du service interne

Jobs

Des tâches AWS IoT peuvent être utilisées pour définir un ensemble d'opérations distantes qui sont envoyées vers un ou plusieurs appareils connectés à AWS IoT.

> Tip
>
> Pour obtenir des exemples de document de tâche, consultez l'exemple jobs-agent.js dans le kit SDK AWS IoT pour JavaScript.

Concepts clés relatifs aux tâches

tâche

> Une tâche est une opération distante qui est envoyée vers un ou plusieurs appareils connectés à AWS IoT et exécutée par ceux-ci. Par exemple, vous pouvez définir une tâche qui ordonne à un ensemble d'appareils de télécharger et d'installer les mises à jour d'une application ou d'un microprogramme, de redémarrer, de procéder à une rotation des certificats ou d'exécuter des opérations de dépannage à distance.

document de tâche

> Pour créer une tâche, vous devez d'abord créer un document de tâche qui est une description des opérations distantes devant être effectuées par les appareils.
>
> Les documents de tâche sont des documents JSON codés en UTF-8 et doivent contenir les informations dont vos appareils ont besoin pour effectuer une tâche. Un document de tâche contient une ou plusieurs URL où l'appareil peut télécharger une mise à jour ou d'autres données. Le document de tâche peut être stocké dans un compartiment Amazon S3 ou être inclus en ligne avec la commande de création de la tâche.

target

> Lorsque vous créez une tâche, vous spécifiez une liste de cibles qui correspondent aux appareils qui doivent effectuer les opérations. Les cibles peuvent être des objets ou des groupes d'objets (p. 210), ou les deux. Le service AWS IoT Jobs envoie un message pour informer chaque cible qu'une tâche est disponible.

exécution de tâche

> Une exécution de tâche est une instance d'une tâche sur un appareil cible. La cible commence une exécution d'une tâche en téléchargeant le document de tâche. Il exécute ensuite les opérations spécifiées dans le document et signale leur progression à AWS IoT. Un numéro d'exécution est un identifiant unique d'une exécution de tâche sur une cible spécifique. Le service Jobs fournit des commandes pour suivre la progression de l'exécution d'une tâche sur une cible et la progression d'une tâche sur l'ensemble des cibles.

tâche d'instantané

> Par défaut, une tâche est envoyée à toutes les cibles que vous spécifiez lorsque vous créez la tâche. Une fois que les cibles ont terminé la tâche (ou indiqué qu'elles étaient dans l'impossibilité de l'exécuter), la tâche est achevée.

tâche continue

> Une tâche continue est envoyée à toutes les cibles que vous spécifiez lorsque vous créez la tâche. Elle continue de s'exécuter et est envoyée à tous les nouveaux appareils (objets) qui sont ajoutées au groupe cible. Par exemple, une tâche continue peut être utilisée pour intégrer ou mettre à niveau

les appareils au fur et à mesure de leur ajout à un groupe. Vous pouvez rendre une tâche continue en définissant un paramètre facultatif lors de la création de la tâche.

déploiements

Vous pouvez spécifier la vitesse à laquelle les cibles sont averties d'une exécution de tâche en attente. Vous pouvez ainsi créer un déploiement étalé afin de mieux gérer les mises à jour, les redémarrages et autres opérations.

Le champ suivant peut être ajouté à la demande `CreateJob` pour spécifier le nombre maximum de tâches cibles à informer par minute. Cet exemple définit une fréquence de déploiement statique.

```
   "jobExecutionRolloutConfig": {
       "maximumPerMinute": "integer"
   }
```

Vous pouvez également définir une fréquence de déploiement variable avec le champ `exponentialRate`. L'exemple suivant crée un déploiement qui possède une fréquence exponentielle.

```
  "jobExecutionsRolloutConfig": {
      "exponentialRate": {
         "baseRatePerMinute": integer,
         "incrementFactor": integer,
         "rateIncreaseCriteria": {
            "numberOfNotifiedThings": integer, // Set one or the other
            "numberOfSucceededThings": integer // of these two values.
         },
         "maximumPerMinute": integer
   }
}
```

Pour plus d'informations sur la configuration des déploiements de tâche, consultez Configuration du déploiement et de l'interruption des tâches.

annuler

Vous pouvez créer un ensemble de conditions pour interrompre les déploiements lorsque les critères que vous spécifiez sont satisfaits. Pour en savoir plus, consultez Configuration du déploiement et de l'interruption des tâches.

URL présignées

Pour autoriser un accès sécurisé et limité dans le temps de l'appareil aux données au-delà de celles incluses dans le document de tâche lui-même, vous pouvez utiliser les URL Amazon S3 présignées. Vous pouvez placer vos données dans un compartiment Amazon S3 et ajouter un lien d'espace réservé aux données du document de tâche. Lorsque le service Jobs reçoit une demande pour le document de tâche, il analyse ce dernier à la recherche des liens d'espace réservé et les remplace par les URL Amazon S3 présignées.

Le lien d'espace réservé se présente sous la forme suivante :

```
${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/key}
```

où *`bucket`* correspond au nom de votre compartiment et *`key`* à l'objet du compartiment vers lequel vous établissez le lien.

Dans les régions de Pékin et de Ningxia, les URL pré-signées fonctionnent uniquement si le propriétaire de la ressource possède une licence ICP. Pour de plus amples informations, veuillez consulterAmazon Simple Storage Servicedans leDémarrer avecAWSServices en Chine.

délais d'expiration

Les délais d'expiration de tâche vous permettent de recevoir une notification chaque fois qu'une exécution de tâche se retrouve bloquée dans l'état `IN_PROGRESS` pendant une période étonnamment longue. Il existe deux types de minuteurs : minuteurs d'avancement et minuteurs d'étape.

Lorsque vous créez une tâche, vous pouvez définir une valeur pour la propriété `inProgressTimeoutInMinutes` de l'objet facultatif TimeoutConfig. Le minuteur en cours ne peut pas être mis à jour et s'applique à toutes les exécutions de tâche pour la tâche. Chaque fois qu'une exécution de tâche demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle défini, l'exécution de la tâche échoue et passe à l'état final `TIMED_OUT`. AWS IoT publie aussi une notification MQTT.

Vous pouvez aussi définir un minuteur d'étape pour une exécution de tâche en définissant une valeur pour `stepTimeoutInMinutes` lorsque vous appelez UpdateJobExecution. Le minuteur d'étape s'applique uniquement à l'exécution des tâches que vous mettez à jour. Vous pouvez définir une nouvelle valeur pour ce minuteur chaque fois que vous mettez à jour une exécution de tâche. Vous pouvez aussi créer un minuteur d'étape lorsque vous appelez StartNextPendingJobExecution. Si l'exécution de tâche demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle du minuteur d'étape, elle échoue et passe à l'état final `TIMED_OUT`. Le minuteur d'étape n'a aucun effet sur le minuteur d'avancement que vous définissez lorsque vous créez une tâche.

Le schéma et la description suivants illustrent la façon dont les délais du minuteur d'avancement et du minuteur d'avancement interagissent les uns avec les autres.

Création de tâche : `CreateJob` définit un minuteur d'avancement qui expire dans vingt minutes. Le minuteur s'applique à toutes les exécutions de tâche et ne peut pas être mise à jour.

12 H 00 : L'exécution des tâches démarre et passe à`IN_PROGRESS`État. Le minuteur d'avancement commence à s'exécuter.

12:05 PM: `UpdateJobExecution` crée un minuteur d'avancement avec la valeur 7 minutes. S'il n'est pas créé de nouveau minuteur d'étape, l'exécution de la tâche échoue à 12:12 PM.

12:10 PM: `UpdateJobExecution` crée un minuteur d'étape avec la valeur 5 minutes. Le minuteur d'étape précédent est ignoré. S'il n'est pas créé de nouveau minuteur d'étape, l'exécution de la tâche échoue à 12:15 PM.

12:13 PM: `UpdateJobExecution` crée un minuteur d'étape avec la valeur 9 minutes. L'exécution de la tâche arrive à expiration à 12:20, car le minuteur d'avancement expire à 12:20. Le minuteur d'étape ne peut pas dépasser la limite absolue créée par le minuteur d'avancement.

`UpdateJobExecution` peut également ignorer un minuteur d'étape qui a déjà été créé en créant un minuteur d'étape avec la valeur -1.

Gestion des tâches

Vous pouvez utiliser la stratégieAWS IoTconsole, l'API HTTPS Jobs, leAWS Command Line Interface, ou leAWSDes kits SDK pour créer et gérer des tâches Pour de plus amples informations, veuillez consulterAPI de gestion et de contrôle des tâches (p. 625),AWS CLIRéférence des commandes : iotouAWSKits SDK et outils.

L'objectif principal des tâches consiste à informer les périphériques d'une mise à jour du logiciel ou du microprogramme. Lors de l'envoi du code aux appareils, la bonne pratique consiste à signer le fichier de code. Cela permet aux appareils de détecter si le code a été modifié en transit.AWS IoTprend actuellement en charge la signature de code dans leAWS CLI. Pour obtenir des instructions sur l'utilisation de la signature de code avec les tâches, consultez??? (p. 599).

Pour en savoir plus, consultez Qu'est-ce que Code Signing pour AWS IoT ?

Avant de créer une tâche, vous devez créer un document de tâche. Si vous utilisez la signature de code pourAWS IoT, vous devez charger votre document de tâche dans un compartiment Amazon S3 versionné. Pour de plus amples informations sur la création d'un compartiment Amazon S3 et le chargement de fichiers, veuillez consulter.Mise en route avec Amazon Simple Storage Servicedans leManuel de mise en route Amazon S3.

Votre document de tâche peut contenir une URL Amazon S3 présignée qui pointe sur votre fichier de code (ou tout autre fichier). Les URL Amazon S3 présignées sont valides pour une quantité de temps limitée et ne sont pas générées jusqu'à ce qu'un appareil demande un document de tâche. Comme l'URL pré-signée n'a pas été créée lorsque vous avez créé le document de tâche, vous placez une URL d'espace réservé dans votre document de tâche à la place. L'URL d'espace réservé se présente comme suit :`${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/<bucket>/<code file>}`où*bucket*Compartiment Amazon S3 qui contient le fichier de code et*fichier de code*La clé Amazon S3 du fichier de code.

Lorsqu'un périphérique demande le document de tâche, AWS IoT génère l'URL pré-signée et remplace l'URL d'espace réservé par l'URL pré-signée. Votre document de tâche est alors envoyé à l'appareil.

Lorsque vous créez une tâche qui utilise les URL Amazon S3 présignées, vous devez fournir un rôle IAM qui accorde l'autorisation de télécharger les fichiers depuis le compartiment Amazon S3 où les données ou les mises à jour sont stockées. Le rôle doit également accorder à AWS IoT l'autorisation d'endosser le rôle.

Vous pouvez éventuellement spécifier un délai d'expiration pour l'URL présignée. Pour plus d'informations, consultez CreateJob (p. 633).

Pour autoriser le service Jobs d'endosser votre rôle

1. Connectez-vous à la AWS Management Console et ouvrez la console IAM à l'adresse https://console.aws.amazon.com/iam/.
2. Dans le panneau de navigation, choisissez Roles (Rôles).
3. Recherchez votre rôle et sélectionnez-le.
4. Dans l'onglet Trust Relationships (Relations d'approbation), sélectionnez Edit Trust Relationship (Modifier les relations d'approbation).
5. Sur la page Modifier la relation d'approbation, remplacez le document de stratégie par le code JSON suivant :

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

6. Choisissez Update Trust Policy (Mettre à jour la stratégie d'approbation).
7. Si votre tâche utilise un document de travail qui est un objet Amazon S3, choisissezAutorisationset avec le code JSON suivant, ajoutez une stratégie qui accorde l'autorisation de télécharger les fichiers depuis votre compartiment Amazon S3 :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::your_S3_bucket/*"
        }
    ]
}
```

Création et gestion de tâches (console)

Pour créer une tâche

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissez Gérer, puis Tâches.
3. Choisissez Créer une tâche.
4. Choisissez Créer une tâche personnalisée.
5. Entrez un ID alphanumérique pour votre tâche, puis une description facultative.

   Note

   Nous vous déconseillons d'utiliser des informations personnelles identifiables dans vos ID ou descriptions de tâche.
6. Sélectionnez l'appareil ou les groupes d'appareils que vous souhaitez mettre à jour.
7. Sous Ajouter un fichier de tâche, choisissez Sélectionner, puis sélectionnez votre document de tâche.
8. Sous Type de tâche, choisissez l'option appropriée pour votre mise à jour, puis choisissez Suivant.
9. Spécifiez les valeurs des configurations avancées, puis choisissez Créer.

Une fois que vous avez créé la tâche, la console génère une signature JSON et la place dans votre document de tâche.

Vous pouvez utiliser la console AWS IoT pour afficher le statut d'une tâche, annuler une tâche ou la supprimer.

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissez Gérer, puis Tâches.

Création et gestion de tâches (interface de ligne de commande)

Cette section décrit comment créer et gérer des tâches.

Création de tâches

Vous utilisez la commande CreateJob pour créer une tâche AWS IoT. La tâche est mise en file d'attente en vue de son exécution sur les cibles (objets ou groupes d'objets) que vous spécifiez. Pour créer unAWS IoT, vous avez besoin d'un document de tâche qui peut être inclus dans le corps de la demande ou en tant que lien vers un document Amazon S3. Si la tâche inclut le téléchargement de fichiers à l'aide d'URL

Amazon S3 présignées, vous avez besoin d'un ARN de rôle IAM qui est autorisé à télécharger le fichier et qui accorde l'autorisation auAWS IoTService des tâches pour assumer le rôle.

Signature de code avec les tâches

Si vous utilisez la signature de code pourAWS IoT, vous devez lancer une tâche de signature de code et inclure la sortie dans votre document de tâche. Utilisation de l'start-signing-jobPour créer une tâche de signature de code.`start-signing-job`retourne un ID de tâche. Utilisation de l'describe-signing-jobCommande pour obtenir l'emplacement Amazon S3 dans lequel la signature est stockée. Vous pouvez alors télécharger la signature depuis Amazon S3. Pour plus d'informations sur les tâches de signature de code, consultezSignature de code pourAWS IoT.

Votre document de tâche doit contenir un espace réservé d'URL pré-signée pour votre fichier de code et la sortie de la signature JSON être placée dans un compartiment Amazon S3 à l'aide de l'start-signing-job, entourée d'une commande`codesign`Élément :

```
{
    "presign": "${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/bucket/image}",
    "codesign": {
        "rawPayloadSize": <image-file-size>,
        "signature": <signature>,
        "signatureAlgorithm": <signature-algorithm>,
        "payloadLocation": {
            "s3": {
                "bucketName": <my-s3-bucket>,
                "key": <my-code-file>,
                "version": <code-file-version-id>
            }
        }
    }
}
```

Création d'une tâche avec un document de tâche

La commande suivante montre comment créer une tâche à l'aide d'un document de tâche (*job-document.json*) stocké dans un compartiment Amazon S3 (*jobBucket*) et un rôle avec l'autorisation de télécharger des fichiers à partir d'Amazon S3 (*S3DownloadRole*).

```
aws iot create-job  \
      --job-id 010  \
      --targets arn:aws:iot:us-east-1:123456789012:thing/thingOne  \
      --document-source https://s3.amazonaws.com/my-s3-bucket/job-document.json  \
      --timeout-config inProgressTimeoutInMinutes=100 \
      --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\": 50,
 \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}}, \"maximumPerMinute\": 1000}" \
      --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
      --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/
S3DownloadRole\", \"expiresInSec\":3600}"
```

La tâche est exécutée sur *thingOne*.

Le paramètre facultatif `timeout-config` spécifie la durée allouée à chaque appareil pour terminer l'exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.

Le minuteur en cours ne peut pas être mis à jour et s'applique à toutes les exécutions de tâche pour la tâche. Chaque fois qu'une exécution de tâche demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle défini, l'exécution de la tâche échoue et passe à l'état final `TIMED_OUT`. AWS IoT publie aussi une notification MQTT.

Pour plus d'informations sur la création de configurations relatives aux déploiements et interruptions de tâche, consultez la section Configuration du déploiement et de l'interruption des tâches.

> Note
>
> Les documents de Job qui sont spécifiés en tant que fichiers Amazon S3 sont extraits au moment de la création de la tâche. La modification du contenu du fichier Amazon S3 que vous avez utilisé comme source de votre document de tâche une fois que vous avez créé la tâche ne modifie pas ce qui est envoyé aux cibles de la tâche.

Mettre à jour une tâche

Vous utilisez la commande UpdateJob pour mettre à jour une tâche. Vous pouvez mettre à jour les champs `description`, `presignedUrlConfig`, `jobExecutionsRolloutConfig`, `abortConfig` et `timeoutConfig` d'une tâche.

```
aws iot update-job  \
  --job-id 010  \
  --description "updated description" \
  --timeout-config inProgressTimeoutInMinutes=100 \
  --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\": 50,
 \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}, \"maximumPerMinute\": 1000}}" \
  --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
  --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/S3DownloadRole\",
 \"expiresInSec\":3600}"
```

Pour en savoir plus, consultez Configuration du déploiement et de l'interruption des tâches.

Annulation d'une tâche

Vous utilisez la commande CancelJob pour annuler une tâche. L'annulation d'une tâche conduit AWS IoT à interrompre le lancement de toute nouvelle exécution pour la tâche. Il annule également toutes les exécutions de travail qui se trouvent dans un`QUEUED`État.AWS IoTlaisse toute exécution de tâche dans un état terminal, car l'appareil a déjà mené à bien la tâche. Si une exécution de tâche a le statut `IN_PROGRESS`, elle reste aussi en l'état, à moins que vous utilisiez le paramètre facultatif `--force`.

La commande suivante montre comment annuler une tâche avec l'ID 010.

```
aws iot cancel-job --job-id 010
```

La commande affiche la sortie suivante :

```
{
    "jobArn": "string",
    "jobId": "string",
    "description": "string"
}
```

Lorsque vous annulez une tâche, les exécutions de tâche dont l'état est `QUEUED` sont annulées. Les exécutions de tâche dont l'état est `IN_PROGRESS` sont annulées si vous spécifiez le paramètre facultatif `--force`. Les exécutions de tâche qui se trouvent dans un état terminal ne sont pas annulées.

> Warning
>
> L'annulation d'une tâche dont l'état est `IN_PROGRESS` (en définissant le paramètre `--force`) a pour effet d'annuler les exécutions de tâche en cours et empêche l'appareil qui exécute la tâche de mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

Le statut d'une tâche annulée ou de l'une de ses exécutions de tâche est cohérent à terme : AWS IoT cesse de planifier de nouvelles exécutions de tâche et les exécutions de tâche `QUEUED` pour cette tâche aux appareils dès que possible. La modification de l'état de l'exécution d'une tâche en `CANCELED` peut prendre du temps, selon le nombre d'appareils et autres facteurs.

Si une tâche est annulée, car elle a satisfait les critères définis par un objet `AbortConfig`, le service ajoute les valeurs renseignées automatiquement pour les champs `reasonCode` et `comment`. Vous pouvez créer vos propres valeurs pour `reasonCode` lorsque la tâche d'annulation est orientée utilisateurs.

Annulation d'une exécution de tâche

La commande CancelJobExecution permet d'annuler une exécution de tâche sur un appareil déterminé. Il annule l'exécution de la tâche qui se trouve dans un état `QUEUED`. Si vous souhaitez annuler une exécution de tâche dont le statut est en cours, vous devez utiliser le paramètre `--force`.

La commande suivante montre comment annuler l'exécution d'une tâche depuis la tâche 010 s'exécutant sur `myThing`.

```
aws iot cancel-job-execution --job-id 010 --thing-name myThing
```

La commande n'affiche aucune sortie.

Une exécution de tâche qui se trouve dans un état `QUEUED` est annulée. Une exécution de tâche dont l'état est `IN_PROGRESS` est annulée si vous spécifiez le paramètre facultatif `--force`. Les exécutions de tâche qui se trouvent dans un état terminal ne peuvent pas être annulées.

> Warning
>
> Lorsque vous annulez une exécution de tâche dont le statut est `IN_PROGRESS`, l'appareil ne peut pas mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Si l'exécution de tâche se trouve dans un état terminal ou si l'exécution de tâche présente le statut `IN_PROGRESS` et que le paramètre `--force` a la valeur `true`, cette commande entraîne une exception `InvalidStateTransitionException`.

Le statut d'une exécution de tâche annulée est cohérent à terme. La modification de l'état de l'exécution d'une tâche en `CANCELED` peut prendre du temps, selon différents facteurs.

Suppression d'une tâche

La commande DeleteJob permet de supprimer une tâche et les exécutions de tâche associées. Par défaut, vous pouvez uniquement supprimer une tâche qui se trouve dans un état terminal (`SUCCEEDED` ou `CANCELED`). Dans le cas contraire, une exception se produit. Vous pouvez supprimer une tâche qui se trouve dans l'état `IN_PROGRESS` si le paramètre `force` a la valeur `true`.

Pour supprimer une tâche, exécutez la commande suivante :

```
aws iot delete-job --job-id 010 --force|--no-force
```

La commande n'affiche aucune sortie.

> Warning
>
> Lorsque vous supprimez une tâche qui est dans l'état `IN_PROGRESS`, le périphérique qui exécute la tâche ne peut pas accéder aux informations sur la tâche ou mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

La suppression d'une tâche peut prendre un certain temps qui varie en fonction du nombre d'exécutions de tâche créées pour la tâche et autres facteurs. Pendant la suppression de la tâche, l'état de celle-ci indique `DELETION_IN_PROGRESS`. Toute tentative de suppression ou d'annulation d'une tâche dont le statut est `DELETION_IN_PROGRESS` entraîne une erreur.

Seules 10 tâches peuvent avoir l'état `DELETION_IN_PROGRESS` en même temps. Sinon, une exception `LimitExceededException` se produit.

Obtention d'un document de tâche

Vous utilisez la commande GetJobDocument pour récupérer un document de tâche pour une tâche. Un document de tâche est une description des opérations distantes à exécuter par les appareils.

Exécutez la commande suivante pour obtenir un document de tâche.

```
aws iot get-job-document --job-id 010
```

La commande renvoie le document de tâche de la tâche spécifiée :

```
{
    "document": "{\n\t\"operation\":\"install\",\n\t\"url\":\"http://amazon.com/
firmWareUpate-01\",\n\t\"data\":\"${aws:iot:s3-presigned-url:https://s3.amazonaws.com/job-
test-bucket/datafile}\"\n}"
}
```

> Note
>
> Lorsque vous récupérez un document de tâche à l'aide de cette commande, les URL d'espace réservé ne sont pas remplacées par les URL Amazon S3 présignées. Lorsqu'un périphérique appelle leGetPendingJobExecutions (p. 684)MQTT, les URL d'espace réservé sont remplacées par les URL Amazon S3 présignées du document de tâche.

Affichage des tâches

Vous utilisez leListJobsCommande pour obtenir la liste de toutes les tâches dans votre Compte AWS . Les données de travail et d'exécution de travail sont conservées pendant une durée limitée. Exécutez la commande suivante pour répertorier toutes les tâches de votre Compte AWS :

```
aws iot list-jobs
```

La commande répertorie toutes les tâches de votre compte, triées sur le statut de la tâche :

```
{
    "jobs": [
        {
            "status": "IN_PROGRESS",
            "lastUpdatedAt": 1486687079.743,
```

```
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/013",
            "createdAt": 1486687079.743,
            "targetSelection": "SNAPSHOT",
            "jobId": "013"
        },
        {
            "status": "SUCCEEDED",
            "lastUpdatedAt": 1486685868.444,
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/012",
            "createdAt": 1486685868.444,
            "completedAt": 148668789.690,
            "targetSelection": "SNAPSHOT",
            "jobId": "012"
        },
        {
            "status": "CANCELED",
            "lastUpdatedAt": 1486678850.575,
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/011",
            "createdAt": 1486678850.575,
            "targetSelection": "SNAPSHOT",
            "jobId": "011"
        }
    ]
}
```

Description d'une tâche

La commande DescribeJob permet d'obtenir le statut d'une tâche. La commande suivante explique comment décrire une tâche :

```
$ aws iot describe-job --job-id 010
```

La commande renvoie le statut de la tâche spécifiée. Exemples :

```
{
    "documentSource": "https://s3.amazonaws.com/job-test-bucket/job-document.json",
    "job": {
        "status": "IN_PROGRESS",
        "jobArn": "arn:aws:iot:us-east-1:123456789012:job/010",
        "targets": [
            "arn:aws:iot:us-east-1:123456789012:thing/myThing"
        ],
        "jobProcessDetails": {
            "numberOfCanceledThings": 0,
            "numberOfFailedThings": 0,
            "numberOfInProgressThings": 0,
            "numberOfQueuedThings": 0,
            "numberOfRejectedThings": 0,
            "numberOfRemovedThings": 0,
            "numberOfSucceededThings": 0,
            "numberOfTimedOutThings": 0,
            "processingTargets": [
                arn:aws:iot:us-east-1:123456789012:thing/thingOne,
                arn:aws:iot:us-east-1:123456789012:thinggroup/thinggroupOne,
                arn:aws:iot:us-east-1:123456789012:thing/thingTwo,
                arn:aws:iot:us-east-1:123456789012:thinggroup/thinggroupTwo
            ]
        },
        "presignedUrlConfig": {
            "expiresInSec": 60,
            "roleArn": "arn:aws:iam::123456789012:role/S3DownloadRole"
        },
        "jobId": "010",
```

```
        "lastUpdatedAt": 1486593195.006,
        "createdAt": 1486593195.006,
        "targetSelection": "SNAPSHOT",
        "jobExecutionsRolloutConfig": {
            "exponentialRate": {
                "baseRatePerMinute": integer,
                "incrementFactor": integer,
                "rateIncreaseCriteria": {
                    "numberOfNotifiedThings": integer, // Set one or the other
                    "numberOfSucceededThings": integer // of these two values.
                },
            "maximumPerMinute": integer
         }
        },
        "abortConfig": {
            "criteriaList": [
                {
                    "action": "string",
                    "failureType": "string",
                    "minNumberOfExecutedThings": integer,
                    "thresholdPercentage": integer
                }
            ]
        },
        "timeoutConfig": {
           "inProgressTimeoutInMinutes": number
          }
    }
}
```

Affichage des exécutions d'une tâche

Une tâche en cours d'exécution sur un appareil spécifique est représentée par un objet d'exécution de tâche. La commande ListJobExecutionsForJob permet d'afficher toutes les exécutions de tâche d'une tâche. L'exemple suivant montre comment afficher les exécutions d'une tâche :

```
aws iot list-job-executions-for-job --job-id 010
```

La commande renvoie une liste d'exécutions de tâche :

```
{
    "executionSummaries": [
    {
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingOne",
        "jobExecutionSummary": {
            "status": "QUEUED",
            "lastUpdatedAt": 1486593196.378,
            "queuedAt": 1486593196.378,
            "executionNumber": 1234567890
        }
    },
    {
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingTwo",
        "jobExecutionSummary": {
            "status": "IN_PROGRESS",
            "lastUpdatedAt": 1486593345.659,
            "queuedAt": 1486593196.378,
            "startedAt": 1486593345.659,
            "executionNumber": 4567890123
        }
    }
    ]
```

```
}
```

Affichage des exécutions de tâche pour un objet

La commande ListJobExecutionsForThing permet d'afficher toutes les exécutions de tâche s'exécutant sur un objet. L'exemple suivant montre comment afficher les exécutions de tâche d'un objet :

```
aws iot list-job-executions-for-thing --thing-name thingOne
```

La commande renvoie la liste des exécutions de tâche qui sont en cours d'exécution ou qui se sont exécutées sur l'objet spécifié :

```
{
    "executionSummaries": [
    {
        "jobExecutionSummary": {
            "status": "QUEUED",
            "lastUpdatedAt": 1486687082.071,
            "queuedAt": 1486687082.071,
            "executionNumber": 9876543210
        },
        "jobId": "013"
    },
    {
        "jobExecutionSummary": {
            "status": "IN_PROGRESS",
            "startAt": 1486685870.729,
            "lastUpdatedAt": 1486685870.729,
            "queuedAt": 1486685870.729,
            "executionNumber": 1357924680
        },
        "jobId": "012"
    },
    {
        "jobExecutionSummary": {
            "status": "SUCCEEDED",
            "startAt": 1486678853.415,
            "lastUpdatedAt": 1486678853.415,
            "queuedAt": 1486678853.415,
            "executionNumber": 4357680912
        },
        "jobId": "011"
    },
    {
        "jobExecutionSummary": {
            "status": "CANCELED",
            "startAt": 1486593196.378,
            "lastUpdatedAt": 1486593196.378,
            "queuedAt": 1486593196.378,
            "executionNumber": 2143174250
        },
        "jobId": "010"
    }
    ]
}
```

Description d'une exécution de tâche

La commande DescribeJobExecution permet d'obtenir le statut d'une exécution de tâche. Pour identifier l'exécution de tâche, vous devez spécifier un ID de tâche, un nom d'objet et éventuellement un numéro d'exécution. La commande suivante explique comment décrire une exécution de tâche :

```
aws iot describe-job-execution --job-id 017 --thing-name thingOne
```

La commande renvoie la chaîne JobExecution (p. 680). Exemples :

```
{
    "execution": {
        "jobId": "017",
        "executionNumber": 4516820379,
        "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/thingOne",
        "versionNumber": 123,
        "createdAt": 1489084805.285,
        "lastUpdatedAt": 1489086279.937,
        "startedAt": 1489086279.937,
        "status": "IN_PROGRESS",
        "approximateSecondsBeforeTimedOut": 100,
        "statusDetails": {
            "status": "IN_PROGRESS",
            "detailsMap": {
                "percentComplete": "10"
            }
        }
    }
}
```

Suppression d'une exécution de tâche

Pour supprimer une exécution de tâche, exécutez la commande DeleteJobExecution. Pour identifier l'exécution de tâche, vous devez spécifier un ID de tâche, un nom d'objet et un numéro d'exécution. La commande suivante explique comment supprimer une exécution de tâche :

```
aws iot delete-job-execution --job-id 017 --thing-name thingOne --execution-number
 1234567890 --force|--no-force
```

La commande n'affiche aucune sortie.

Par défaut, le statut de l'exécution d'une tâche doit être `QUEUED` ou dans un état terminal (`SUCCEEDED`, `FAILED`, `REJECTED`, `TIMED_OUT`, `REMOVED` ou `CANCELED`). Dans le cas contraire, une erreur se produit. Pour supprimer une exécution de tâche avec le statut `IN_PROGRESS`, vous pouvez définir le paramètre `force` sur `true`.

> Warning
>
> Lorsque vous supprimez une exécution de tâche qui est dans l'état `IN_PROGRESS`, le périphérique qui exécute la tâche ne peut pas accéder aux informations sur la tâche ou mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Modèles de tâche

> Note
>
> La fonction Modèles de tâche est en version préliminaire et susceptible d'être modifiée.

Les modèles de Job vous permettent de préconfigurer des tâches afin de pouvoir les déployer sur plusieurs ensembles d'équipements cibles. Ils offrent un moyen efficace de créer des configurations standard pour les actions à distance que vous devez déployer à plusieurs reprises sur vos appareils. Lorsque vous créez des modèles de tâche, vous pouvez spécifier les configurations et les ressources suivantes.

- Documents de Job
- Critères de déploiement et d'annulation
- Les critères de délai d'attente

Le cas échéant, vous pouvez créer des modèles à partir de travaux existants. Cela peut être utile lorsque vous devez effectuer des opérations, telles que des correctifs de sécurité et des corrections de bogues, de manière standardisée pour les clients.

Vous pouvez créer des tâches à partir de modèles de tâche à l'aide de l'outilAWS CLIet l'AWS IoTconsole Les opérateurs peuvent également créer des tâches à partir de modèles de tâche en utilisant Fleet Hub pourAWS IoTApplications Web de gestion des périphériques. Pour de plus amples informations sur l'utilisation des modèles de tâche dans les applications Fleet Hub, veuillez consulterUtilisation des modèles de tâche dans Fleet Hub pourAWS IoTGestion des appareils.

Rubriques

Création et gestion de modèles de tâche (console)

> Note
>
> La fonction Modèles de tâche est en version préliminaire et susceptible d'être modifiée.

Cette rubrique explique comment créer, supprimer et afficher les détails sur les modèles de tâche à l'aide de l'AWS IoTconsole

Créer un modèle de tâche à partir de zéro

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissezGérer, puisModèles de tâche.

   > Note
   >
   > Vous pouvez également accéder au siteModèles de tâcheà partir de la pageServices connexespage sousHub de flotte.
3. ChoisissezCréer un modèle de tâche.
4. Entrez un identifiant alphanumérique pour votre travail.
5. Entrez une description alphanumérique pour votre travail.

   > Note
   >
   > Nous vous déconseillons d'utiliser des informations personnelles identifiables dans vos ID ou descriptions de tâche.
6. UnderDocument de modèle de Job, entrez l'URL S3 ou choisissezParcourir S3, puis accédez à votre document de travail et sélectionnez-le.

   > Note
   >
   > Vous ne pouvez sélectionner que les compartiments S3 de votre région.
7. Spécifiez les valeurs des configurations avancées, puis choisissezCréer un modèle de tâche. Votre nouveau modèle de tâche s'affiche sur la pageModèles de tâche.

   Pour plus d'informations sur les configurations de déploiement et d'abandon, consultez??? (p. 709). Pour plus d'informations sur les configurations de ces délais, consultez.délais d'expiration (p. 597).

Créer un modèle de tâche à partir d'une tâche existante

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissez Gérer, puis Tâches.
3. Sélectionnez la tâche que vous souhaitez utiliser comme base du modèle de tâche et choisissezCopier dans le modèle de tâche.
4. Entrez un identifiant alphanumérique pour votre travail.
5. Entrez une description alphanumérique pour votre travail.
6. Le cas échéant, sélectionnez un autre document de travail ou modifiez les configurations avancées de la tâche d'origine, puis choisissezCréer un modèle de tâche. Votre nouveau modèle de tâche s'affiche sur la pageModèles de tâche.

Créer une tâche à partir d'un modèle de tâche

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissezGérer, puisModèles de tâche.
3. Recherchez le modèle de tâche que vous souhaitez utiliser, puis accédez à sa page de détails. ChoisissezCréer une tâche avec ce modèle.
4. Entrez un nom alphanumérique unique pour la tâche.
5. Entrez une description alphanumérique pour la tâche.
6. Si vous le souhaitez, ajoutez des balises à la tâche. Choisissez Next (Suivant).
7. Dans la pageConfiguration des fichiers, sousAppareils, sélectionnez les objets et les groupes de choses pour la tâche à cibler.
8. UnderFichier, vérifiez que l'URL Amazon S3 est correcte. Si vous souhaitez utiliser un autre document de tâche, choisissezParcouriret sélectionnez un compartiment et un document différents. Choisissez Next (Suivant).
9. Dans la pageConfiguration Job, sélectionnez le type d'exécution du travail. Un travail peut être continu ou instantané. Une tâche d'instantané est terminée lorsqu'elle termine son exécution sur les machines et les groupes cibles. Un travail continu s'applique aux groupes de choses et s'exécute sur n'importe quel périphérique que vous ajoutez ultérieurement à un groupe cible spécifié.
10. Vous pouvez également modifier les configurations avancées depuis le modèle de tâche, puis choisissezSuivant.
11. Dans la pageVérifier et créer, consultez les propriétés et les configurations de votre tâche. ChoisissezModifierPour tout ensemble de propriétés et configurations que vous souhaitez modifier.
12. Choisissez Submit. Votre nouvelle tâche s'affiche sur la page Tâches.

Vous pouvez également créer des tâches à partir de modèles de tâche avec les applications Web Fleet Hub. Pour plus d'informations sur la création de tâches dans Fleet Hub, consultezUtilisation des modèles de tâche dans Fleet Hub pourAWS IoTGestion des appareils.

Suppression d'un modèle de tâche

1. Accédez à la console AWS IoT.
2. Dans le volet de navigation, choisissezGérer, puisModèles de tâche.
3. Recherchez et sélectionnez le modèle de tâche à supprimer.
4. Sélectionnez Delete (Supprimer).
5. Le modèle de tâche n'apparaît plus sur la pageModèles de tâche.

Création et gestion de modèles de tâche (interface de ligne de commande)

> **Note**
>
> La fonction Modèles de tâche est en version préliminaire et susceptible d'être modifiée.

Cette rubrique explique comment créer, supprimer et récupérer des détails sur les modèles de tâche à l'aide de l'AWS CLI.

Créer un modèle de tâche à partir de zéro

Procédez comme suit :AWS CLILa commande montre comment créer une tâche à l'aide d'un document de tâche (*job-document.json*) stocké dans un compartiment Amazon S3 (*jobBucket*) et un rôle avec l'autorisation de télécharger des fichiers à partir d'Amazon S3 (*S3DownloadRole*).

```
aws iot create-job-template  \
      --job-template-id 010  \
      --document-source https://s3.amazonaws.com/my-s3-bucket/job-document.json  \
      --timeout-config inProgressTimeoutInMinutes=100 \
      --job-executions-rollout-config "{ \"exponentialRate\": { \"baseRatePerMinute\":
 50, \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000,
 \"numberOfSucceededThings\": 1000}}, \"maximumPerMinute\": 1000}" \
      --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\":
 \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action
\": \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200,
 \"thresholdPercentage\": 50}]}" \
      --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/S3DownloadRole
\", \"expiresInSec\":3600}"
```

Le paramètre facultatif `timeout-config` spécifie la durée allouée à chaque appareil pour terminer l'exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur`TIMED_OUT`.

Le minuteur en cours ne peut pas être mis à jour et s'applique à toutes les exécutions de tâche pour la tâche. Chaque fois qu'une exécution de tâche demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle défini, l'exécution de la tâche échoue et passe à l'état final `TIMED_OUT`. AWS IoT publie aussi une notification MQTT.

Pour plus d'informations sur la création de configurations relatives aux déploiements et interruptions de tâche, consultez la section Configuration du déploiement et de l'interruption des tâches.

> **Note**
>
> Les documents de Job qui sont spécifiés en tant que fichiers Amazon S3 sont extraits au moment de la création de la tâche. La modification du contenu du fichier Amazon S3 que vous avez utilisé comme source de votre document de tâche une fois que vous avez créé la tâche ne modifie pas ce qui est envoyé aux cibles de la tâche.

Créer un modèle de tâche à partir d'une tâche existante

Procédez comme suit :AWS CLIcrée un modèle de tâche en spécifiant l'Amazon Resource Name (ARN) d'une tâche existante. Le nouveau modèle de tâche utilise toutes les configurations spécifiées dans le travail. Le cas échéant, vous pouvez modifier n'importe quelle configuration du travail existant à l'aide de l'un des paramètres facultatifs.

```
aws iot create-job-template  \
      --job-arn arn:aws:iot:region:123456789012:job/job-name  \
      --timeout-config inProgressTimeoutInMinutes=100
```

Obtenir des informations sur un modèle de tâche

Procédez comme suit :AWS CLICommande obtient les détails d'un modèle de tâche spécifié.

```
aws iot describe-job-template \
      --job-template-id template-id
```

La commande affiche la sortie suivante.

```
{
   "abortConfig": {
      "criteriaList": [
         {
            "action": "string",
            "failureType": "string",
            "minNumberOfExecutedThings": number,
            "thresholdPercentage": number
         }
      ]
   },
   "createdAt": number,
   "description": "string",
   "document": "string",
   "documentSource": "string",
   "jobExecutionsRolloutConfig": {
      "exponentialRate": {
         "baseRatePerMinute": number,
         "incrementFactor": number,
         "rateIncreaseCriteria": {
            "numberOfNotifiedThings": number,
            "numberOfSucceededThings": number
         }
      },
      "maximumPerMinute": number
   },
   "jobTemplateArn": "string",
   "jobTemplateId": "string",
   "presignedUrlConfig": {
      "expiresInSec": number,
      "roleArn": "string"
   },
   "timeoutConfig": {
      "inProgressTimeoutInMinutes": number
   }
}
```

Modèles de tâche

Procédez comme suit :AWS CLILa liste de tous les modèles de tâche dans votre Compte AWS .

```
aws iot list-job-templates
```

La commande affiche la sortie suivante.

```
{
   "jobTemplates": [
      {
         "createdAt": number,
         "description": "string",
         "jobTemplateArn": "string",
         "jobTemplateId": "string"
      }
   ],
   "nextToken": "string"
}
```

Utilisez la valeur de la stratégie`nextToken`pour récupérer des pages supplémentaires de résultats.

Suppression d'un modèle de tâche

Procédez comme suit :AWS CLIcommande supprime un modèle de tâche spécifié.

```
aws iot delete-job-template \
      --job-template-id template-id
```

La commande n'affiche aucune sortie.

Créer une tâche à partir d'un modèle de tâche

Procédez comme suit :AWS CLIcommande crée une tâche à partir d'un modèle de tâche Il cible un périphérique nommé`thingOne`et spécifie l'Amazon Resource Name (ARN) du modèle de tâche à utiliser comme base de tâche. Vous pouvez remplacer les configurations avancées, telles que les configurations de délai d'expiration et d'annulation, en transmettant les paramètres associés de la propriété`create-job`Commande.

```
aws iot create-job \
      --targets arn:aws:iot:region:123456789012:thing/thingOne  \
      --job-template-arn arn:aws:iot:region:123456789012:jobtemplate/template-id
```

Appareils et tâches

Device communication with jobs

Les appareils peuvent communiquer avec le service AWS IoT Jobs via l'une de ces méthodes :

- MQTT
- HTTP Signature Version 4
- HTTP TLS

Using the MQTT protocol

La communication entre le service AWS IoTJobs et vos appareils peut se produire via le protocole MQTT. Les appareils s'abonnent aux rubriques MQTT afin d'être informés des nouvelles tâches et de recevoir les réponses du service AWS IoT Jobs. Les appareils publient sur les rubriques MQTT pour interroger ou mettre à jour l'état de l'exécution d'une tâche. Chaque appareil a sa propre rubrique MQTT générale. Pour plus d'informations sur la publication et l'abonnement aux rubriques MQTT, consultez the section called “Protocoles de communication de périphérique” (p. 79).

Note

Vous devez utiliser le point de terminaison correct lorsque vous communiquez avec le service AWS IoT Jobs via MQTT. Utilisez la commande DescribeEndpoint pour le trouver. Par exemple, si vous exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:Data-ATS
```

vous verrez une réponse comme celle-ci :

```
{
    "endpointAddress": "a1b2c3d4e5f6g7-ats.iot.us-west-2.amazonaws.com"
}
```

Avec cette méthode, votre appareil utilise le certificat et la clé privée qui lui sont propres pour s'authentifier auprès du service AWS IoT Jobs.

Les appareils peuvent :

- Être informé chaque fois qu'une exécution de tâche est ajoutée ou supprimée de la liste des exécutions de tâche en attente en s'abonnant à la rubrique MQTT `$aws/things/thing-name/jobs/notify`, où `thing-name` désigne le nom de l'objet associé à l'appareil.
- Être informés lorsque la prochaine exécution de tâche en attente est modifiée en s'abonnant à la rubrique MQTT `$aws/things/thing-name/jobs/notify-next`, où `thing-name` désigne le nom de l'objet associé à l'appareil.
- Mettre à jour le statut d'une exécution de tâche en appelant l'API UpdateJobExecution (p. 698).
- Interroger le statut d'une exécution de tâche en appelant l'API DescribeJobExecution (p. 693).
- Récupérer la liste des exécutions de tâche en attente en appelant l'API GetPendingJobExecutions (p. 684).
- Récupérer la prochaine exécution de tâche en attente en appelant l'API DescribeJobExecution (p. 693) avec le jobId `$next`.
- Obtenir et démarrer la prochaine exécution de tâche en attente en appelant l'API StartNextPendingJobExecution (p. 687).

Le service AWS IoT Jobs publie les messages de succès et d'échec dans une rubrique MQTT. La rubrique est formée en ajoutant `accepted` ou `rejected` à la rubrique utilisée pour effectuer la demande. Par exemple, si un message de demande est publié sur la rubrique `$aws/things/myThing/jobs/get`, le service AWS IoT Jobs publie les messages de succès sur la rubrique `$aws/things/myThing/jobs/get/accepted` et les messages de refus sur la rubrique `$aws/things/myThing/jobs/get/rejected`.

Using HTTP Signature Version 4

La communication entre le service AWS IoT Jobs et vos appareils peut se produire via le protocole HTTP Signature Version 4 sur le port 443. C'est la méthode utilisée par leAWSSDK et CLI. Pour plus d'informations sur ces outils, consultezAWS CLIRéférence de commande : iot-jobs-dataouAWSKits SDK et outilset reportez-vous à la section IotJobsDataPlane pour connaître votre langue préférée.

Note

Vous devez utiliser le point de terminaison correct lorsque vous communiquez avecAWS IoTTâches via HTTP Signature Version 4 ou à l'aide d'unAWSSDK ou CLIIotJobsDataPlaneCommande. Utilisez la commande DescribeEndpoint pour le trouver. Par exemple, si vous exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

vous verrez une réponse comme celle-ci :

```
{
    "endpointAddress": "a1b2c3d4e5f6g7.jobs.iot.us-west-2.amazonaws.com"
}
```

Avec cette méthode de communication, votre appareil utilise les informations d'identification IAM pour s'authentifier auprès du service AWS IoT Jobs.

Les commandes suivantes sont disponibles à l'aide de cette méthode :

- DescribeJobExecution

  ```
  aws iot-jobs-data describe-job-execution ...
  ```
- GetPendingJobExecutions

  ```
  aws iot-jobs-data get-pending-job-executions ...
  ```
- StartNextPendingJobExecution

  ```
  aws iot-jobs-data start-next-pending-job-execution ...
  ```
- UpdateJobExecution

  ```
  aws iot-jobs-data update-job-execution ...
  ```

Using HTTP TLS

La communication entre le service AWS IoT Jobs et vos appareils peut se produire via HTTP TLS sur le port 8443 à l'aide d'un client logiciel tiers prenant en charge ce protocole.

Note

Vous devez utiliser le point de terminaison correct lorsque vous communiquez avec le service AWS IoT Jobs via HTTP TLS. Utilisez la commande DescribeEndpoint pour le trouver. Par exemple, si vous exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

vous verrez une réponse comme celle-ci :

```
{
    "endpointAddress": "a1b2c3d4e5f6g7.jobs.iot.us-west-2.amazonaws.com"
}
```

Avec cette méthode, votre appareil utilise l'authentification basée sur le certificat X.509 (par exemple, à l'aide du certificat et de la clé privée qui lui sont propres.)

Les commandes suivantes sont disponibles à l'aide de cette méthode :

- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

Programmation des appareils pour une utilisation avec Jobs

Les exemples de cette section utilisent MQTT pour illustrer la façon dont un appareil utilise le service AWS IoT Jobs. Sinon, vous pouvez utiliser les commandes de l'interface de ligne de commande ou de l'API correspondante. Pour ces exemples, nous supposons qu'un appareil appelé `MyThing` s'abonne aux rubriques MQTT suivantes :

- `$aws/things/MyThing/jobs/notify` (ou `$aws/things/MyThing/jobs/notify-next`)
- `$aws/things/MyThing/jobs/get/accepted`
- `$aws/things/MyThing/jobs/get/rejected`
- `$aws/things/MyThing/jobs/jobId/get/accepted`
- `$aws/things/MyThing/jobs/jobId/get/rejected`

Si vous utilisez la signature de code pourAWS IoTLe code de votre appareil doit vérifier la signature de votre fichier de code. La signature se trouve dans le document de tâche dans la propriété `codesign`. Pour plus d'informations sur la vérification d'une signature de fichier de code, consultez Exemple d'agent d'appareil.

Flux de travail des appareils

Il existe deux manières dont un appareil peut traiter les tâches qu'il lui revient d'exécuter.

Option a: Get the next job

1. Lorsqu'un appareil est mis en ligne pour la première fois, il doit s'abonner à la rubrique `notify-next` de l'appareil.
2. Appelez l'API MQTT DescribeJobExecution (p. 693) avec le jobId `$next` pour obtenir la tâche suivante, son document de tâche et d'autres détails, y compris tout état enregistré dans `statusDetails`. Si le document de tâche possède une signature de fichier de code, vous devez vérifier la signature avant de continuer le traitement de la demande de tâche.
3. Appelez l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de la tâche. Ou, pour combiner cette étape et la précédente en un seul appel, l'appareil peut appeler StartNextPendingJobExecution (p. 687).
4. Le cas échéant, vous pouvez ajouter un minuteur d'étape en définissant une valeur pour `stepTimeoutInMinutes` lorsque vous appelez UpdateJobExecution (p. 698) ou StartNextPendingJobExecution (p. 687).
5. Exécutez les actions spécifiées par le document de tâche à l'aide de l'API MQTT UpdateJobExecution (p. 698) pour faire état de l'avancement de la tâche.
6. Continuez de surveiller l'exécution de tâche en appelant l'API MQTT DescribeJobExecution (p. 693) avec ce jobId. Si l'exécution de la tâche est supprimée,DescribeJobExecution (p. 693)renvoie un`ResourceNotFoundException`.

   Le périphérique doit être capable de reprendre un état valide si l'exécution de la tâche est annulée ou supprimée pendant que l'appareil exécute la tâche.

7. Une fois que la tâche est terminée, appelez l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de la tâche et faire état de sa réussite ou de son échec.
8. Comme le statut d'exécution de cette tâche est passé à un état terminal, la prochaine tâche disponible pour une exécution (le cas échéant) change également. L'appareil est averti que la prochaine exécution de tâche en attente a changé. À ce stade, l'appareil doit continuer comme décrit à l'étape 2.

Si l'appareil demeure en ligne, il continue à recevoir les notifications de la prochaine exécution de tâche en attente, y compris les données d'exécution de tâche, chaque fois qu'il a terminé une tâche ou qu'une nouvelle exécution de tâche en attente est ajoutée. Dans ce cas, l'appareil continue comme décrit à l'étape 2.

Option b: Pick from available jobs

1. Lorsqu'un appareil est mis en ligne pour la première fois, il doit s'abonner à la rubrique `notify` de l'objet.
2. Appelez l'API MQTT GetPendingJobExecutions (p. 684) pour obtenir la liste des exécutions de tâche en attente.
3. Si la liste contient une ou plusieurs exécutions de tâche, sélectionnez-en une.
4. Appelez l'API MQTT DescribeJobExecution (p. 693) pour obtenir le document de tâche et autres détails, y compris tout état enregistré dans `statusDetails`.
5. Appelez l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de la tâche. Si, dans cette commande, le champ `includeJobDocument` a la valeur `true`, l'appareil peut ignorer l'étape précédente et récupérer le document de tâche à ce stade.
6. Le cas échéant, vous pouvez ajouter un minuteur d'étape en définissant une valeur pour `stepTimeoutInMinutes` lorsque vous appelez UpdateJobExecution (p. 698).
7. Exécutez les actions spécifiées par le document de tâche à l'aide de l'API MQTT UpdateJobExecution (p. 698) pour faire état de l'avancement de la tâche.
8. Continuez de surveiller l'exécution de tâche en appelant l'API MQTT DescribeJobExecution (p. 693) avec ce jobId. Si l'exécution de tâche est annulée ou supprimée pendant que l'appareil exécute la tâche, celui-ci doit être capable de reprendre un état valide.
9. Une fois que la tâche est terminée, appelez l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de la tâche et faire état de sa réussite ou de son échec.

Si l'appareil demeure en ligne, il est averti de toutes les exécutions de tâche en attente chaque fois qu'une nouvelle exécution de tâche en attente devient disponible. Dans ce cas, l'appareil continue comme décrit à l'étape 2.

Si l'appareil n'est pas en mesure d'exécuter la tâche, il doit appeler l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de la tâche avec la valeur `REJECTED`.

Démarrage d'une nouvelle tâche

New job notification

Lorsqu'une nouvelle tâche est créée, le service AWS IoTJobs publie un message sur la rubrique `$aws/things/thing-name/jobs/notify` pour chaque appareil cible.

More Information(1)

Le message contient les informations suivantes :

```
{
    "timestamp":1476214217017,
    "jobs":{
```

```
        "QUEUED":[{
            "jobId":"0001",
            "queuedAt":1476214216981,
            "lastUpdatedAt":1476214216981,
            "versionNumber" : 1
        }]
    }
}
```

L'appareil reçoit ce message sur la rubrique `'$aws/things/thingName/jobs/notify'` lorsque l'exécution de la tâche est en file d'attente.

Get job information

Pour obtenir plus d'informations sur l'exécution d'une tâche, l'appareil appelle l'API MQTT DescribeJobExecution (p. 693) avec le champ `includeJobDocument` défini sur `true` (valeur par défaut).

More Information(2)

Si la demande aboutit, le service AWS IoT Jobs publie un message sur la rubrique `$aws/things/MyThing/jobs/0023/get/accepted` :

```
{
    "clientToken" : "client-001",
    "timestamp" : 1489097434407,
    "execution" : {
        "approximateSecondsBeforeTimedOut": number,
        "jobId" : "023",
        "status" : "QUEUED",
        "queuedAt" : 1489097374841,
        "lastUpdatedAt" : 1489097374841,
        "versionNumber" : 1,
        "jobDocument" : {
            < contents of job document >
        }
    }
}
```

Note

Si la demande échoue, le service AWS IoTJobs publie un message sur la rubrique `$aws/things/MyThing/jobs/0023/get/rejected`.

L'appareil dispose désormais du document de tâche, qu'il peut utiliser pour effectuer les opérations distantes pour la tâche. Si le document de tâche contient une URL Amazon S3 présignée, l'appareil peut utiliser cette URL pour télécharger les fichiers requis pour la tâche.

Rapport du statut d'exécution de tâche

Update execution status

Au fur et à mesure que l'appareil exécute la tâche, il peut appeler l'API MQTT UpdateJobExecution (p. 698) pour mettre à jour le statut de l'exécution de la tâche.

More information (3)

Par exemple, un appareil peut mettre à jour le statut de l'exécution de tâche `IN_PROGRESS` en publiant le message suivant sur la rubrique `$aws/things/MyThing/jobs/0023/update` :

```
{
    "status":"IN_PROGRESS",
    "statusDetails": {
        "progress":"50%"
    },
    "expectedVersion":"1",
    "clientToken":"client001"
}
```

Jobs répond en publiant un message dans la rubrique `$aws/things/MyThing/jobs/0023/update/accepted` ou `$aws/things/MyThing/jobs/0023/update/rejected` :

```
{
    "clientToken":"client001",
    "timestamp":1476289222841
}
```

Le périphérique permet de combiner les deux demandes précédentes en appelant StartNextPendingJobExecution (p. 687). La prochaine exécution de tâche en attente est ainsi obtenue et démarrée et l'appareil peut mettre à jour le statut d'exécution de la tâche. La demande renvoie aussi le document de tâche lorsqu'il y a une exécution de tâche en attente.

Si la tâche contient une valeur TimeoutConfig, le minuteur d'avancement commence à s'exécuter. Vous pouvez aussi définir un minuteur d'étape pour une exécution de tâche en définissant une valeur pour `stepTimeoutInMinutes` lorsque vous appelez UpdateJobExecution. Le minuteur d'étape s'applique uniquement à l'exécution des tâches que vous mettez à jour. Vous pouvez définir une nouvelle valeur pour ce minuteur chaque fois que vous mettez à jour une exécution de tâche. Vous pouvez aussi créer un minuteur d'étape lorsque vous appelez StartNextPendingJobExecution. Si l'exécution de tâche demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle du minuteur d'étape, elle échoue et passe à l'état final `TIMED_OUT`. Le minuteur d'étape n'a aucun effet sur le minuteur d'avancement que vous définissez lorsque vous créez une tâche.

Le champ `status` peut avoir la valeur `IN_PROGRESS`, `SUCCEEDED` ou `FAILED`. Vous ne pouvez pas mettre à jour le statut d'une exécution de tâche qui est déjà dans un état terminal.

Report execution completed

Lorsque l'appareil a terminé l'exécution de la tâche, il appelle l'API MQTT UpdateJobExecution (p. 698). Si la tâche aboutit, définissez `status` sur `SUCCEEDED` et, dans le champ `statusDetails` de la charge utile du message, ajoutez d'autres informations sur la tâche sous la forme de paires nom-valeur. Les minuteurs d'avancement et d'étape prennent fin lorsque l'exécution de la tâche est terminée.

More Information(4)

Exemples :

```
{
    "status":"SUCCEEDED",
    "statusDetails": {
        "progress":"100%"
    },
    "expectedVersion":"2",
    "clientToken":"client-001"
}
```

Si la tâche n'a pas réussi, définissez `status` sur `FAILED` et, dans `statusDetails`, ajoutez les informations sur l'erreur qui s'est produite :

```
{
    "status":"FAILED",
    "statusDetails": {
        "errorCode":"101",
        "errorMsg":"Unable to install update"
    },
    "expectedVersion":"2",
    "clientToken":"client-001"
}
```

Note

L'attribut `statusDetails` peut contenir n'importe quel nombre de paires nom-valeur.

Lorsque le service AWS IoT Jobs reçoit cette mise à jour, il publie un message sur la rubrique `$aws/things/MyThing/jobs/notify` pour indiquer que l'exécution de la tâche est terminée :

```
{
    "timestamp":1476290692776,
    "jobs":{}
}
```

Tâches supplémentaires

Additional jobs

S'il existe d'autres exécutions de tâche en attente pour le périphérique, elles sont incluses dans le message publié sur `$aws/things/MyThing/jobs/notify`.

More Information(5)

Exemples :

```
{
    "timestamp":1476290692776,
    "jobs":{
        "QUEUED":[{
            "jobId":"0002",
            "queuedAt":1476290646230,
            "lastUpdatedAt":1476290646230
        }],
        "IN_PROGRESS":[{
            "jobId":"0003",
            "queuedAt":1476290646230,
            "lastUpdatedAt":1476290646230
        }]
    }
}
```

Notifications Jobs

Le service AWS IoT Jobs publie des messages MQTT dans des rubriques réservées lorsque des tâches sont en attente ou que la première exécution de tâche de la liste change. Les appareils peuvent suivre les tâches en attente en s'abonnant à ces rubriques.

Les notifications de tâche sont publiées dans les rubriques MQTT en tant que charges utilises JSON. Il existe deux types de notifications :

- Une `ListNotification` contient la liste de 10 exécutions de tâche en attente au plus. Les exécutions de tâche contenues dans cette liste ont chacune une valeur de statut `IN_PROGRESS` ou `QUEUED`. Elles sont triées par statut (les exécutions de tâche `IN_PROGRESS` précèdent les exécutions de tâche `QUEUED`), puis selon le moment auquel elles ont été mises en file d'attente.

  Une `ListNotification` est publiée chaque fois que l'une des conditions ci-dessous est remplie.
  - Une nouvelle exécution de tâche est mise en file d'attente ou passe à un statut qui n'est pas final (`IN_PROGRESS` ou `QUEUED`).
  - Une ancienne exécution d'état acquiert le statut final (`FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` ou `REMOVED`).
- Une `NextNotification` contient le résumé d'informations de l'exécution de tâche suivante dans la file d'attente.

  Une `NextNotification` est publiée chaque fois que la première exécution de tâche de la liste change.
  - Une nouvelle exécution de tâche est ajoutée à la liste au statut `QUEUED` et constitue le premier élément de la liste.
  - Le statut d'une exécution de tâche existante qui n'était pas le premier élément de la liste passe de `QUEUED` à `IN_PROGRESS` et devient le premier élément de la liste. (Cette situation se produit lorsque la liste ne contient aucune autre exécution de tâche `IN_PROGRESS` ou que l'exécution de tâche dont le statut passe de `QUEUED` à `IN_PROGRESS` a été mise en file d'attente avant toutes les exécutions de tâche `IN_PROGRESS` de la liste.)
  - Le statut de l'exécution de tâche qui est la première de la liste passe au statut final et est supprimée de la liste.

Pour plus d'informations sur la publication et l'abonnement aux rubriques MQTT, consultez the section called “Protocoles de communication de périphérique” (p. 79).

> **Note**
>
> Les notifications ne sont pas disponibles lorsque vous utilisez HTTP Signature Version 4 ou HTTP TLS pour communiquer avec les tâches.

Job pending

Le service AWS IoT Jobs publie un message dans une rubrique MQTT lorsqu'une tâche est ajoutée à la liste des exécutions de tâche en attente d'un objet ou qu'elle en est supprimée, ou que la première exécution de tâche de la liste change :

- `$aws/things/`*`thingName`*`/jobs/notify`
- `$aws/things/`*`thingName`*`/jobs/notify-next`

More Information(6)

Les messages contiennent les exemples de charge utile suivants :

`$aws/things/`*`thingName`*`/jobs/notify`:

```
{
  "timestamp" : 10011,
  "jobs" : {
    "IN_PROGRESS" : [ {
      "jobId" : "other-job",
      "queuedAt" : 10003,
      "lastUpdatedAt" : 10009,
      "executionNumber" : 1,
      "versionNumber" : 1
    } ],
```

```
    "QUEUED" : [ {
      "jobId" : "this-job",
      "queuedAt" : 10011,
      "lastUpdatedAt" : 10011,
      "executionNumber" : 1,
      "versionNumber" : 0
    } ]
  }
}
```

`$aws/things/thingName/jobs/notify-next`:

```
{
  "timestamp" : 10011,
  "execution" : {
    "jobId" : "other-job",
    "status" : "IN_PROGRESS",
    "queuedAt" : 10009,
    "lastUpdatedAt" : 10009,
    "versionNumber" : 1,
    "executionNumber" : 1,
    "jobDocument" : {"c":"d"}
  }
}
```

Les valeurs possibles d'état d'exécution de tâche sont `QUEUED`, `IN_PROGRESS`, `FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` et `REMOVED`.

La série d'exemples ci-dessous présente les différents types de notifications qui sont publiées dans une rubrique au fur et à mesure que des exécutions de tâche sont créées et qu'elles passent d'un statut à un autre.

D'abord, une tâche appelée `job1` est créée. Cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517016948,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Cette notification est publiée dans la rubrique `jobs/notify-next` :

```
{
  "timestamp": 1517016948,
  "execution": {
    "jobId": "job1",
    "status": "QUEUED",
    "queuedAt": 1517016947,
    "lastUpdatedAt": 1517016947,
    "versionNumber": 1,
    "executionNumber": 1,
```

```
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Lorsqu'une autre tâche (`job2`) est créée, cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517017192,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Aucune notification n'est pas publiée dans la rubrique `jobs/notify-next`, car la tâche suivante de la file d'attente (`job1`) n'a pas changé. Lorsque `job1` commence à s'exécuter, son statut devient `IN_PROGRESS`. Aucune notification n'est publiée, car la liste des tâches et la tâche suivante dans la file d'attente n'ont pas changé.

Lorsqu'une troisième tâche (`job3`) est ajoutée, cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517017906,
  "jobs": {
    "IN_PROGRESS": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517017472,
        "startedAt": 1517017472,
        "executionNumber": 1,
        "versionNumber": 2
      }
    ],
    "QUEUED": [
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job3",
```

```
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517017905,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

Aucune notification n'est publiée dans la rubrique `jobs/notify-next`, car la tâche suivante de la file d'attente est toujours `job1`.

Une fois la tâche `job1` terminée, son statut passe à `SUCCEEDED` et cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517186269,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517017905,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

À ce stade, la tâche `job1` a été supprimée de la file d'attente et la prochaine tâche à s'exécuter est `job2`. Cette notification est publiée dans la rubrique `jobs/notify-next` :

```
{
  "timestamp": 1517186269,
  "execution": {
    "jobId": "job2",
    "status": "QUEUED",
    "queuedAt": 1517017191,
    "lastUpdatedAt": 1517017191,
    "versionNumber": 1,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Si la tâche `job3` doit commencer à s'exécuter avant la tâche `job2` (ce qui n'est pas recommandé), le statut de la tâche `job3` peut être modifié pour devenir `IN_PROGRESS`. Dans cette éventualité, `job2` n'est plus la tâche suivante dans la file d'attente et cette notification est publiée dans la rubrique `jobs/notify-next` :

```
{
```

```
  "timestamp": 1517186779,
  "execution": {
    "jobId": "job3",
    "status": "IN_PROGRESS",
    "queuedAt": 1517017905,
    "startedAt": 1517186779,
    "lastUpdatedAt": 1517186779,
    "versionNumber": 2,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

Aucune notification n'est publiée dans la rubrique `jobs/notify`, car aucune tâche n'a été ajoutée ou supprimée.

Si l'appareil rejette la tâche `job2` et met à jour son statut sur `REJECTED`, cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517189392,
  "jobs": {
    "IN_PROGRESS": [
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517186779,
        "startedAt": 1517186779,
        "executionNumber": 1,
        "versionNumber": 2
      }
    ]
  }
}
```

Si la tâche `job3` (toujours en cours) est supprimée de force, cette notification est publiée dans la rubrique `jobs/notify` :

```
{
  "timestamp": 1517189551,
  "jobs": {}
}
```

À ce stade, la file d'attente est vide. Cette notification est publiée dans la rubrique `jobs/notify-next` :

```
{
  "timestamp": 1517189551
}
```

Utilisation des API de tâche AWS IoT

Il existe deux catégories d'API utilisées dans le service AWS IoT Jobs :

- Celles utilisées pour la gestion et le contrôle des tâches.

- Celles utilisées par les appareils exécutant ces tâches.

En général, la gestion et le contrôle des tâches utilisent une API du protocole HTTPS. Les appareils peuvent utiliser une API MQTT ou une API du protocole HTTPS. (L'API HTTPS est conçue pour un faible volume d'appels lors de la création et du suivi des tâches. Elle ouvre généralement une connexion pour une seule demande, puis la ferme après réception de la réponse. L'API MQTT permet les interrogations longues. Elle est conçue pour d'importantes quantités de trafic qui peuvent atteindre plusieurs millions d'appareils.)

> **Note**
>
> Chaque API HTTPS AWS IoT Jobs dispose d'une commande correspondante qui vous permet d'appeler l'API depuis l'AWS CLI. Les commandes sont en minuscules, avec un trait d'union entre les mots qui composent le nom de l'API. Par exemple, vous pouvez appeler l'API `CreateJob` sur l'interface de ligne de commande en tapant :

```
aws iot create-job ...
```

API de gestion et de contrôle des tâches

Les commandes suivantes sont disponibles pour la gestion et le contrôle des Job dans l'interface de ligne de commande et via le protocole HTTPS.

- AssociateTargetsWithJob (p. 626)
- CancelJob (p. 628)
- CancelJobExecution (p. 630)
- CreateJob (p. 633)
- DeleteJob (p. 642)
- DeleteJobExecution (p. 645)
- DescribeJob (p. 648)
- DescribeJobExecution (p. 655)
- GetJobDocument (p. 659)
- ListJobExecutionsForJob (p. 660)
- ListJobExecutionsForThing (p. 663)
- ListJobs (p. 666)
- UpdateJob (p. 669)

Recherchez la valeur*endpoint-url*pour vos commandes CLI à l'aide de cette commande.

```
aws iot describe-endpoint --endpoint-type=iot:Jobs
```

Cette commande renvoie la sortie suivante.

```
{
"endpointAddress": "account-specific-prefix.jobs.iot.aws-region.amazonaws.com"
}
```

> **Note**
>
> Le point de terminaison Jobs ne prend pas en charge ALPN`z-amzn-http-ca`.

AssociateTargetsWithJob

AssociateTargetsWithJob command

Associe un groupe à une tâche continue. Pour plus d'informations, consultez CreateJob (p. 633). Les critères suivants doivent être satisfaits :

- Lors de la création de la tâche, le champ `targetSelection` doit être défini sur `CONTINUOUS`.
- Le statut de la tâche doit actuellement être `IN_PROGRESS`.
- Le nombre total de cibles associées à une tâche ne doit pas dépasser 100.

HTTPS (1)

Requête:

```
POST /jobs/jobId/targets

{
"targets": [ "string" ],
"comment": "string"
}
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`targets`

Liste des ARN du groupe d'objets qui définissent les cibles de la tâche.

`comment`

Facultatif. Chaîne de commentaire qui décrit la raison pour laquelle la tâche a été associée aux cibles.

Réponse:

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

`jobArn`

ARN identifiant la tâche.

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`description`

Brève description de la tâche.

CLI (1)

Résumé :

```
aws iot  associate-targets-with-job \
--targets <value> \
--job-id <value> \
[--comment <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"targets": [
"string"
],
"jobId": "string",
"comment": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
targets	liste membre : TargetArn	Liste des ARN du groupe d'objets qui définissent les cibles de la tâche.
TargetArn	chaîne	
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
comment	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Chaîne facultative qui décrit la raison pour laquelle la tâche a été associée aux cibles.

Sortie :

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
jobArn	chaîne	ARN identifiant la tâche.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Nom	Type	Description
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Brève description de la tâche.

CancelJob

CancelJob command

Annule une tâche.

HTTPS (2)

Requête:

```
PUT /jobs/jobId/cancel

{
"force": boolean,
"comment": "string",
"reasonCode": "string"
}
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`force`

[Facultatif] Si `true`, les exécutions de tâche avec l'état `IN_PROGRESS` ou `QUEUED` sont annulées. Sinon, seules les exécutions de tâche avec l'état `QUEUED` sont annulées. La valeur par défaut est `false`.

Warning

L'annulation d'une tâche dont le statut est `IN_PROGRESS` empêche l'appareil exécutant la tâche de mettre à jour le statut d'exécution de tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

`comment`

[Facultatif] Chaîne de commentaire décrivant la raison pour laquelle la tâche a été annulée.

`reasonCode`

[Facultatif] Une chaîne de code de motif qui explique pourquoi la tâche a été annulée. Si une tâche est annulée parce qu'elle répond aux conditions définies par un `abortConfig`, ce champ est renseigné automatiquement.

Réponse:

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

`jobArn`

ARN de la tâche.

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`description`

Brève description de la tâche.

CLI (2)

Résumé :

```
aws iot  cancel-job \
--job-id <value> \
[--force <value>]  \
[--comment <value>]  \
[--reasonCode <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"force": boolean,
"comment": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
force	boolean	Si la valeur est true, les tâches avec le statut QUEUED et IN_PROGRESS sont annulées. Sinon, seules les tâches avec l'état QUEUED sont annulées. **Warning** L'annulation d'une tâche dont le statut est IN_PROGRESS empêche l'appareil exécutant la tâche de mettre à jour le statut d'exécution de tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une

Nom	Type	Description
		tâche annulée est en mesure de reprendre un état valide.
comment	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Chaîne facultative décrivant la raison pour laquelle la tâche a été annulée.
reasonCode	chaîne longueur max. : 128 modèle : [\p{Upper}\p{Digit}_]+	Chaîne facultative expliquant pourquoi la tâche a été annulée. Si une tâche est annulée parce qu'elle répond aux conditions définies par `abortConfig`, ce champ est renseigné automatiquement.

Sortie :

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
jobArn	chaîne	ARN de la tâche.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Brève description de la tâche.

CancelJobExecution

CancelJobExecution command

Annule une exécution de tâche sur un appareil.

HTTPS (3)

Requête:

```
PUT /things/thingName/jobs/jobId/cancel
```

```
{
"force": boolean,
"expectedVersion": "string",
"statusDetails": {
    "string": "string"
    ...
}
}
```

`thingName`

Nom de l'objet dont l'exécution de tâche sera annulée.

`jobId`

Identifiant unique que vous avez attribué à la tâche lors de sa création.

`force`

Facultatif. Si la valeur est `true`, une exécution de tâche avec un statut `IN_PROGRESS` ou `QUEUED` peut être annulée. Sinon, seule une exécution de tâche avec un statut `QUEUED` peut être annulée. Si vous essayez d'annuler une exécution de tache dont le statut est `IN_PROGRESS` et que vous ne définissez pas `force` sur `true`, une exception `InvalidStateTransitionException` est levée. La valeur par défaut est `false`.

> Warning
>
> L'annulation d'une tâche dont le statut est `IN_PROGRESS` empêche l'appareil exécutant la tâche de mettre à jour le statut d'exécution de tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

`expectedVersion`

Facultatif. Version actuelle attendue de l'exécution de tâche. Sa version est incrémentée à chaque mise à jour de l'exécution de tâche. Si la version de l'exécution de tâche stockée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une erreur `VersionConflictException` et un élément `ErrorResponse` contenant le statut de l'exécution de tâche en cours est renvoyé. (Il est donc inutile d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche.)

`statusDetails`

Facultatif. Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.

Réponse:

```
{
}
```

CLI (3)

Résumé :

```
aws iot  cancel-job-execution \
--job-id <value> \
--thing-name <value> \
[--force | --no-force] \
[--expected-version <value>] \
[--status-details <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"thingName": "string",
"force": boolean,
"expectedVersion": long,
"statusDetails": {
"string": "string"
}
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Tâche à annuler.
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet dont l'exécution de la tâche sera annulée.
force	boolean	Facultatif. Si la valeur est `true`, l'exécution de tâche est annulée si son état est IN_PROGRESS ou QUEUED). Sinon, l'exécution de tâche est annulée uniquement si son état est QUEUED. Cependant, si vous essayez d'annuler une exécution de tache dont le statut est IN_PROGRESS et que vous ne définissez pas `--force` sur `true`, une exception `InvalidStateTransitionException` est déclenchée. La valeur par défaut est `false`. **Warning** L'annulation d'une tâche dont le statut est IN_PROGRESS empêche l'appareil exécutant la tâche de mettre à jour le statut d'exécution de tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

Nom	Type	Description
expectedVersion	long classe Java : java.lang.Long	Facultatif. Version actuelle attendue de l'exécution de tâche. Sa version est incrémentée à chaque mise à jour de l'exécution de tâche. Si la version de l'exécution de tâche stockée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une erreur `VersionMismatch` et un élément `ErrorResponse` contenant le statut de l'exécution de tâche en cours est renvoyé. (Il est donc inutile d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche.)
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations `statusDetails` demeurent inchangées.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	

Sortie :

Aucun

CreateJob

CreateJob command

Crée une tâche. Vous pouvez fournir le document de tâche comme lien vers un fichier dans un compartiment Amazon S3 (`documentSource`Paramètre) ou dans le corps de la demande (`document`).

Une tâche peut être rendue continue en définissant le paramètre facultatif `targetSelection` sur `CONTINUOUS`. (La valeur par défaut est `SNAPSHOT`.) Une tâche continue peut être utilisée pour intégrer les appareils ou les mettre à niveau au fur et à mesure qu'ils sont ajoutés à un groupe, car elle

continue de s'exécuter et est exécutée sur les objets nouvellement ajoutés, même après que les objets du groupe au moment où la tâche a été créé ont terminé la tâche.

Une tâche peut avoir une valeur TimeoutConfig facultative, qui définit la valeur du minuteur d'avancement. Le minuteur d'avancement ne peut pas être mis à jour et s'applique à toutes les exécutions de la tâche.

Les validations suivantes sont effectuées sur les arguments de l'API `CreateJob` :

- L'argument `targets` doit être une liste d'ARN d'objets ou de groupes d'objets valides. Toutes les choses et groupes d'objets doivent être dans votre Compte AWS .
- La `.documentSource`doit être une URL Amazon S3 valide pour un document de tâche. Les URL Amazon S3 se présentent sous la forme :`https://s3.amazonaws.com/`*`bucketName`*`/`*`objectName`*.
- Le document stocké dans l'URL spécifiée par l'argument `documentSource` doit être un document JSON codé en UTF-8.
- La taille d'un document de tâche est limité à 32 Ko en raison de la limite de la taille d'un message MQTT (128 Ko) et du chiffrement.
- La `.jobId`doit être unique au sein de votre Compte AWS .

HTTPS (4)

Requête:

```
PUT /jobs/jobId

{
"targets": [ "string" ],
"document": "string",
"documentSource": "string",
"description": "string",
"jobTemplateArn": "string",
"presignedUrlConfigData": {
    "roleArn": "string",
    "expiresInSec": "integer"
},
"targetSelection": "CONTINUOUS|SNAPSHOT",
"jobExecutionsRolloutConfig": {
    "exponentialRate": {
       "baseRatePerMinute": integer,
       "incrementFactor": integer,
       "rateIncreaseCriteria": {
          "numberOfNotifiedThings": integer, // Set one or the other
          "numberOfSucceededThings": integer // of these two values.
       },
       "maximumPerMinute": integer
  }
},
"abortConfig": {
   "criteriaList": [
      {
         "action": "string",
         "failureType": "string",
         "minNumberOfExecutedThings": integer,
         "thresholdPercentage": integer
      }
   ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": long
}
```

```
}
```

`jobId`

Identifiant de tâche, qui doit être unique pour votre Compte AWS . Nous vous recommandons d'utiliser un UUID. Les caractères alphanumériques, « - » et « _ » peuvent être utilisés ici.

`targets`

Liste des ARN d'objets ou de groupes d'objets qui définit les cibles de la tâche.

`document`

Facultatif. Document de tâche.

`documentSource`

Facultatif. Lien Amazon S3 vers le document de tâche.

`description`

Facultatif. Brève description de la tâche.

`jobTemplateArn`

ARN du modèle de tâche utilisé pour créer la tâche.

`presignedUrlConfigData`

Facultatif. Informations de configuration pour les URL Amazon S3 présignées.

`roleArn`

ARN du rôle IAM qui contient les autorisations pour accéder au compartiment Amazon S3. Il s'agit du compartiment qui contient les données que les appareils téléchargent avec les URL Amazon S3 présignées. Le rôle doit aussi accorder à AWS IoT l'autorisation d'endosser le rôle. Pour plus d'informations, consultez Création de tâches (p. 599).

`expiresInSec`

Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.

`targetSelection`

Facultatif. Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si la valeur est CONTINUOUS, la tâche peut aussi être planifiée pour s'exécuter sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche est planifiée pour s'exécuter sur un objet lorsque ce dernier est ajouté à un groupe cible, même après que la tâche a été effectuée par tous les objets qui se trouvaient à l'origine dans le groupe.

`jobExecutionRolloutConfig`

Facultatif. Permet de créer un déploiement étalé d'une tâche.

`maximumPerMinute`

Nombre maximal d'objets sur lesquels la tâche est envoyée pour l'exécution, par minute. Valeurs valides : de 1 à 1000. Si la valeur n'est pas spécifiée, la valeur par défaut est 1000. Le nombre réel d'objets qui reçoivent la tâche peut être inférieur pendant un intervalle de minutes particulier (en raison de la latence système), mais ne peut pas être supérieur à celui de la valeur spécifiée.

`exponentialRate`

Permet de définir la fréquence exponentielle de déploiement d'une tâche.

`baseRatePerMinute`

Nombre minimal d'objets qui sont informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.

`incrementFactor`

Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.

`rateIncreaseCriteria`

Critères pour initier l'augmentation du taux de déploiement pour une tâche. Définissez les valeurs de `numberOfNotifiedThings` ou `numberOfSucceededThings`, mais pas les deux.

`numberOfNotifiedThings`

Seuil du nombre d'objets notifiés qui déclenche l'augmentation de la fréquence de déploiement.

`numberOfSucceededThings`

Seuil du nombre d'objets réussis qui déclenche l'augmentation de la fréquence de déploiement.

`abortConfig`

Facultatif. Détails des critères d'annulation d'une tâche.

`criteriaList`

Liste de critères d'annulation afin de définir des règles pour annuler la tâche.

`action`

Type d'action d'annulation pour lancer une annulation de tâche.

`failureType`

Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.

`minNumberOfExecutedThings`

Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.

`thresholdPercentage`

Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche.

`timeoutConfig`

Facultatif. Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.

`inProgressTimeoutInMinutes`

Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Un minuteur est démarré ou redémarré, chaque fois que l'état d'exécution de la tâche est spécifié comme `IN_PROGRESS` avec ce champ renseigné. Si le statut d'exécution de la tâche n'est pas défini sur un état terminal avant que le minuteur n'expire, ou qu'une autre mise à jour d'état d'exécution de tâche ne soit envoyée avec ce champ renseigné, le statut est automatiquement défini sur `TIMED_OUT`.

Réponse:

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

`jobArn`

ARN de la tâche.

`jobId`

Identifiant unique que vous avez affecté à cette tâche.

`description`

Description brève et facultative de la tâche.

CLI (4)

Résumé :

```
aws iot  create-job \
--job-id <value> \
--targets <value> \
[--document-source <value>] \
[--document <value>] \
[--description <value>] \
[--job-template-arn <value>] \
[--presigned-url-config <value>] \
[--target-selection <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--document-parameters <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"targets": [
"string"
],
"documentSource": "string",
"document": "string",
"description": "string",
"jobTemplateArn": "string",
"presignedUrlConfig": {
"roleArn": "string",
"expiresInSec": long
},
"targetSelection": "string",
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
     "baseRatePerMinute": integer,
      "incrementFactor": integer,
      "rateIncreaseCriteria": {
         "numberOfNotifiedThings": integer, // Set one or the other
         "numberOfSucceededThings": integer // of these two values.
```

```
      },
  "maximumPerMinute": integer
  }
},
"abortConfig": {
 "criteriaList": [
    {
       "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": integer,
        "thresholdPercentage": integer
     }
  ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": long
},
"documentParameters": {
"string": "string"
}
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant de tâche, qui doit être unique pour votre Compte AWS . Nous vous recommandons d'utiliser un UUID. Les caractères alphanumériques, « - » et « _ » peuvent être utilisés ici.
targets	liste membre : TargetArn	Liste des objets et des groupes d'objets auxquels la tâche doit être envoyée.
TargetArn	chaîne	
documentSource	chaîne longueur max. : 1 350 min. : 1	Lien S3 vers le document de tâche.
document	chaîne longueur max. : 32 768	Document de tâche.
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Brève description de la tâche.
JobTemplateArn	chaîne longueur max. : 2 600 motif : ^arn : [! -~] +$	ARN du modèle de tâche utilisé pour créer la tâche.
presignedUrlConfig	PresignedUrlConfig	Informations de configuration pour les URL S3 présignées.

Nom	Type	Description
roleArn	chaîne longueur max. : 2 048 min. : 20	ARN d'un rôle IAM qui octroie l'autorisation de télécharger les fichiers depuis le compartiment Amazon S3 où les données de tâche ou les mises à jour sont stockées. Le rôle doit également accorder l'autorisation à AWS IoT de télécharger les fichiers.
expiresInSec	long classe Java : java.lang.Long plage - max. : 3 600 min. : 60	Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.
targetSelection	chaîne enum : CONTINU | INSTANTANÉ	Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si elle est continue, la tâche peut également être exécutée sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un objet lorsque ce dernier est ajouté à un groupe cible, même après que la tâche a été exécutée par tous les objets initialement dans le groupe.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Permet de créer un déploiement étalé de la tâche.
maximumPerMinute	entier classe Java : java.lang.Integer plage - max : 1 000 min. : 1	Nombre maximal d'objets qui sont informés de la présence d'une tâche en attente, par minute. Ce paramètre vous permet de créer un déploiement étalé.
exponentialRate	ExponentialRolloutRate	Taux d'augmentation pour le déploiement d'une tâche. Ce paramètre vous permet de définir un taux exponentiel pour le déploiement d'une tâche.

Nom	Type	Description
baseRatePerMinute	classe Java : java.lang.Integer	Nombre minimal d'objets qui seront informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.
incrementFactor	classe java : java.lang.Double	Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.
rateIncreaseCriteria	RateIncreaseCriteria	Permet de définir un critère pour initier l'augmentation de la fréquence de déploiement pour une tâche. Définissez les valeurs de `numberOfNotifiedThings` ou `numberOfSucceededThings`, mais pas les deux.
numberOfNotifiedThings	classe java : java.lang.Double	Seuil du nombre d'objets notifiés qui lancera l'augmentation du taux de déploiement.
numberOfSucceededThings	classe java : java.lang.Double	Seuil du nombre d'objets réussis qui lancera l'augmentation du taux de déploiement.
abortConfig	AbortConfig	Vous permet de créer des critères pour annuler une tâche.
criteriaList	AbortCriteria	Liste de critères d'annulation afin de définir des règles pour annuler la tâche.
action	classe java : java.lang.String (CANCEL)	Type d'action d'annulation pour lancer une annulation de tâche.
failureType	classe java : java.lang.String (FAILED | REJECTED | TIMED_OUT | ALL)	Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.
minNumberOfExecutedThings	classe Java : java.lang.Integer)	Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.

Nom	Type	Description
thresholdPercentage	classe java : java.lang.Double)	Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche. AWS IoT prend en charge jusqu'à deux chiffres après la virgule (par exemple, 10,9 et 10,99, mais pas 10,999).
timeoutConfig	TimeoutConfig	Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.
inProgressTimeoutInMinutes	long	Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Un minuteur est démarré ou redémarré, chaque fois que l'état d'exécution de la tâche est spécifié comme `IN_PROGRESS` avec ce champ renseigné. Si le statut d'exécution de la tâche n'est pas défini sur un état terminal avant que le minuteur n'expire, ou qu'une autre mise à jour d'état d'exécution de tâche ne soit envoyée avec ce champ renseigné, le statut est automatiquement défini sur `TIMED_OUT`.
documentParameters	map Clé : ParameterKey Valeur : ParameterValue	Paramètres pour le document de tâche.
ParameterKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	

Nom	Type	Description
ParameterValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]+	

Sortie :

```
{
"jobArn": "string",
"jobId": "string",
"description": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
jobArn	chaîne	ARN de la tâche.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez affecté à cette tâche.
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Description de la tâche.

DeleteJob

DeleteJob command

Supprime une tâche et ses exécutions de tâche associées.

Selon le nombre d'exécutions de tâche créées pour la tâche et divers autres facteurs, la suppression d'une tâche peut prendre du temps. Pendant la suppression de la tâche, l'état de celle-ci indique « DELETION_IN_PROGRESS ». Toute tentative de suppression ou d'annulation d'une tâche dont le statut est « DELETION_IN_PROGRESS » entraîne une erreur.

HTTPS (5)

Syntaxe de la demande :

```
DELETE /jobs/jobId?force=force
```

Paramètres de demande URI :

Nom	Type	Dem ?	Description
jobId	JobId	oui	ID de la tâche à supprimer.

Nom	Type	Dem ?	Description
force	ForceFlag	non	(Facultatif) Lorsque ce paramètre a la valeur true, vous pouvez supprimer un exécution de tâche dont le statut est « IN_PROGRESS ». Sinon, vous pouvez uniquement supprimer une tâche qui se trouve dans un état terminal (« SUCCEEDED » ou « CANCELED »), ou bien une exception se produit. La valeur par défaut est false. Note La suppression d'une tâche dont le statut est « IN_PROGRESS » empêche l'appareil qui exécute la tâche d'accéder aux informations de cette dernière ou de mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

Erreurs:

`InvalidRequestException`

Le contenu de la demande n'était pas valide. Par exemple, ce code est renvoyé lorsqu'une demande UpdateJobExecution contient des détails d'état non valides. Le message contient des détails sur l'erreur.

Code de réponse HTTP : code 400

`InvalidStateTransitionException`

Une mise à jour a tenté de faire passer la tâche ou l'exécution de tâche dans un état non valide en raison de son état actuel (par exemple, une tentative de faire passer une demande de l'état SUCCEEDED à l'état IN_PROGRESS). Dans ce cas, le corps du message d'erreur contient aussi le champ executionState.

Code de réponse HTTP : code 409

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

Code de réponse HTTP : code 404

`ThrottlingException`

Le tarif dépasse la limite.

Code de réponse HTTP : code 429

`ServiceUnavailableException`

Le service est temporairement indisponible.

Code de réponse HTTP : code 503

CLI (5)

Résumé :

```
aws iot  delete-job \
--job-id <value> \
[--force | --no-force]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"force": boolean
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	ID de la tâche à supprimer.
force	boolean	(Facultatif) Lorsque ce paramètre a la valeur true, vous pouvez supprimer un exécution de tâche dont le statut est « IN_PROGRESS ». Sinon, vous pouvez uniquement supprimer une tâche qui

Nom	Type	Description
		se trouve dans un état terminal (« SUCCEEDED » ou « CANCELED »), ou bien une exception se produit. La valeur par défaut est false. **Note** La suppression d'une tâche dont le statut est « IN_PROGRESS » empêche l'appareil qui exécute la tâche d'accéder aux informations de cette dernière ou de mettre à jour le statut d'exécution de la tâche. Soyez vigilant et vérifiez que chaque appareil exécutant une tâche annulée est en mesure de reprendre un état valide.

Sortie :

Aucun

DeleteJobExecution

DeleteJobExecution command

Supprime une exécution de tâche.

HTTPS (6)

Syntaxe de la demande :

```
DELETE /things/thingName/jobs/jobId/executionNumber/executionNumber?force=force
```

Paramètres de demande URI :

Nom	Type	Dem ?	Description
jobId	JobId	oui	ID de la tâche dont l'exécution sera supprimée.
thingName	ThingName	oui	Nom de l'objet dont l'exécution de la tâche sera supprimée.
executionNumber	ExecutionNumber	oui	ID de l'exécution de tâche à supprimer.

Nom	Type	Dem ?	Description
force	ForceFlag	non	Lorsque ce paramètre a la valeur true, vous pouvez supprimer une exécution de tâche dont le statut est « IN_PROGRESS ». Sinon, vous pouvez uniquement supprimer une exécution de tâche qui se trouve dans un état terminal (SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED ou CANCELED) ou une exception se produit. La valeur par défaut est false. Note La suppression d'une exécution de tâche dont le statut est « IN_PROGRESS » empêche l'appareil d'accéder aux informations de la tâche ou de mettre à jour son statut d'exécution. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Erreurs:

`InvalidRequestException`

Le contenu de la demande n'était pas valide. Par exemple, ce code est renvoyé lorsqu'une demande UpdateJobExecution contient des détails d'état non valides. Le message contient des détails sur l'erreur.

Code de réponse HTTP : code 400

`InvalidStateTransitionException`

Une mise à jour a tenté de faire passer l'exécution de tâche dans un état qui n'est pas valide en raison de l'état actuel de l'exécution de tâche (par exemple, une tentative de modification d'une demande de l'état SUCCEEDED à l'état IN_PROGRESS). Dans ce cas, le corps du message d'erreur contient aussi le champ `executionState`.

Code de réponse HTTP : code 409

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

Code de réponse HTTP : code 404

`ThrottlingException`

Le tarif dépasse la limite.

Code de réponse HTTP : code 429

`ServiceUnavailableException`

Le service est temporairement indisponible.

Code de réponse HTTP : code 503

CLI (6)

Résumé :

```
aws iot  delete-job-execution \
--job-id <value> \
--thing-name <value> \
--execution-number <value> \
[--force | --no-force]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"thingName": "string",
"executionNumber": long,
"force": boolean
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	ID de la tâche dont l'exécution sera supprimée.
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet dont l'exécution de la tâche sera supprimée.

Nom	Type	Description
executionNumber	long classe Java : java.lang.Long	ID de l'exécution de tâche à supprimer.
force	boolean	Lorsque ce paramètre a la valeur true, vous pouvez supprimer une exécution de tâche dont le statut est « IN_PROGRESS ». Sinon, vous pouvez uniquement supprimer une exécution de tâche qui se trouve dans un état terminal (SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED ou CANCELED) ou une exception se produit. La valeur par défaut est false. **Note** La suppression d'une exécution de tâche dont le statut est « IN_PROGRESS » empêche l'appareil d'accéder aux informations de la tâche ou de mettre à jour son statut d'exécution. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Sortie :

Aucun

DescribeJob

DescribeJob command

Obtient les détails de la tâche spécifiée.

HTTPS (7)

Requête:

```
GET /jobs/jobId
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Réponse:

```
{
"documentSource": "string",
"job": Job
}
```

`documentSource`

Lien Amazon S3 vers le document de tâche.

`job`

Objet Job (p. 676)

CLI (7)

Résumé :

```
aws iot  describe-job \
--job-id <value>  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Sortie :

```
{
"documentSource": "string",
"job": {
"jobArn": "string",
"jobId": "string",
"targetSelection": "string",
"status": "string",
"forceCanceled": boolean,
"comment": "string",
"targets": [
  "string"
],
"description": "string",
"jobTemplateArn": "string",
"presignedUrlConfig": {
  "roleArn": "string",
  "expiresInSec": long
},
```

```
"jobExecutionsRolloutConfig": {
    "exponentialRate": {
       "baseRatePerMinute": integer,
       "incrementFactor": integer,
       "rateIncreaseCriteria": {
          "numberOfNotifiedThings": integer, // Set one or the other
          "numberOfSucceededThings": integer // of these two values.
       },
       "maximumPerMinute": integer
  }
},
"abortConfig": {
   "criteriaList": [
      {
         "action": "string",
         "failureType": "string",
         "minNumberOfExecutedThings": integer,
         "thresholdPercentage": integer
      }
   ]
},
"createdAt": "timestamp",
"lastUpdatedAt": "timestamp",
"completedAt": "timestamp",
"jobProcessDetails": {
  "processingTargets": [
    "string"
  ],
  "numberOfCanceledThings": "integer",
  "numberOfSucceededThings": "integer",
  "numberOfFailedThings": "integer",
  "numberOfRejectedThings": "integer",
  "numberOfQueuedThings": "integer",
  "numberOfInProgressThings": "integer",
  "numberOfRemovedThings": "integer",
  "numberOfTimedOutThings": "integer"
},
"documentParameters": {
  "string": "string"
},
"timeoutConfig": {
   "inProgressTimeoutInMinutes": number
}
}
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
documentSource	chaîne longueur max. : 1 350 min. : 1	Lien Amazon S3 vers le document de tâche.
tâche	Job	Informations sur la tâche.
jobArn	chaîne	ARN identifiant la tâche au format « arn:aws:iot:region:account:job/jobId ».
jobId	chaîne longueur max. : 64 min. : 1	Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Nom	Type	Description
	modèle : [a-zA-Z0-9_-]+	
targetSelection	chaîne enum : CONTINU | INSTANTANÉ	Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si elle est continue, la tâche peut également être exécutée sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un appareil lorsque l'objet représentant ce dernier est ajouté à un groupe cible, même après que la tâche a été effectuée par tous les objets initialement dans le groupe.
status	chaîne enum : IN_PROGRESS | ANNULÉ | RÉUSSI	Statut de la tâche : `IN_PROGRESS`, `CANCELED` ou `SUCCEEDED`.
forceCanceled	boolean classe Java : java.lang.Boolean	Prend la valeur `true` si la tâche a été annulée avec le paramètre facultatif `force` défini sur `true`.
comment	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Si la tâche a été mise à jour, décrit la raison de la mise à jour.
targets	liste membre : TargetArn	Liste des objets et des groupes d'objets AWS IoT auxquels la tâche doit être envoyée.
TargetArn	chaîne	
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Brève description de la tâche.
JobTemplateArn	chaîne longueur max. : 2 600 motif : ^arn : [! -~] +$	ARN du modèle de tâche utilisé pour créer la tâche.
presignedUrlConfig	PresignedUrlConfig	Configuration pour les URL Amazon S3 présignées.

Nom	Type	Description
roleArn	chaîne longueur max. : 2 048 min. : 20	ARN d'un rôle IAM qui octroie l'autorisation de télécharger les fichiers depuis le compartiment Amazon S3 où les données de tâche ou les mises à jour sont stockées. Le rôle doit également accorder l'autorisation au service AWS IoT Jobs de télécharger les fichiers.
expiresInSec	long classe Java : java.lang.Long plage - max. : 3 600 min. : 60	Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Permet de créer un déploiement étalé de la tâche.
maximumPerMinute	entier classe Java : java.lang.Integer plage - max : 1 000 min. : 1	Nombre maximal d'objets qui sont informés de la présence d'une tâche en attente, par minute. Ce paramètre vous permet de créer un déploiement étalé.
exponentialRate	ExponentialRolloutRate	Taux d'augmentation pour le déploiement d'une tâche. Ce paramètre vous permet de définir un taux exponentiel pour le déploiement d'une tâche.
baseRatePerMinute	classe Java : java.lang.Integer	Nombre minimal d'objets qui sont informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.
incrementFactor	classe java : java.lang.Double	Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.

Nom	Type	Description
rateIncreaseCriteria	RateIncreaseCriteria	Permet de définir un critère pour initier l'augmentation de la fréquence de déploiement pour une tâche. Définissez les valeurs de `numberOfNotifiedThings` ou `numberOfSucceededThings`, mais pas les deux.
numberOfNotifiedThings	classe java : java.lang.Double	Seuil du nombre d'objets notifiés qui déclenche l'augmentation de la fréquence de déploiement.
numberOfSucceededThings	classe java : java.lang.Double	Seuil du nombre d'objets réussis qui déclenche l'augmentation de la fréquence de déploiement.
abortConfig	AbortConfig	Vous permet de créer des critères pour annuler une tâche.
criteriaList	AbortCriteria	Liste de critères d'annulation afin de définir des règles pour annuler la tâche.
action	classe java : java.lang.String (CANCEL)	Type d'action d'annulation pour lancer une annulation de tâche.
failureType	classe java : java.lang.String (FAILED | REJECTED | TIMED_OUT | ALL)	Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.
minNumberOfExecutedThings	classe Java : java.lang.Integer)	Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.
thresholdPercentage	classe java : java.lang.Double)	Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche. AWS IoT prend en charge jusqu'à deux chiffres après la virgule (par exemple, 10,9 et 10,99, mais pas 10,999).
createdAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été créée.
lastUpdatedAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été mise à jour.

Nom	Type	Description
completedAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été exécutée.
jobProcessDetails	JobProcessDetails	Détails sur le processus de tâche.
processingTargets	liste membre : ProcessingTargetName classe Java : java.util.List	Appareils sur lesquels la tâche est en cours d'exécution.
ProcessingTargetName	chaîne	
numberOfCanceledThings	entier classe Java : java.lang.Integer	Nombre d'objets ayant annulé la tâche.
numberOfSucceededThings	entier classe Java : java.lang.Integer	Nombre d'objets ayant exécuté la tâche avec succès.
numberOfFailedThings	entier classe Java : java.lang.Integer	Nombre d'objets n'ayant pas réussi à exécuter la tâche.
numberOfRejectedThings	entier classe Java : java.lang.Integer	Nombre d'objets ayant rejeté la tâche.
numberOfQueuedThings	entier classe Java : java.lang.Integer	Nombre d'objets en attente d'exécution de la tâche.
numberOfInProgressThings	entier classe Java : java.lang.Integer	Nombre d'objets exécutant actuellement la tâche.
numberOfRemovedThings	entier classe Java : java.lang.Integer	Nombre d'objets qui ne sont plus programmés pour exécuter la tâche, car ils ont été supprimés ou retirés du groupe qui était une cible de la tâche.
numberOfTimedOutThings	entier classe Java : java.lang.Integer	Le nombre d'objets dont le statut de l'exécution de la tâche est `TIMED_OUT`.
documentParameters	map Clé : ParameterKey Valeur : ParameterValue	Paramètres spécifiés pour le document de tâche.

Nom	Type	Description
ParameterKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
ParameterValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]+	
timeoutConfig	TimeoutConfig	Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Un minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration du minuteur, il est défini sur `TIMED_OUT`.
inProgressTimeoutInMinutes	long	Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Le délai peut être n'importe quelle valeur comprise entre 1 minute et 7 jours (1 à 10 080 minutes). Le minuteur en cours ne peut pas être mis à jour et s'applique à toutes les exécutions de tâche pour la tâche. Chaque fois qu'une tâche d'exécution demeure dans l'état `IN_PROGRESS` plus longtemps que l'intervalle défini, l'exécution de la tâche échoue et passe à l'état final `TIMED_OUT`.

DescribeJobExecution

DescribeJobExecution command

Obtient les détails d'une exécution de tâche. Le statut de l'exécution de tâche doit être `SUCCEEDED` ou `FAILED`.

HTTPS (8)

Requête:

```
GET /things/thingName/jobs/jobId?executionNumber=executionNumber
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`thingName`

Nom de l'objet associé à l'appareil sur lequel l'exécution de tâche est en cours.

`executionNumber`

Facultatif. Nombre qui permet de spécifier une exécution de tâche sur un appareil. (Voir JobExecution (p. 680).) S'il n'est pas indiqué, la dernière exécution de tâche est renvoyée.

Réponse:

```
{
"execution": { JobExecution }
}
```

`execution`

Objet JobExecution (p. 680)

CLI (8)

Résumé :

```
aws iot  describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"thingName": "string",
"executionNumber": long
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
thingName	chaîne longueur max. : 128 min. : 1	Nom de l'objet sur lequel l'exécution de tâche est en cours.

Nom	Type	Description
	modèle : [a-zA-Z0-9:_-]+	
executionNumber	long classe Java : java.lang.Long	Chaîne (comprenant les chiffres 0 à 9), qui est utilisée pour spécifier une exécution de tâche particulière sur un appareil donné.

Sortie :

```
{
"execution": {
"approximateSecondsBeforeTimedOut": "number"
"jobId": "string",
"status": "string",
"forceCanceled": boolean,
"statusDetails": {
  "detailsMap": {
    "string": "string"
  }
},
"thingArn": "string",
"queuedAt": "timestamp",
"startedAt": "timestamp",
"lastUpdatedAt": "timestamp",
"executionNumber": long,
"versionNumber": long
}
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
execution	JobExecution	Informations sur l'exécution de tâche.
approximateSecondsBeforeTimedOut	Long	Estimation du nombre de secondes qui restent avant que le statut d'exécution de la tâche ne devienne `TIMED_OUT`. Le délai peut être n'importe quelle valeur comprise entre 1 minute et 7 jours (1 à 10 080 minutes). L'expiration réelle du délai d'exécution de la tâche peut intervenir jusqu'à 60 secondes plus tard que la durée prévue. Cette valeur ne sera pas incluse si l'exécution de la tâche a atteint un statut final.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à la tâche lors de sa création.

Nom	Type	Description
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	Statut de l'exécution de la tâche (IN_PROGRESS, QUEUED, FAILED, SUCCEEDED, TIMED_OUT, CANCELED ou REJECTED).
forceCanceled	boolean classe Java : java.lang.Boolean	Prend la valeur `true` si l'exécution de la tâche a été annulée avec le paramètre facultatif `force` défini sur `true`.
statusDetails	JobExecutionStatusDetails	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.
detailsMap	map Clé : DetailsKey Valeur : DetailsValue	Statut de l'exécution de tâche.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	
thingArn	chaîne	ARN de l'objet sur lequel l'exécution de tâche est en cours.
queuedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

Nom	Type	Description
executionNumber	long classe Java : java.lang.Long	Chaîne (comprenant les chiffres 0 à 9), qui identifie cette exécution de tâche particulière sur cet appareil. Elle peut être utilisée dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.

GetJobDocument

GetJobDocument command

Obtient le document de tâche pour une tâche.

Note

Les URL d'espace réservé ne sont pas remplacées par les URL Amazon S3 présignées du document retourné. Les URL présignées sont générées uniquement lorsque le service AWS IoT Jobs reçoit une demande via MQTT.

HTTPS (9)

Requête:

```
GET /jobs/jobId/job-document
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Réponse:

```
{
"document": "string"
}
```

`document`

Contenu du document de tâche.

CLI (9)

Résumé :

```
aws iot  get-job-document \
--job-id <value>  \
[--cli-input-json <value>] \
```

```
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.

Sortie :

```
{
"document": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
document	chaîne longueur max. : 32 768	Contenu du document de tâche.

ListJobExecutionsForJob

ListExecutionsForJob command

Obtient la liste des exécutions de tâche d'une tâche.

HTTPS (10)

Requête:

```
GET /jobs/jobId/things?status=status&maxResults=maxResults&nextToken=nextToken
```

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`status`

Facultatif. Filtre qui vous permet d'explorer les tâches ayant le statut spécifié : QUEUED, IN_PROGRESS, SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED ou CANCE

`maxResults`

Facultatif. Nombre maximal de résultats à renvoyer par demande.

`nextToken`

Facultatif. Jeton permettant de récupérer l'ensemble suivant de résultats.

Réponse:

```
{
"executionSummaries": [ JobExecutionSummary ... ]
}
```

`executionSummaries`

Liste d'objets JobExecutionSummary (p. 681) associés à l'ID de tâche spécifié.

CLI (10)

Résumé :

```
aws iot  list-job-executions-for-job \
--job-id <value> \
[--status <value>] \
[--max-results <value>] \
[--next-token <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"status": "string",
"maxResults": "integer",
"nextToken": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	Statut de la tâche.
maxResults	entier classe Java : java.lang.Integer plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer par demande.
nextToken	chaîne	Jeton permettant de récupérer l'ensemble suivant de résultats.

Sortie :

```
{
"executionSummaries": [
{
  "thingArn": "string",
  "jobExecutionSummary": {
    "status": "string",
    "queuedAt": "timestamp",
    "startedAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "executionNumber": long
  }
}
],
"nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
executionSummaries	liste membre : JobExecutionSummaryForJob classe Java : java.util.List	Liste des récapitulatifs d'exécution de tâche.
JobExecutionSummaryForJob	JobExecutionSummaryForJob	
thingArn	chaîne	ARN de l'objet sur lequel l'exécution de tâche est en cours.
jobExecutionSummary	JobExecutionSummary	Contient un sous-ensemble d'informations sur une exécution de tâche.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	Statut de l'exécution de tâche.
queuedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

Nom	Type	Description
executionNumber	long classe Java : java.lang.Long	Chaîne (comprenant les chiffres 0 à 9), qui identifie cette exécution de tâche particulière sur cet appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.
nextToken	chaîne	Jeton pour l'ensemble de résultats suivant, ou null s'il n'y a pas de résultats supplémentaires.

ListJobExecutionsForThing

ListJobExecutionsForThing command

Obtient la liste des exécutions de tâche d'un objet.

HTTPS (11)

Requête:

```
GET /things/thingName/jobs?status=status&maxResults=maxResults&nextToken=nextToken
```

`thingName`

Nom de l'objet pour lequel `JobExecutions` est affiché.

`status`

Filtre facultatif qui vous permet de rechercher les tâches qui ont le statut spécifié : QUEUED, IN_PROGRESS, SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED ou CANCE

`maxResults`

Nombre maximal de résultats à renvoyer par demande.

`nextToken`

Jeton pour l'ensemble de résultats suivant, ou `null` s'il n'y a pas de résultats supplémentaires.

Réponse:

```
{
"executionSummaries": [ JobExecutionSummary ... ]
}
```

`executionSummaries`

Liste des objets JobExecutionSummary (p. 681) pour les exécutions de tâche associées à l'objet spécifié.

CLI (11)

Résumé :

```
aws iot  list-job-executions-for-thing \
--thing-name <value> \
[--status <value>] \
[--max-results <value>] \
[--next-token <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"thingName": "string",
"status": "string",
"maxResults": "integer",
"nextToken": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	Filtre facultatif qui vous permet de rechercher les tâches qui ont le statut spécifié.
maxResults	entier classe Java : java.lang.Integer plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer par demande.
nextToken	chaîne	Jeton permettant de récupérer l'ensemble suivant de résultats.

Sortie :

```
{
"executionSummaries": [
{
  "jobId": "string",
  "jobExecutionSummary": {
    "status": "string",
    "queuedAt": "timestamp",
    "startedAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "executionNumber": long
  }
}
],
```

```
"nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
executionSummaries	liste membre : JobExecutionSummaryForThing classe Java : java.util.List	Liste des récapitulatifs d'exécution de tâche.
JobExecutionSummaryForThing	JobExecutionSummaryForThing	
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
jobExecutionSummary	JobExecutionSummary	Contient un sous-ensemble d'informations sur une exécution de tâche.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT| REJETÉ | SUPPRIMÉ | ANNULÉ	Statut de l'exécution de tâche.
queuedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	timestamp	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.
executionNumber	long classe Java : java.lang.Long	Chaîne (comprenant les chiffres 0 à 9), qui identifie cette exécution de tâche particulière sur cet appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.
nextToken	chaîne	Jeton pour l'ensemble de résultats suivant, ou null s'il n'y a pas de résultats supplémentaires.

ListJobs

ListJobs command

Obtient la liste des tâches de votre Compte AWS .

HTTPS (12)

Requête:

```
GET /jobs?
status=status&targetSelection=targetSelection&thingGroupName=thingGroupName&thingGroupId=thingGroup
```

`status`

Facultatif. Filtre qui vous permet d'explorer les tâches ayant le statut spécifié : IN_PROGRESS, ANNULATION ou SUCCISÉ.

`targetSelection`

Facultatif. Filtre qui vous permet de rechercher les tâches ayant la valeur spécifiée`targetSelection`Valeur : CONTINU ou instantané.

`thingGroupName`

Facultatif. Filtre qui vous permet de rechercher les tâches ayant le nom de groupe d'objets spécifié comme cible.

`thingGroupId`

Facultatif. Filtre qui vous permet de rechercher les tâches ayant l'ID de groupe d'objets spécifié comme cible.

`maxResults`

Facultatif. Nombre maximal de résultats à renvoyer par demande.

`nextToken`

Facultatif. Jeton permettant de récupérer l'ensemble suivant de résultats.

Réponse:

```
{
"jobs": [ JobSummary ... ],
}
```

`jobs`

Une liste deJobSummary (p. 679), un pour chaque tâche dans votre Compte AWS qui correspond aux critères de filtrage spécifiés.

CLI (12)

Résumé :

```
aws iot  list-jobs \
[--status <value>] \
[--target-selection <value>] \
[--max-results <value>] \
[--next-token <value>] \
```

```
[--thing-group-name <value>] \
[--thing-group-id <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"status": "string",
"targetSelection": "string",
"maxResults": "integer",
"nextToken": "string",
"thingGroupName": "string",
"thingGroupId": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
status	chaîne enum : IN_PROGRESS | ANNULÉ | RÉUSSI	Filtre facultatif qui vous permet de rechercher les tâches qui ont le statut spécifié.
targetSelection	chaîne enum : CONTINU | INSTANTANÉ	Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si elle est continue, la tâche peut également être exécutée sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un objet lorsque ce dernier est ajouté à un groupe cible, même après que la tâche a été exécutée par tous les objets initialement dans le groupe.
maxResults	entier classe Java : java.lang.Integer plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer par demande.
nextToken	chaîne	Jeton permettant de récupérer l'ensemble suivant de résultats.
thingGroupName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Filtre qui limite les tâches renvoyées à celles concernant le groupe spécifié.
thingGroupId	chaîne longueur max. : 128 min. : 1	Filtre qui limite les tâches renvoyées à celles concernant le groupe spécifié.

Nom	Type	Description
	modèle : [a-zA-Z0-9-]+	

Sortie :

```
{
"jobs": [
{
  "jobArn": "string",
  "jobId": "string",
  "thingGroupId": "string",
  "targetSelection": "string",
  "status": "string",
  "createdAt": "timestamp",
  "lastUpdatedAt": "timestamp",
  "completedAt": "timestamp"
}
],
"nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
jobs	liste membre : JobSummary classe Java : java.util.List	Liste de tâches.
JobSummary	JobSummary	
jobArn	chaîne	ARN de la tâche.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
thingGroupId	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9-]+	ID du groupe d'objets.
targetSelection	chaîne enum : CONTINU | INSTANTANÉ	Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si elle est continue, la tâche peut également être exécutée sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un objet lorsque ce dernier

Nom	Type	Description
		est ajouté à un groupe cible, même après que la tâche a été exécutée par tous les objets initialement dans le groupe.
status	chaîne enum : IN_PROGRESS | ANNULÉ | RÉUSSI	Statut du récapitulatif de la tâche.
createdAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été créée.
lastUpdatedAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été mise à jour.
completedAt	timestamp	Durée écoulée, en secondes, depuis le moment où la tâche a été exécutée.
nextToken	chaîne	Jeton pour l'ensemble de résultats suivant, ou null s'il n'y a pas de résultats supplémentaires.

UpdateJob

UpdateJob command

Met à jour les champs pris en charge de la tâche spécifiée. Les valeurs mises à jour de `timeoutConfig` ne prennent effet que pour les exécutions nouvellement en cours. Les exécutions actuellement en cours continuent à s'exécuter avec l'ancienne configuration du délai d'expiration.

HTTPS (13)

Requête:

```
PATCH /jobs/jobId
{
"description": "string",
"presignedUrlConfig": {
  "expiresInSec": number,
  "roleArn": "string"
},
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
     "baseRatePerMinute": number,
     "incrementFactor": number,
     "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
     },
  "maximumPerMinute": number
  },
"abortConfig": {
  "criteriaList": [
     {
```

```
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
  ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": number
}
}
```

`jobId`

Identifiant de tâche, qui doit être unique pour votreAWS. Nous vous recommandons d'utiliser un UUID. Les caractères alphanumériques, « - » et « _ » peuvent être utilisés ici.

`description`

Facultatif. Brève description de la tâche.

`presignedUrlConfigData`

Facultatif. Informations de configuration pour les URL Amazon S3 présignées.

`roleArn`

ARN du rôle IAM qui contient les autorisations pour accéder au compartiment Amazon S3. Il s'agit du compartiment qui contient les données que les appareils téléchargent avec les URL Amazon S3 présignées. Le rôle doit aussi accorder à AWS IoT l'autorisation d'endosser le rôle. Pour plus d'informations, consultez Création de tâches (p. 599).

`expiresInSec`

Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.

`jobExecutionRolloutConfig`

Facultatif. Permet de créer un déploiement étalé d'une tâche.

`maximumPerMinute`

Nombre maximal d'objets sur lesquels la tâche est envoyée pour l'exécution, par minute. Valeurs valides : de 1 à 1000. Si la valeur n'est pas spécifiée, la valeur par défaut est 1000. Le nombre réel d'objets qui reçoivent la tâche peut être inférieur pendant un intervalle de minutes particulier (en raison de la latence système), mais ne peut pas être supérieur à celui de la valeur spécifiée.

`exponentialRate`

Permet de définir la fréquence exponentielle de déploiement d'une tâche.

`baseRatePerMinute`

Nombre minimal d'objets qui sont informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.

`incrementFactor`

Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.

`rateIncreaseCriteria`

Critères pour initier l'augmentation du taux de déploiement pour une tâche. Définissez les valeurs de `numberOfNotifiedThings` ou `numberOfSucceededThings`, mais pas les deux.

`numberOfNotifiedThings`

Seuil du nombre d'objets notifiés qui déclenche l'augmentation de la fréquence de déploiement.

`numberOfSucceededThings`

Seuil du nombre d'objets réussis qui déclenche l'augmentation de la fréquence de déploiement.

`abortConfig`

Facultatif. Détails des critères d'annulation d'une tâche.

`criteriaList`

Liste de critères d'annulation afin de définir des règles pour annuler la tâche.

`action`

Type d'action d'annulation pour lancer une annulation de tâche.

`failureType`

Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.

`minNumberOfExecutedThings`

Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.

`thresholdPercentage`

Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche.

`timeoutConfig`

Facultatif. Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.

`inProgressTimeoutInMinutes`

Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Un minuteur est démarré ou redémarré, chaque fois que l'état d'exécution de la tâche est spécifié comme `IN_PROGRESS` avec ce champ renseigné. Si le statut d'exécution de la tâche n'est pas défini sur un état terminal avant que le minuteur n'expire, ou qu'une autre mise à jour d'état d'exécution de tâche ne soit envoyée avec ce champ renseigné, le statut est automatiquement défini sur `TIMED_OUT`.

Réponse:

```
HTTP/1.1 200
```

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

CLI (13)

Résumé :

```
aws iot  update-job \
--job-id <value> \
[--description <value>] \
[--presigned-url-config <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"description": "string",
"presignedUrlConfig": {
  "expiresInSec": number,
  "roleArn": "string"
},
"jobExecutionsRolloutConfig": {
  "exponentialRate": {
     "baseRatePerMinute": number,
     "incrementFactor": number,
     "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
     }
  },
  "maximumPerMinute": number
},
"abortConfig": {
  "criteriaList": [
     {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
     }
  ]
},
"timeoutConfig": {
  "inProgressTimeoutInMinutes": number
}
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant de tâche, qui doit être unique pour votre Compte AWS . Nous vous recommandons d'utiliser un UUID. Les caractères alphanumériques, « - » et « _ » peuvent être utilisés ici.
description	chaîne longueur max. : 2 028 modèle : [^\\p{C}]+	Brève description de la tâche.

Nom	Type	Description
presignedUrlConfig	PresignedUrlConfig	Informations de configuration pour les URL S3 présignées.
roleArn	chaîne longueur max. : 2 048 min. : 20	ARN d'un rôle IAM qui octroie l'autorisation de télécharger les fichiers depuis le compartiment S3 où les données de tâche/ les mises à jour sont stockées. Le rôle doit également accorder l'autorisation à AWS IoT de télécharger les fichiers.
expiresInSec	long classe Java : java.lang.Long plage - max. : 3 600 min. : 60	Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Permet de créer un déploiement étalé de la tâche.
maximumPerMinute	entier classe Java : java.lang.Integer plage - max : 1 000 min. : 1	Nombre maximal d'objets qui sont informés de la présence d'une tâche en attente, par minute. Ce paramètre vous permet de créer un déploiement étalé.
exponentialRate	ExponentialRolloutRate	Taux d'augmentation pour le déploiement d'une tâche. Ce paramètre vous permet de définir un taux exponentiel pour le déploiement d'une tâche.
baseRatePerMinute	classe Java : java.lang.Integer	Nombre minimal d'objets qui sont informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.
incrementFactor	classe java : java.lang.Double	Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.

Nom	Type	Description
rateIncreaseCriteria	RateIncreaseCriteria	Permet de définir un critère pour initier l'augmentation de la fréquence de déploiement pour une tâche. Définissez les valeurs de `numberOfNotifiedThings` ou `numberOfSucceededThings`, mais pas les deux.
numberOfNotifiedThings	classe java : java.lang.Double	Seuil du nombre d'objets notifiés qui déclenche l'augmentation de la fréquence de déploiement.
numberOfSucceededThings	classe java : java.lang.Double	Seuil du nombre d'objets réussis qui déclenche l'augmentation de la fréquence de déploiement.
abortConfig	AbortConfig	Vous permet de créer des critères pour annuler une tâche.
criteriaList	AbortCriteria	Liste de critères d'annulation afin de définir des règles pour annuler la tâche.
action	classe java : java.lang.String (CANCEL)	Type d'action d'annulation pour lancer une annulation de tâche.
failureType	classe java : java.lang.String (FAILED | REJECTED | TIMED_OUT | ALL)	Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.
minNumberOfExecutedThings	classe Java : java.lang.Integer)	Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.
thresholdPercentage	classe java : java.lang.Double)	Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche. AWS IoT prend en charge jusqu'à deux chiffres après la virgule (par exemple, 10,9 et 10,99, mais pas 10,999).

Nom	Type	Description
timeoutConfig	TimeoutConfig	Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.
inProgressTimeoutInMinutes	long	Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Un minuteur est démarré ou redémarré, chaque fois que l'état d'exécution de la tâche est spécifié comme `IN_PROGRESS` avec ce champ renseigné. Si le statut d'exécution de la tâche n'est pas défini sur un état terminal avant que le minuteur n'expire, ou qu'une autre mise à jour d'état d'exécution de tâche ne soit envoyée avec ce champ renseigné, le statut est automatiquement défini sur `TIMED_OUT`.
documentParameters	map Clé : ParameterKey Valeur : ParameterValue	Paramètres pour le document de tâche.
ParameterKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
ParameterValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]+	

Sortie :

```
HTTP/1.1 200
```

Si l'action aboutit, le service renvoie une réponse HTTP 200 avec un corps HTTP vide.

Types de données de gestion et de contrôle des tâches

Les types de données suivants sont utilisés par les applications de gestion et de contrôle pour communiquer avec le service AWS IoT Jobs.

Job

Job data type

L'objet `Job` contient des informations sur une tâche.

Syntax (1)

```
{
    "jobArn": "string",
    "jobId": "string",
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED",
    "forceCanceled": boolean,
    "targetSelection": "CONTINUOUS|SNAPSHOT",
    "comment": "string",
    "targets": ["string"],
    "description": "string",
    "createdAt": timestamp,
    "lastUpdatedAt": timestamp,
    "completedAt": timestamp,
    "jobProcessDetails": {
        "processingTargets": ["string"],
        "numberOfCanceledThings": long,
        "numberOfSucceededThings": long,
        "numberOfFailedThings": long,
        "numberOfRejectedThings": long,
        "numberOfQueuedThings": long,
        "numberOfInProgressThings": long,
        "numberOfRemovedThings": long,
        "numberOfTimedOutThings": long
    },
    "presignedUrlConfig": {
        "expiresInSec": number,
        "roleArn": "string"
    },
    "jobExecutionsRolloutConfig": {
        "exponentialRate": {
           "baseRatePerMinute": integer,
           "incrementFactor": integer,
           "rateIncreaseCriteria": {
              "numberOfNotifiedThings": integer, // Set one or the other
              "numberOfSucceededThings": integer // of these two values.
           },
           "maximumPerMinute": integer
      }
    },
    "abortConfig": {
       "criteriaList": [
          {
             "action": "string",
             "failureType": "string",
             "minNumberOfExecutedThings": integer,
             "thresholdPercentage": integer
          }
       ]
    },
```

```
    "timeoutConfig": {
        "inProgressTimeoutInMinutes": long
    }
}
```

Description (1)

`jobArn`

ARN identifiant la tâche selon le format « arn:aws:iot:*region*:*account*:job/*jobId* ».

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`status`

Statut de la tâche : `IN_PROGRESS`, `CANCELED` ou `SUCCEEDED`.

`targetSelection`

Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si la valeur est CONTINUOUS, la tâche peut aussi s'exécuter sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un objet lorsque ce dernier est ajouté à un groupe cible, même après que la tâche a été exécutée par tous les objets initialement dans le groupe.

`comment`

Si la tâche a été mise à jour, décrit la raison de la mise à jour.

`targets`

Liste des objets et des groupes d'objets AWS IoT auxquels la tâche doit être envoyée.

`description`

Brève description de la tâche.

`createdAt`

Durée écoulée, en secondes, depuis le moment où la tâche a été créée.

`lastUpdatedAt`

Durée écoulée, en secondes, depuis le moment où la tâche a été mise à jour.

`completedAt`

Durée écoulée, en secondes, depuis le moment où la tâche a été exécutée.

`jobProcessDetails`

Détails sur le processus de tâche :

`processingTargets`

Liste des objets et groupes d'objets AWS IoT qui exécutent actuellement la tâche.

`numberOfCanceledThings`

Nombre d'objets AWS IoT ayant annulé la tâche.

`numberOfSucceededThings`

Nombre d'objets AWS IoT ayant exécuté la tâche avec succès.

`numberOfFailedThings`

Nombre d'objets AWS IoT ayant échoué à exécuter la tâche.

`numberOfRejectedThings`

Nombre d'objets AWS IoT ayant rejeté la tâche.

`numberOfQueuedThings`

Nombre d'objets AWS IoT en attente d'exécution de la tâche.

`numberOfInProgressThings`

Nombre d'objets AWS IoT en cours d'exécution de la tâche.

`numberOfRemovedThings`

Nombre d'objets AWS IoT qui ne sont plus programmés pour exécuter la tâche, car ils ont été supprimés ou retirés du groupe qui était une cible de la tâche.

`numberOfTimedOutThings`

Le nombre d'objets dont le statut de l'exécution de la tâche est `TIMED_OUT`.

`presignedUrlConfig`

Informations de configuration pour les URL Amazon S3 présignées.

`expiresInSec`

Durée de validité (en secondes) des URL présignées. Les valeurs valides sont comprises entre 60 et 3600. La valeur par défaut est de 3 600 secondes. Les URL présignées sont générées lorsque le service AWS IoTJobs reçoit une demande MQTT pour le document de tâche.

`roleArn`

ARN d'un rôle IAM qui octroie l'autorisation de télécharger des fichiers depuis un compartiment Amazon S3. Le rôle doit également accorder l'autorisation à AWS IoT de télécharger les fichiers. Pour plus d'informations sur la création et la configuration du rôle, consultez Création de tâches (p. 599).

`jobExecutionRolloutConfig`

Facultatif. Permet de créer un déploiement étalé d'une tâche.

`maximumPerMinute`

Nombre maximal d'objets (appareils) auxquels la tâche est envoyée pour exécution, par minute.

`exponentialRate`

Permet de définir la fréquence exponentielle de déploiement d'une tâche.

`baseRatePerMinute`

Nombre minimal d'objets qui sont informés de la présence d'une tâche en attente, par minute au début du déploiement d'une tâche. Ce paramètre vous permet de définir le taux initial de déploiement.

`incrementFactor`

Facteur exponentielle pour augmenter la fréquence de déploiement d'une tâche.

`rateIncreaseCriteria`

Critères pour initier l'augmentation du taux de déploiement pour une tâche. Vous pouvez spécifier `numberOfNotifiedThings` ou `numberOfSucceededThing`, mais pas les deux.

`numberOfNotifiedThings`

Seuil du nombre d'objets notifiés qui déclenche l'augmentation de la fréquence de déploiement.

`numberOfSucceededThings`

Seuil du nombre d'objets réussis qui déclenche l'augmentation de la fréquence de déploiement.

`abortConfig`

Facultatif. Détails des critères d'annulation d'une tâche.

`criteriaList`

Liste de critères d'annulation afin de définir des règles pour annuler la tâche.

`action`

Type d'action d'annulation pour lancer une annulation de tâche.

`failureType`

Type d'échec d'exécution de tâche afin de définir une règle pour lancer une annulation de tâche.

`minNumberOfExecutedThings`

Nombre minimal d'objets exécutés avant d'évaluer une règle d'annulation.

`thresholdPercentage`

Seuil sous forme de pourcentage du nombre total d'objets exécutés qui déclenchent une annulation de tâche.

`timeoutConfig`

Facultatif. Spécifie la durée pendant laquelle chaque appareil doit terminer son exécution de la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état de l'exécution de la tâche n'est pas défini sur un autre état terminal avant l'expiration, il est défini avec la valeur `TIMED_OUT`.

`inProgressTimeoutInMinutes`

Spécifie la durée (en minutes) pendant laquelle cet appareil doit terminer l'exécution de la tâche. Un minuteur est démarré ou redémarré, chaque fois que l'état d'exécution de la tâche est spécifié comme `IN_PROGRESS` avec ce champ renseigné. Si le statut d'exécution de la tâche n'est pas défini sur un état terminal avant que le minuteur n'expire, ou qu'une autre mise à jour d'état d'exécution de tâche ne soit envoyée avec ce champ renseigné, le statut est automatiquement défini sur `TIMED_OUT`.

JobSummary

JobSummary data type

L'objet `JobSummary` contient un résumé de tâche.

Syntax (2)

```
{
    "jobArn": "string",
    "jobId": "string",
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED",
```

```
    "targetSelection": "CONTINUOUS|SNAPSHOT",
    "thingGroupId": "string",
    "createdAt": timestamp,
    "lastUpdatedAt": timestamp,
    "completedAt": timestamp
}
```

Description (2)

`jobArn`

ARN qui identifie la tâche.

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`status`

L'état de la tâche. Peut avoir l'une des valeurs suivantes : `IN_PROGRESS`, `CANCELED` ou `SUCCEEDED`.

`targetSelection`

Indique si la tâche continue à s'exécuter (CONTINUOUS) ou est terminée une fois que tous les objets spécifiés comme cibles l'ont exécutée (SNAPSHOT). Si la valeur est CONTINUOUS, la tâche peut aussi s'exécuter sur un objet lorsqu'une modification est détectée dans une cible. Par exemple, une tâche s'exécute sur un objet lorsque ce dernier est ajouté à un groupe cible, même après que la tâche a été exécutée par tous les objets initialement dans le groupe.

`thingGroupId`

ID du groupe d'objets.

`createdAt`

Horodatage UNIX de la date de création de la tâche.

`lastUpdatedAt`

Horodatage UNIX de la dernière date de mise à jour de la tâche.

`completedAt`

Horodatage UNIX de la date de fin de la tâche.

JobExecution

JobExection data type

L'objet `JobExecution` représente l'exécution d'une tâche sur un appareil.

Syntax (3)

```
{
    "approximateSecondsBeforeTimedOut": 50,
    "executionNumber": 1234567890,
    "forceCanceled": true|false,
    "jobId": "string",
    "lastUpdatedAt": timestamp,
    "queuedAt": timestamp,
    "startedAt": timestamp,
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|
REMOVED",
```

```
    "forceCanceled": boolean,
    "statusDetails": {
        "detailsMap": {
            "string": "string" ...
        },
        "status": "string"
    },
    "thingArn": "string",
    "versionNumber": 123
}
```

Description (3)

approximateSecondsBeforeTimedOut

Estimation du nombre de secondes qui restent avant que le statut d'exécution de la tâche ne devienne `TIMED_OUT`. Le délai peut être n'importe quelle valeur comprise entre 1 minute et 7 jours (1 à 10 080 minutes). L'expiration réelle du délai d'exécution de la tâche peut intervenir jusqu'à 60 secondes plus tard que la durée prévue.

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`executionNumber`

Nombre qui identifie l'exécution de la tâche sur cet appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.

`thingArn`

ARN de l'objet AWS IoT.

`queuedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.

`lastUpdatedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

`startedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.

`status`

Statut de l'exécution de tâche. Peut avoir la valeur `QUEUED`, `IN_PROGRESS`, `FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` ou `REMOVED`.

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.

JobExecutionSummary

JobExecutionSummary data type

L'objet `JobExecutionSummary` contient les informations récapitulatives sur l'exécution de tâche :

Syntax (4)

```
{
```

```
    "executionNumber": 1234567890,
    "queuedAt": timestamp,
    "lastUpdatedAt": timestamp,
    "startedAt": timestamp,
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED"
}
```

Description (4)

`executionNumber`

Nombre qui identifie une exécution de tâche sur un appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.

`queuedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.

`lastUpdatedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

`startAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.

`status`

Statut de l'exécution de tâche : `QUEUED`, `IN_PROGRESS`, `FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED` ou `REMOVED`.

JobExecutionSummaryForJob

JobExecutionSummaryForJob data type

L'objet `JobExecutionSummaryForJob` contient un récapitulatif des informations sur les exécutions de tâche d'une tâche spécifique.

Syntax (5)

```
{
    "executionSummaries": [
        {
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyThing",
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            }
        },
        ...
    ]
}
```

Description (5)

`thingArn`

ARN de l'objet AWS IoT.

`jobExecutionSummary`

Objet JobExecutionSummary (p. 681)

JobExecutionSummaryForThing

JobExecutionSummaryForThing data type

L'objet `JobExecutionSummaryForThing` contient un récapitulatif des informations sur une exécution de tâche sur un objet spécifique.

Syntax (6)

```
{
    "executionSummaries": [
        {
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            },
            "jobId": "MyThingJob"
        },
        ...
    ]
}
```

Description (6)

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`jobExecutionSummary`

Objet JobExecutionSummary (p. 681)

API MQTT et HTTPS pour les appareils Jobs

Les commandes suivantes sont disponibles sur les protocoles HTTPS et MQTT.

- GetPendingJobExecutions (p. 684)
- StartNextPendingJobExecution (p. 687)
- DescribeJobExecution (p. 693)
- UpdateJobExecution (p. 698)
- JobExecutionsChanged (p. 704)
- NextJobExecutionChanged (p. 705)

> **Note**
>
> Les commandes de périphérique des tâches peuvent être émises en publiant des messages MQTT dans leRubriques réservées utilisées pour les commandes Tâches (p. 98). Votre client est automatiquement abonné aux rubriques de message de réponse de ces commandes, ce qui signifie que le courtier de messages publiera des rubriques de messages de réponse sur le client qui a publié le message de commande, que votre client ait souscrit ou non aux rubriques de message de réponse.

Étant donné que le courtier de messages publie des messages de réponse, même sans s'y abonner explicitement, votre client doit être configuré pour recevoir et identifier les messages qu'il reçoit. Votre client doit également confirmer que le*thingName*dans la rubrique du message entrant s'applique au nom du truc du client avant que le client n'agisse sur le message. Messages queAWS IoTenvoie en réponse aux messages de commande de l'API MQTT Jobs sont facturés à votre compte, que vous y ayez souscrit explicitement ou non.

GetPendingJobExecutions

Obtient la liste de toutes les tâches d'un objet qui ne se trouvent pas dans un état terminal.

Note

AWS IoTpublie les messages de réponse directement au client qui a fait la demande, que le client ait souscrit ou non aux rubriques de réponse. Les clients devraient s'attendre à recevoir les messages de réponse même s'ils ne s'y sont pas abonnés. Ces messages de réponse ne passent pas par le courtier de messages et ils ne peuvent pas être abonnés par d'autres clients ou règles. Messages de progression de Job qui sont traités par l'intermédiaire du courtier de messages et peuvent être utilisés parAWS IoTsont publiées en tant queÉvénements Jobs (p. 1055).

MQTT (12)

Pour appeler cette API, publiez un message sur `$aws/things/`*`thingName`*`/jobs/get`.

Charge utile de la demande :

```
{ "clientToken": "string" }
```

`clientToken`

Facultatif. Jeton client utilisé pour établir une corrélation entre les demandes et les réponses. Saisissez une valeur arbitraire ici et elle sera reflétée dans la réponse.

Le courtier de messages publiera`$aws/things/`*`thingName`*`/jobs/get/accepted`and`$aws/things/`*`thingName`*`/jobs/get/rejected`même sans abonnement spécifique à eux. Cependant, pour que votre client reçoive les messages, il doit les écouter. Pour de plus amples informations, veuillez consulterla note sur les messages de l'API Jobs (p. 683).

Charge utile de la réponse :

```
{
"inProgressJobs" : [ JobExecutionSummary ... ],
"queuedJobs" : [ JobExecutionSummary ... ],
"timestamp" : 1489096425069,
"clientToken" : "client-001"
}
```

`inProgressJobs`

Liste des objets JobExecutionSummary (p. 707) avec le statut `IN_PROGRESS`.

`queuedJobs`

Liste des objets JobExecutionSummary (p. 707) avec le statut `QUEUED`.

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses.

`timestamp`

Durée écoulée, en secondes depuis la date epoch Unix où le message a été envoyé.

HTTPS (12)

Requête:

```
GET /things/thingName/jobs
```

`thingName`

Nom de l'objet associé à l'appareil.

Réponse:

```
{
"inProgressJobs" : [ JobExecutionSummary ... ],
"queuedJobs" : [ JobExecutionSummary ... ]
}
```

`inProgressJobs`

Une liste d'objets JobExecutionSummary (p. 707).

`queuedJobs`

Une liste d'objets JobExecutionSummary (p. 707).

CLI (12)

Résumé :

```
aws iot-jobs-data  get-pending-job-executions \
--thing-name <value>  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"thingName": "string"
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet qui exécute la tâche.

Sortie :

```
{
"inProgressJobs": [
{
  "jobId": "string",
  "queuedAt": long,
  "startedAt": long,
  "lastUpdatedAt": long,
  "versionNumber": long,
  "executionNumber": long
}
],
"queuedJobs": [
{
  "jobId": "string",
  "queuedAt": long,
  "startedAt": long,
  "lastUpdatedAt": long,
  "versionNumber": long,
  "executionNumber": long
}
]
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
inProgressJobs	liste membre : JobExecutionSummary classe Java : java.util.List	Liste d'objets JobExecutionSummary avec le statut IN_PROGRESS.
JobExecutionSummary	JobExecutionSummary	
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
queuedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	long classe Java : java.lang.Long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois que le service AWS IoT Jobs reçoit une mise à jour d'un appareil.

Nom	Type	Description
executionNumber	long classe Java : java.lang.Long	Nombre qui identifie une exécution de tâche sur un appareil.
queuedJobs	liste membre : JobExecutionSummary classe Java : java.util.List	Liste d'objets JobExecutionSummary avec le statut QUEUED.
JobExecutionSummary	JobExecutionSummary	
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
queuedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	long classe Java : java.lang.Long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois que le service AWS IoT Jobs reçoit une mise à jour d'un appareil.
executionNumber	long classe Java : java.lang.Long	Nombre qui identifie une exécution de tâche sur un appareil.

StartNextPendingJobExecution

Obtient et démarre l'exécution de tâche en attente suivante (statut IN_PROGRESS ou QUEUED) pour un objet.

- Toutes les exécutions de tâche avec le statut IN_PROGRESS sont renvoyées en premier.
- Les exécutions de tâche sont renvoyées dans l'ordre selon lequel elles ont été créées.
- Si la prochaine exécution de tâche en attente est QUEUED, son état est modifié en IN_PROGRESS et les détails du statut de l'exécution de la tâche sont définis comme indiqué.
- Si la prochaine exécution de tâche en attente est déjà IN_PROGRESS, les informations détaillées de son statut ne sont pas modifiées.
- Si aucune exécution de tâche n'est en attente, la réponse n'inclut pas le champ `execution`.

- Le cas échéant, vous pouvez créer un minuteur d'étape en définissant une valeur pour la propriété `stepTimeoutInMinutes`. Si vous ne mettez pas à jour la valeur de cette propriété en exécutant `UpdateJobExecution`, l'exécution de la tâche expire lorsque le minuteur d'étape expire.

  Note

  AWS IoTpublie les messages de réponse directement au client qui a fait la demande, que le client ait souscrit ou non aux rubriques de réponse. Les clients devraient s'attendre à recevoir les messages de réponse même s'ils ne s'y sont pas abonnés. Ces messages de réponse ne passent pas par le courtier de messages et ils ne peuvent pas être abonnés par d'autres clients ou règles. Messages de progression de Job qui sont traités par l'intermédiaire du courtier de messages et peuvent être utilisés parAWS IoTsont publiées en tant queÉvénements Jobs (p. 1055).

MQTT (13)

Pour appeler cette API, publiez un message sur `$aws/things/thingName/jobs/start-next`.

Charge utile de la demande :

```
{
"statusDetails": {
    "string": "job-execution-state"
    ...
},
"stepTimeoutInMinutes": long,
"clientToken": "string"
}
```

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations `statusDetails` demeurent inchangées.

`stepTimeOutInMinutes`

Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans le champ `stepTimeoutInMinutes`), l'état de l'exécution de la tâche est automatiquement défini avec la valeur `TIMED_OUT`. La définition du délai d'expiration n'a aucun effet sur le délai d'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (`CreateJob` à l'aide du champ `timeoutConfig`).

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses. Saisissez une valeur arbitraire ici et elle sera reflétée dans la réponse.

Le courtier de messages publiera`$aws/things/thingName/jobs/start-next/accepted`and`$aws/things/thingName/jobs/start-next/rejected`même sans abonnement spécifique à eux. Cependant, pour que votre client reçoive les messages, il doit les écouter. Pour de plus amples informations, veuillez consulterla note sur les messages de l'API Jobs (p. 683).

Charge utile de la réponse :

```
{
"execution" : JobExecutionData,
```

```
"timestamp" : timestamp,
"clientToken" : "string"
}
```

`execution`

Objet JobExecution (p. 705) Exemples :

```
{
"execution" : {
    "jobId" : "022",
    "thingName" : "MyThing",
    "jobDocument" : "< contents of job document >",
    "status" : "IN_PROGRESS",
    "queuedAt" : 1489096123309,
    "lastUpdatedAt" : 1489096123309,
    "versionNumber" : 1,
    "executionNumber" : 1234567890
},
"clientToken" : "client-1",
"timestamp" : 1489088524284,
}
```

`timestamp`

Durée écoulée, en millisecondes depuis la date epoch Unix où le message a été envoyé à l'appareil.

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses.

HTTPS (13)

Requête:

```
PUT /things/thingName/jobs/$next
{
"statusDetails": {
    "string": "string"
    ...
},
"stepTimeoutInMinutes": long
}
```

`thingName`

Nom de l'objet associé à l'appareil.

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations `statusDetails` demeurent inchangées.

`stepTimeOutInMinutes`

Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans le champ `stepTimeoutInMinutes`), l'état de l'exécution de la tâche est automatiquement défini avec la

valeur `TIMED_OUT`. La définition du délai d'expiration n'a aucun effet sur le délai d'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (`CreateJob` à l'aide du champ `timeoutConfig`).

Réponse:

```
{
"execution" :  JobExecution
}
```

`execution`

Objet JobExecution (p. 705) Exemples :

```
{
"execution" : {
    "jobId" : "022",
    "thingName" : "MyThing",
    "jobDocument" : "< contents of job document >",
    "status" : "IN_PROGRESS",
    "queuedAt" : 1489096123309,
    "lastUpdatedAt" : 1489096123309,
    "versionNumber" : 1,
    "executionNumber" : 1234567890
},
"clientToken" : "client-1",
"timestamp" : 1489088524284,
}
```

CLI (13)

Résumé :

```
aws iot-jobs-data  start-next-pending-job-execution \
--thing-name <value> \
{--step-timeout-in-minutes <value>] \
[--status-details <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"thingName": "string",
"statusDetails": {
"string": "string"
},
"stepTimeoutInMinutes": long
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
thingName	chaîne longueur max. : 128 min. : 1	Nom de l'objet associé à l'appareil.

Nom	Type	Description
	modèle : [a-zA-Z0-9:_-]+	
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations statusDetails demeurent inchangées.
stepTimeOutInMinutes	long	Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans le champ `stepTimeoutInMinutes`), l'état de l'exécution de la tâche est automatiquement défini avec la valeur `TIMED_OUT`. La définition du délai d'expiration n'a aucun effet sur le délai d'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (`CreateJob` à l'aide du champ `timeoutConfig`).
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	

Sortie :

```
{
"execution": {
"jobId": "string",
"thingName": "string",
"status": "string",
"statusDetails": {
  "string": "string"
},
```

```
"queuedAt": long,
"startedAt": long,
"lastUpdatedAt": long,
"versionNumber": long,
"executionNumber": long,
"jobDocument": "string"
}
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
execution	JobExecution	Objet JobExecution.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet qui exécute la tâche.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	Statut de l'exécution de tâche. Peut avoir l'une des valeurs suivantes : QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED_OUT, REJECTED ou REMOVED.
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	
queuedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	long classe Java : java.lang.Long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.

Nom	Type	Description
lastUpdatedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.
executionNumber	long classe Java : java.lang.Long	Nombre qui identifie une exécution de tâche sur un appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.
jobDocument	chaîne longueur max. : 32 768	Contenu du document de tâche.

DescribeJobExecution

Permet d'obtenir des informations détaillées sur une exécution de tâche.

Vous pouvez définir le `jobId` sur `$next` pour revenir à la prochaine exécution de tâche en attente (statut IN_PROGRESS ou QUEUED) pour un objet.

MQTT (14)

Pour appeler cette API, publiez un message sur `$aws/things/`*`thingName`*`/jobs/`*`jobId`*`/get`.

Charge utile de la demande :

```
{
"executionNumber": long,
"includeJobDocument": boolean,
"clientToken": "string"
}
```

`thingName`

Nom de l'objet associé à l'appareil.

`jobId`

Identifiant unique attribué à cette tâche lors de sa création.

Ou utilisez `$next` pour revenir à la prochaine exécution de tâche en attente (statut IN_PROGRESS ou QUEUED) pour un objet. Dans ce cas, toutes les exécutions de tâche avec le statut IN_PROGRESS sont renvoyées en premier. Les exécutions de tâche sont renvoyées dans l'ordre selon lequel elles ont été créées.

`executionNumber`

Facultatif. Nombre qui identifie une exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est renvoyée.

`includeJobDocument`

Facultatif. La réponse contient le document de tâche, sauf si la valeur est `false`. La valeur par défaut est `true`.

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses. Saisissez une valeur arbitraire ici et elle sera reflétée dans la réponse.

Le courtier de messages publiera`$aws/things/thingName/jobs/jobId/get/accepted`and`$aws/things/thingName/jobs/jobId/get/rejected`même sans abonnement spécifique à eux. Cependant, pour que votre client reçoive les messages, il doit les écouter. Pour de plus amples informations, veuillez consulterla note sur les messages de l'API Jobs (p. 683).

Charge utile de la réponse :

```
{
"execution" : JobExecutionData,
"timestamp": "timestamp",
"clientToken": "string"
}
```

`execution`

Objet JobExecution (p. 705)

`timestamp`

Durée écoulée, en secondes depuis la date epoch Unix où le message a été envoyé.

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses.

HTTPS (14)

Le statut de l'exécution de tâche doit être `QUEUED` ou `IN_PROGRESS`.

Requête:

```
GET /things/thingName/jobs/jobId?
executionNumber=executionNumber&includeJobDocument=includeJobDocument
```

`thingName`

Nom de l'objet associé à l'appareil.

`jobId`

Identifiant unique attribué à cette tâche lors de sa création.

Ou utilisez `$next` pour revenir à la prochaine exécution de tâche en attente (statut IN_PROGRESS ou QUEUED) pour un objet. Dans ce cas, toutes les exécutions de tâche avec le statut IN_PROGRESS sont renvoyées en premier. Les exécutions de tâche sont renvoyées dans l'ordre selon lequel elles ont été créées.

`includeJobDocument`

Facultatif. La réponse contient le document de tâche, sauf si la valeur est `false`. La valeur par défaut est `true`.

`executionNumber`

Facultatif. Nombre qui identifie une exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est renvoyée.

Réponse:

```
{
"execution" : JobExecution,
}
```

`execution`

Objet JobExecution (p. 705)

CLI (14)

Le statut de l'exécution de tâche doit être `QUEUED` ou `IN_PROGRESS`.

Résumé :

```
aws iot-jobs-data  describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"thingName": "string",
"includeJobDocument": boolean,
"executionNumber": long
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne modèle : [a-zA-Z0-9_-]+|^$next	L'identifiant unique attribué à cette tâche lors de sa création, ou `$next` pour revenir à la prochaine exécution de tâche en attente (statut IN_PROGRESS ou QUEUED) pour un objet. Dans ce cas, toutes les exécutions de tâche avec le statut IN_PROGRESS sont renvoyées en premier. Les exécutions de tâche sont renvoyées dans l'ordre selon lequel elles ont été créées.

Nom	Type	Description
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet associé à l'appareil sur lequel l'exécution de tâche est en cours.
includeJobDocument	boolean classe Java : java.lang.Boolean	Facultatif. La réponse contient le document de tâche, sauf si la valeur est false. Par défaut, la valeur est true.
executionNumber	long classe Java : java.lang.Long	Facultatif. Nombre qui identifie une exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est renvoyée.

Sortie :

```
{
"execution": {
"jobId": "string",
"thingName": "string",
"status": "string",
"statusDetails": {
  "string": "string"
},
"queuedAt": long,
"startedAt": long,
"lastUpdatedAt": long,
"versionNumber": long,
"executionNumber": long,
"jobDocument": "string"
}
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
execution	JobExecution	Contient des données sur une exécution de tâche.
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique que vous avez attribué à cette tâche lors de sa création.
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet qui exécute la tâche.
status	chaîne	Statut de l'exécution de tâche. Peut avoir l'une des valeurs suivantes :

Nom	Type	Description
	enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED_OUT, REJECTED ou REMOVED.
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	
queuedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.
startedAt	long classe Java : java.lang.Long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.
lastUpdatedAt	long	Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.
executionNumber	long classe Java : java.lang.Long	Nombre qui identifie une exécution de tâche sur un appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.
jobDocument	chaîne longueur max. : 32 768	Contenu du document de tâche.

UpdateJobExecution

Met à jour le statut d'une exécution de tâche. Le cas échéant, vous pouvez créer un minuteur d'étape en définissant une valeur pour la propriété `stepTimeoutInMinutes`. Si vous ne mettez pas à jour la valeur de cette propriété en exécutant à nouveau `UpdateJobExecution`, l'exécution de la tâche expire lorsque le minuteur d'étape expire.

> **Note**
>
> AWS IoTpublie les messages de réponse directement au client qui a fait la demande, que le client ait souscrit ou non aux rubriques de réponse. Les clients devraient s'attendre à recevoir les messages de réponse même s'ils ne s'y sont pas abonnés. Ces messages de réponse ne passent pas par le courtier de messages et ils ne peuvent pas être abonnés par d'autres clients ou règles. Messages de progression de Job qui sont traités par l'intermédiaire du courtier de messages et peuvent être utilisés parAWS IoTsont publiées en tant queÉvénements Jobs (p. 1055).

MQTT (15)

Pour appeler cette API, publiez un message sur `$aws/things/`*`thingName`*`/jobs/`*`jobId`*`/update`.

Charge utile de la demande :

```
{
"status": "job-execution-state",
"statusDetails": {
    "string": "string"
    ...
},
"expectedVersion": "number",
"executionNumber": long,
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"stepTimeoutInMinutes": long,
"clientToken": "string"
}
```

`status`

Nouveau statut de l'exécution de tâche (IN_PROGRESS, FAILED, SUCCEEDED ou REJECTED). Il doit être spécifié à chaque mise à jour.

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations `statusDetails` demeurent inchangées.

`expectedVersion`

Version actuelle attendue de l'exécution de tâche. Sa version est incrémentée à chaque mise à jour de l'exécution de tâche. Si la version de l'exécution de tâche stockée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une erreur `VersionMismatch` et un élément ErrorResponse (p. 708) contenant le statut de l'exécution de tâche en cours est renvoyé. (Il est donc inutile d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche.)

`executionNumber`

Facultatif. Nombre qui identifie une exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est utilisée.

`includeJobExecutionState`

Facultatif. Quand le champ est inclus et a la valeur `true`, la réponse contient le champ `JobExecutionState`. La valeur par défaut est `false`.

`includeJobDocument`

Facultatif. Quand le champ est inclus et a la valeur `true`, la réponse contient le champ `JobDocument`. La valeur par défaut est `false`.

`stepTimeoutInMinutes`

Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans ce champ), l'état de l'exécution de la tâche est automatiquement défini avec la valeur `TIMED_OUT`. La définition ou la réinitialisation du délai d'expiration n'a aucun effet sur le délai d'expiration de l'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (en utilisant `CreateJob` à l'aide du champ `timeoutConfig`).

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses. Saisissez une valeur arbitraire ici et elle sera reflétée dans la réponse.

Le courtier de messages publiera`$aws/things/`*`thingName`*`/jobs/`*`jobId`*`/update/accepted`and`$aws/things/`*`thingName`*`/jobs/`*`jobId`*`/update/rejected`même sans abonnement spécifique à eux. Cependant, pour que votre client reçoive les messages, il doit les écouter. Pour de plus amples informations, veuillez consulterla note sur les messages de l'API Jobs (p. 683).

Charge utile de la réponse :

```
{
"executionState": JobExecutionState,
"jobDocument": "string",
"timestamp": timestamp,
"clientToken": "string"
}
```

`executionState`

Objet JobExecutionState (p. 706)

`jobDocument`

Objet de document de tâche (p. 595).

`timestamp`

Durée écoulée, en secondes depuis la date epoch Unix où le message a été envoyé.

`clientToken`

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses.

HTTPS (15)

Requête:

```
POST /things/thingName/jobs/jobId
{
"status": "job-execution-state",
"statusDetails": {
    "string": "string"
    ...
},
```

```
"expectedVersion": "number",
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"stepTimeoutInMinutes": long,
"executionNumber": long
}
```

`thingName`

Nom de l'objet associé à l'appareil.

`jobId`

Identifiant unique attribué à cette tâche lors de sa création.

`status`

Nouveau statut de l'exécution de tâche (IN_PROGRESS, FAILED, SUCCEEDED ou REJECTED). Il doit être spécifié à chaque mise à jour.

`statusDetails`

Facultatif. Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations `statusDetails` demeurent inchangées.

`expectedVersion`

Facultatif. Version actuelle attendue de l'exécution de tâche. Sa version est incrémentée à chaque mise à jour de l'exécution de tâche. Si la version de l'exécution de tâche stockée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une erreur `VersionMismatch` et un élément ErrorResponse (p. 708) contenant le statut de l'exécution de tâche en cours est renvoyé. (Il est donc inutile d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche.)

`includeJobExecutionState`

Facultatif. Quand le champ est inclus et a la valeur `true`, la réponse contient les données JobExecutionState. La valeur par défaut est `false`.

`includeJobDocument`

Facultatif. Lorsque la valeur est `true`, la réponse contient le document de tâche. La valeur par défaut est `false`.

`stepTimeoutInMinutes`

Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans ce champ), l'état de l'exécution de la tâche est automatiquement défini avec la valeur `TIMED_OUT`. La définition ou la réinitialisation du délai d'expiration n'a aucun effet sur le délai d'expiration de l'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (en utilisant `CreateJob` à l'aide du champ `timeoutConfig`).

`executionNumber`

Facultatif. Nombre qui identifie une exécution de tâche sur un appareil.

Réponse:

```
{
"executionState": JobExecutionState,
"jobDocument": "string"
}
```

`executionState`

Objet JobExecutionState (p. 706)

`jobDocument`

Contenu du document de tâche (p. 595).

CLI (15)

Résumé :

```
aws iot-jobs-data  update-job-execution \
--job-id <value> \
--thing-name <value> \
--status <value> \
[--status-details <value>] \
[--expected-version <value>] \
[--include-job-execution-state | --no-include-job-execution-state] \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>]  \
[--cli-input-json <value>] \
[--step-timeout-in-minutes <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json` :

```
{
"jobId": "string",
"thingName": "string",
"status": "string",
"statusDetails": {
"string": "string"
},
"stepTimeoutInMinutes": number,
"expectedVersion": long,
"includeJobExecutionState": boolean,
"includeJobDocument": boolean,
"executionNumber": long
}
```

Champs **`cli-input-json`** :

Nom	Type	Description
jobId	chaîne longueur max. : 64 min. : 1 modèle : [a-zA-Z0-9_-]+	Identifiant unique attribué à cette tâche lors de sa création.
thingName	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	Nom de l'objet associé à l'appareil.
status	chaîne enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT	Nouveau statut de l'exécution de tâche (IN_PROGRESS, FAILED, SUCCEEDED ou

Nom	Type	Description
	| REJETÉ | SUPPRIMÉ | ANNULÉ	REJECTED). Il doit être spécifié à chaque mise à jour.
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Facultatif. Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche. Si aucune valeur n'est spécifiée, les informations statusDetails demeurent inchangées.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	
stepTimeoutInMinutes long Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si le statut d'exécution de tâche n'est pas défini sur un état de mise hors service avant que ce minuteur n'expire, ou avant que le minuteur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur dans ce champ), l'état de l'exécution de la tâche est automatiquement défini avec la valeur `TIMED_OUT`. La définition ou la réinitialisation du délai d'expiration n'a aucun effet sur le délai d'expiration de l'exécution de la tâche qui peut avoir été spécifié lorsque la tâche a été créée (en utilisant `CreateJob` à l'aide du champ `timeoutConfig`).		

Nom	Type	Description
expectedVersion	long classe Java : java.lang.Long	Facultatif. Version actuelle attendue de l'exécution de tâche. Sa version est incrémentée à chaque mise à jour de l'exécution de tâche. Si la version de l'exécution de tâche stockée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une erreur VersionMismatch et une réponse d'erreur contenant le statut de l'exécution de tâche en cours est renvoyée. (Il est donc inutile d'effectuer une demande DescribeJobExecution distincte pour obtenir les données du statut d'exécution de tâche.)
includeJobExecutionState	boolean classe Java : java.lang.Boolean	Facultatif. Quand le champ est inclus et a la valeur true, la réponse contient les données JobExecutionState. La valeur par défaut est false.
includeJobDocument	boolean classe Java : java.lang.Boolean	Facultatif. Lorsque la valeur est true, la réponse contient le document de tâche. La valeur par défaut est false.
executionNumber	long classe Java : java.lang.Long	Facultatif. Nombre qui identifie une exécution de tâche sur un appareil.

Sortie :

```
{
"executionState": {
"status": "string",
"statusDetails": {
  "string": "string"
},
"versionNumber": long
},
"jobDocument": "string"
}
```

Champs de sortie de l'interface de ligne de commande :

Nom	Type	Description
executionState	JobExecutionState	Objet JobExecutionState.
status	chaîne	Statut de l'exécution de tâche. Peut avoir l'une

Nom	Type	Description
	enum : FILE D'ATTENTE | IN_PROGRESS | RÉUSSI | ÉCHEC | TIMED_OUT | REJETÉ | SUPPRIMÉ | ANNULÉ	des valeurs suivantes : QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED_OUT, REJECTED ou REMOVED.
statusDetails	map Clé : DetailsKey Valeur : DetailsValue	Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.
DetailsKey	chaîne longueur max. : 128 min. : 1 modèle : [a-zA-Z0-9:_-]+	
DetailsValue	chaîne longueur max. : 1 024 min. : 1 modèle : [^\\p{C}]*+	
versionNumber	long	Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.
jobDocument	chaîne longueur max. : 32 768	Contenu des documents de tâche.

JobExecutionsChanged

Envoyé chaque fois qu'une exécution de tâche est ajoutée à la liste des exécutions de tâche en attente pour un objet, ou en est supprimée.

MQTT (16)

Sujet: `$aws/things/thingName/jobs/notify`

Charge utile du message :

```
{
"jobs" : {
    "JobExecutionState": [ JobExecutionSummary (p. 707) ... ]
},
"timestamp": timestamp,
}
```

HTTPS (16)

Indisponible.

CLI (16)

Indisponible.

NextJobExecutionChanged

Envoyé chaque fois qu'il y a une modification apportée à l'exécution de tâche définie comme exécution suivante dans la liste des exécutions de tâche en attente pour un objet, comme défini pour DescribeJobExecution (p. 693) avec le jobId `$next`. Le message n'est pas envoyé en cas de modification des détails de l'exécution de tâche suivante, mais uniquement lorsque la prochaine tâche qui sera renvoyée par `DescribeJobExecution` avec le jobId `$next` a changé. Considérons les exécutions de tâche J1 et J2 avec l'état QUEUED. J1 est l'exécution suivante sur la liste des exécutions de tâche en attente. Si l'état de J2 devient IN_PROGRESS tandis que l'état de J1 reste inchangé, cette notification est envoyée et contient les détails de J2.

MQTT (17)

Sujet: `$aws/things/thingName/jobs/notify-next`

Charge utile du message :

```
{
"execution" : JobExecution (p. 705),
"timestamp": timestamp,
}
```

HTTPS (17)

Indisponible.

CLI (17)

Indisponible.

Types de données MQTT et HTTPS pour les appareils TPS

Les types de données suivants sont utilisés pour communiquer avec le service AWS IoTJobs via les protocoles MQTT et HTTPS.

JobExecution

JobExecution data type

Contient des données sur une exécution de tâche.

Syntax (7)

```
{
    "jobId" : "string",
    "thingName" : "string",
    "jobDocument" : "string",
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|
REMOVED",
    "statusDetails": {
        "string": "string"
    },
    "queuedAt" : "timestamp",
    "startedAt" : "timestamp",
    "lastUpdatedAt" : "timestamp",
    "versionNumber" : "number",
```

```
    "executionNumber": long
}
```

Description (7)

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`thingName`

Nom de l'objet qui exécute la tâche.

`jobDocument`

Contenu du document de tâche.

`status`

Statut de l'exécution de tâche. Peut avoir l'une des valeurs suivantes : QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED_OUT, REJECTED ou REMOVED.

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.

`queuedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.

`startedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.

`lastUpdatedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

`versionNumber`

Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.

`executionNumber`

Nombre qui identifie une exécution de tâche sur un appareil. Elle peut être utilisée ultérieurement dans les commandes qui renvoient ou mettent à jour les informations d'exécution de tâche.

JobExecutionState

JobExecutionState data type

Contient les données sur l'état d'une exécution de tâche.

Syntax (8)

```
{
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|
REMOVED",
    "statusDetails": {
        "string": "string"
        ...
    }
```

```
    "versionNumber": "number"
}
```

Description (8)

`status`

Statut de l'exécution de tâche. Peut avoir l'une des valeurs suivantes : QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED_OUT, REJECTED ou REMOVED.

`statusDetails`

Ensemble de paires nom-valeur décrivant le statut de l'exécution de la tâche.

`versionNumber`

Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois qu'elles sont mises à jour par un appareil.

JobExecutionSummary

JobExecutionSummary data type

Contient un sous-ensemble d'informations sur une exécution de tâche.

Syntax (9)

```
{
    "jobId": "string",
    "queuedAt": timestamp,
    "startedAt": timestamp,
    "lastUpdatedAt": timestamp,
    "versionNumber": "number",
    "executionNumber": long
}
```

Description (9)

`jobId`

Identifiant unique que vous avez attribué à cette tâche lors de sa création.

`queuedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été placée en file d'attente.

`startedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a démarré.

`lastUpdatedAt`

Durée écoulée, en secondes, depuis le moment où l'exécution de tâche a été mise à jour.

`versionNumber`

Version de l'exécution de tâche. Les versions d'exécution de tâche sont incrémentées chaque fois que le service AWS IoT Jobs reçoit une mise à jour d'un appareil.

`executionNumber`

Nombre qui identifie une exécution de tâche sur un appareil.

ErrorResponse

ErrorResponse data type

Contient les informations sur une erreur qui s'est produite au cours d'une opération du service AWS IoT Jobs.

Syntax (10)

```
{
    "code": "ErrorCode",
    "message": "string",
    "clientToken": "string",
    "timestamp": timestamp,
    "executionState": JobExecutionState
}
```

Description (10)

`code`

ErrorCode peut être défini sur :

InvalidTopic

La demande a été envoyée à une rubrique de l'espace de noms AWS IoT Jobs qui n'est pas mappée à une API.

InvalidJson

Le contenu de la demande n'a pas pu être interprété comme JSON codé en UTF-8 valide.

InvalidRequest

Le contenu de la demande n'était pas valide. Par exemple, ce code est renvoyé lorsqu'une demande `UpdateJobExecution` contient des détails d'état non valides. Le message contient des détails sur l'erreur.

InvalidStateTransition

Une mise à jour a tenté de faire passer l'exécution de tâche dans un état qui n'est pas valide en raison de l'état actuel de l'exécution de tâche (par exemple, une tentative de modification d'une demande de l'état SUCCEEDED à l'état IN_PROGRESS). Dans ce cas, le corps du message d'erreur contient aussi le champ `executionState`.

ResourceNotFound

La valeur `JobExecution` spécifiée par la rubrique de la demande n'existe pas.

VersionMismatch

La version attendue spécifiée dans la demande ne correspond pas à la version de l'exécution de la tâche dans le service AWS IoT Jobs. Dans ce cas, le corps du message d'erreur contient aussi le champ `executionState`.

InternalError

Une erreur interne s'est produite pendant le traitement de la demande.

RequestThrottled

La demande a été limitée.

TerminalStateReached

Se produit quand une commande pour décrire une tâche est exécutée sur une tâche qui se trouve dans un état terminal.

`message`

Chaîne de message d'erreur.

`clientToken`

Chaîne arbitraire utilisée pour mettre en corrélation une demande et sa réponse.

`timestamp`

Nombre de secondes depuis la date epoch Unix.

`executionState`

Objet JobExecutionState (p. 706) Ce champ est inclus uniquement lorsque le champ `code` a la valeur `InvalidStateTransition` ou `VersionMismatch`. Il est donc inutile dans ces cas-là d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche en cours.

Configuration du déploiement et de l'interruption des tâches

Les tâches AWS IoT peuvent être déployées à l'aide de fréquences de déploiement variables lorsque divers critères et seuils sont satisfaits. Les déploiements de tâches peuvent également être abandonnés si le nombre de tâches en échec correspond à un ensemble de critères. Ces configurations de déploiement vous permettent d'effectuer un meilleur contrôle granulaire sur le rayon d'impact d'une tâche. Les critères de fréquence de déploiement des tâches sont définis lors de la création de la tâche via l'objet `JobExecutionsRolloutConfig`. Les critères d'annulation de tâche sont définis lors de la création de la tâche via l'objet `AbortConfig`.

Utilisation des fréquences de déploiement des tâches

Vous définissez une fréquence de déploiement des tâches en configurant la propriété `ExponentialRolloutRate` de l'objet `JobExecutionsRolloutConfig` lorsque vous exécutez l'API `CreateJob`. L'exemple suivant définit une fréquence de déploiement variable en utilisant le paramètre `exponentialRate`.

```
{
...
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": 50,
      "incrementFactor": 2,
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": 1000,
        "numberOfSucceededThings": 1000
      },
      "maximumPerMinute": 1000
    }
  }
...
}
```

Le paramètre `baseRatePerMinute` spécifie la fréquence à laquelle les tâches sont exécutées `numberOfNotifiedThings` jusqu'à ce que le seuil `numberOfSucceededThings` ait été atteint.

Le paramètre `incrementFactor` spécifie le facteur exponentiel par lequel la fréquence de déploiement augmente après que le seuil `numberOfNotifiedThings` ou `numberOfSucceededThings` a été atteint.

Le paramètre `rateIncreaseCriteria` est un objet qui spécifie le seuil `numberOfNotifiedThings` ou `numberOfSucceededThings`.

Le paramètre `maximumPerMinute` spécifie la limite supérieure de la fréquence à laquelle les exécutions de tâche peuvent se produire. Les valeurs valides s'étendent de 1 à 1 000. Ce paramètre est obligatoire lorsque vous transmettez un objet `ExponentialRate`. Dans une fréquence de déploiement variable, cette valeur établit la limite supérieure de la fréquence de déploiement d'une tâche.

Un déploiement de tâche avec la configuration ci-dessus démarre à la fréquence de 50 exécutions de tâche par minute. Il continue à ce rythme jusqu'à ce que 1 000 éléments aient reçu les notifications d'exécution de tâche (si une valeur pour `numberOfNotifiedThings` a été spécifiée) ou que 1 000 exécutions de tâche réussies se soient produites (si une valeur pour `numberOfSucceededThings` a été spécifiée).

Le tableau suivant illustre la façon dont le déploiement procéderait sur les quatre premiers incréments.

Fréquence de lancement par minute	50	100	200	400
Nombre d'exécutions réussies ou de périphériques notifiés	1000	2000	3000	4000

La configuration suivante définit une fréquence de déploiement statique.

```
{
...
  "jobExecutionsRolloutConfig": {
         "maximumPerMinute": 1000
...
}
```

Le paramètre `maximumPerMinute` spécifie la limite supérieure de la fréquence à laquelle les exécutions de tâche peuvent se produire. Les valeurs valides s'étendent de 1 à 1 000. Ce paramètre est facultatif. Si vous ne spécifiez aucune valeur, la valeur par défaut 1000 est utilisée.

Utilisation des configurations du déploiement et de l'interruption des tâches

Vous configurez une condition d'annulation de tâche en configurant l'objet `AbortConfig` facultatif lorsque vous exécutez l'API `CreateJob`. Cette section décrit l'effet que l'exemple de configuration suivant aurait sur un déploiement de tâche confronté à plusieurs exécutions ayant échoué.

```
  "abortConfig": {
     "criteriaList": [
        {
           "action": "CANCEL",
           "failureType": "FAILED",
           "minNumberOfExecutedThings": 100,
           "thresholdPercentage": 20
```

```
        },
        {
           "action": "CANCEL",
           "failureType": "TIMED_OUT",
           "minNumberOfExecutedThings": 200,
           "thresholdPercentage": 50
        }
    ]
}
```

Le paramètre `action` spécifie l'action à entreprendre lorsque les critères d'annulation sont satisfaits. Ce paramètre est obligatoire et `CANCEL` est la seule valeur valide.

Le paramètre `failureType` spécifie les types de défaillance qui déclenchent une annulation de tâche. Les valeurs valides sont `FAILED`, `REJECTED`, `TIMED_OUT` et `ALL`.

Le paramètre `minNumberOfExecutedThings` spécifie le nombre d'exécutions de tâches terminées qui doivent se produire avant que le service ne vérifie si les critères d'annulation de tâche ont été satisfaits. Dans cet exemple, AWS IoT ne vérifie pas si une annulation de tâche doit se produire tant que 100 appareils au moins n'ont pas terminé les exécutions de tâche.

Le paramètre `thresholdPercentage` spécifie le nombre total d'objets exécutés qui déclenchent une annulation de tâche. Dans cet exemple, AWS IoT lance une annulation de tâche et annule le déploiement de tâche si au moins 20 % de toutes les exécutions réalisées ont échoué d'une façon ou d'une autre après 100 exécutions.

> **Note**
>
> La suppression d'exécutions de tâche affecte la valeur de calcul de l'exécution totale achevée. Lorsqu'une tâche est annulée, le service crée un `comment` et un `reasonCode` automatiques pour différencier une annulation guidée par l'utilisateur d'une annulation par interruption de tâche.

Limites des tâches

Pour plus d'informations sur les limites de tâche, consultezAWS AWS IoTPoints de terminaison et quotas de gestion des périphériquesdans leAWSRéférence générale.

Tunneling sécurisé AWS IoT

Lorsque des appareils sont déployés derrière des pare-feu restreints sur des sites distants, vous devez pouvoir accéder à ces appareils pour le dépannage, les mises à jour de configuration et d'autres tâches opérationnelles. Le tunneling sécurisé aide les clients à établir une communication bidirectionnelle avec des appareils distants via une connexion sécurisée qui est gérée par AWS IoT. Le tunneling sécurisé ne nécessite pas de mises à jour de votre règle de pare-feu entrant existante. Vous pouvez donc conserver le même niveau de sécurité que celui fourni par les règles de pare-feu sur un site distant.

Prenons l'exemple d'un capteur situé dans une usine à quelques centaines de kilomètres de distance et qui ne parvient pas à mesurer la température de l'usine. Vous pouvez utiliser le tunneling sécurisé pour ouvrir et démarrer rapidement une session sur ce capteur. Après avoir identifié le problème (par exemple, un fichier de configuration incorrect), vous pouvez réinitialiser le fichier et redémarrer le capteur via la même session. Par rapport à un dépannage plus traditionnel (par exemple, envoyer un technicien à l'usine pour vérifier le capteur), le tunneling sécurisé réduit le temps de réponse aux incidents et le temps de récupération ainsi que les coûts d'exploitation.

Concepts de tunneling sécurisés

Jeton d'accès client (CAT)

: Une paire de jetons générée par le tunneling sécurisé lors de la création d'un nouveau tunnel. Le CAT est utilisé par les appareils source et de destination pour se connecter au service de tunneling sécurisé.

Application de destination

: Application qui s'exécute sur l'appareil de destination. L'application de destination peut être, par exemple, un démon SSH permettant d'établir une session SSH à l'aide du tunneling sécurisé.

Appareil de destination

: Appareil distant auquel vous souhaitez accéder.

Agent d'appareil

: Application IoT qui se connecte à la passerelle d'appareils AWS IoT et écoute les nouvelles notifications de tunnel via MQTT.

Proxy local

: Proxy logiciel qui s'exécute sur les appareils source et de destination et relaie un flux de données entre le service de tunneling sécurisé et l'application d'appareil. Le proxy local peut être exécuté en mode source ou en mode destination. Pour plus d'informations, consultez Proxy local (p. 721).

Appareil source

: Appareil qu'un opérateur utilise pour lancer une session vers l'appareil de destination, généralement un ordinateur portable ou un ordinateur de bureau.

Tunnel

: Une voie logique via AWS IoT qui active une communication bidirectionnelle entre un appareil source et un appareil de destination.

Didacticiel sur le tunneling sécurisé AWS IoT

Dans ce didacticiel, vous ouvrez un tunnel et l'utilisez pour démarrer une session SSH sur un appareil distant. L'appareil distant se trouve derrière des pare-feu qui bloquent l'ensemble du trafic entrant, empêchant l'accès SSH direct dans l'appareil. Avant de commencer, assurez-vous que vous comprenez comment enregistrer un appareil dans le registre AWS IoT et connecter un appareil à la passerelle d'appareils AWS IoT.

Prerequisites

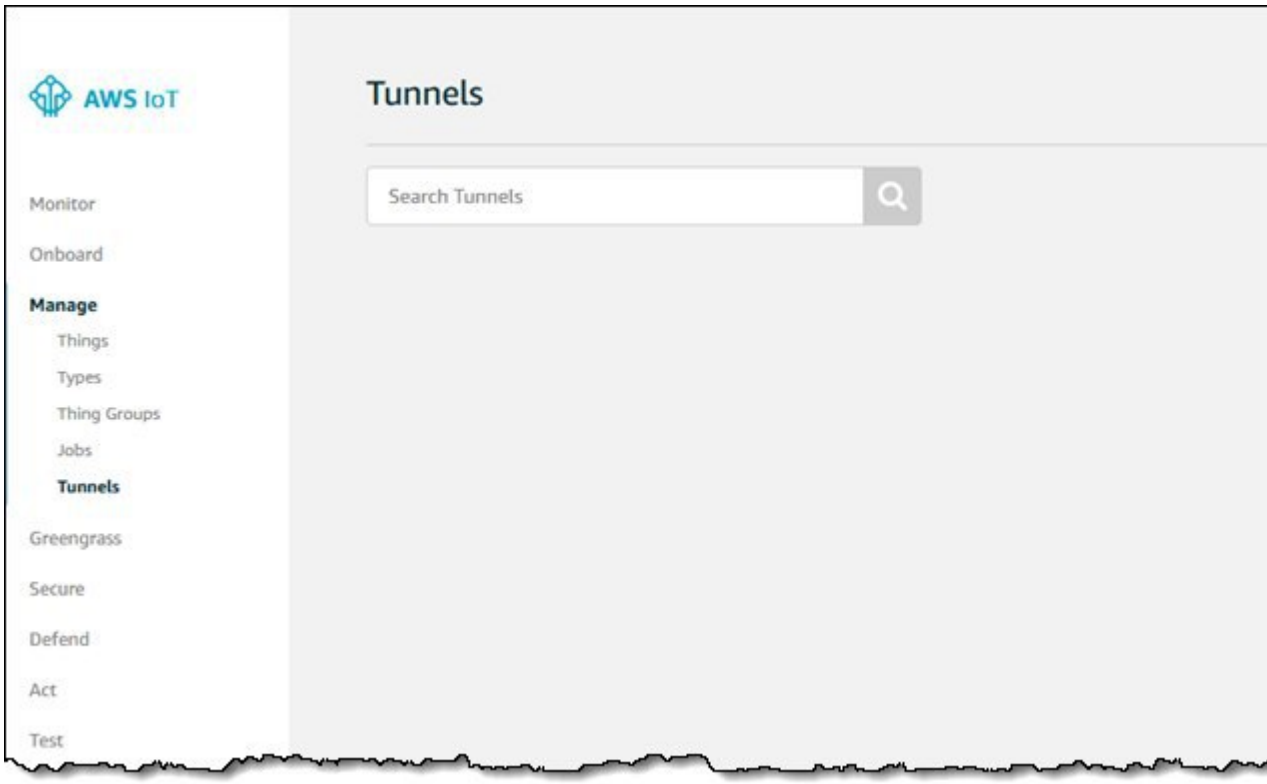
- Les pare-feu derrière l'appareil distant doivent permettre le trafic sortant sur le port 443.
- Vous avez créé un objet IoT nommé `RemoteDeviceA` dans le registre AWS IoT.
- Vous disposez d'un agent d'appareil IoT qui s'exécute sur l'appareil distant qui se connecte à la passerelle d'appareils AWS IoT et est configuré avec une inscription de rubrique MQTT. Ce didacticiel inclut un extrait de code qui vous montre comment implémenter un agent. Pour plus d'informations, consultez Extrait de l'agent IoT (p. 726).
- Vous devez disposer d'un démon SSH s'exécutant sur l'appareil distant.
- Vous avez téléchargé le code source du proxy local à partir de GitHub et l'avez conçu pour la plateforme de votre choix. Dans ce didacticiel, nous utilisons `localproxy` pour nous référer au fichier exécutable du proxy local.

Ouvrir un tunnel

Si vous configurez la destination lorsque vous appelez `OpenTunnel`, le service Secure Tunneling fournit le jeton d'accès du client de destination à l'appareil distant via MQTT et la rubrique MQTT réservée (`$aws/things/RemoteDeviceA/tunnels/notify`). Pour plus d'informations, consultez Rubriques réservées (p. 91). À la réception du message MQTT, l'agent IoT sur l'appareil distant démarre le proxy local en mode destination. Vous pouvez omettre la configuration de destination si vous souhaitez transmettre le jeton d'accès client de destination à l'appareil distant via une autre méthode. Pour plus d'informations, consultez Configuration d'un appareil distant (p. 727).

Pour ouvrir un tunnel dans la console

1. DansAWS IoTconsole, accédez àGérerandTunnel.



2. Tâche de sélectionOuvrir Nouveau.
3. Dans la pageOuvrez un nouveau tunnel sécurisé, entrez ce qui suit :

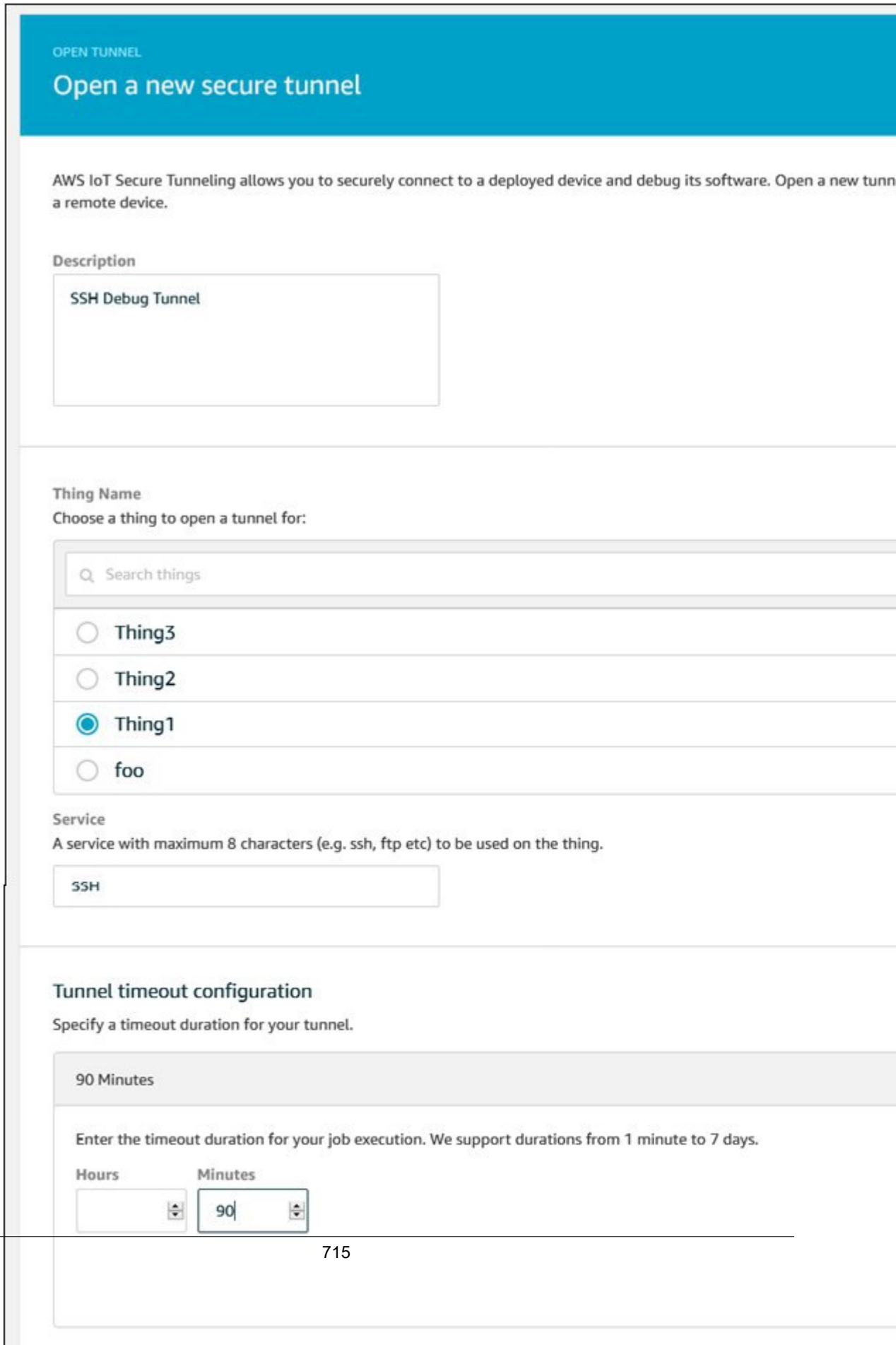
   a. Description : une description pour votre tunnel.

   b. Chose Name : la chose pour laquelle vous voulez ouvrir un tunnel.

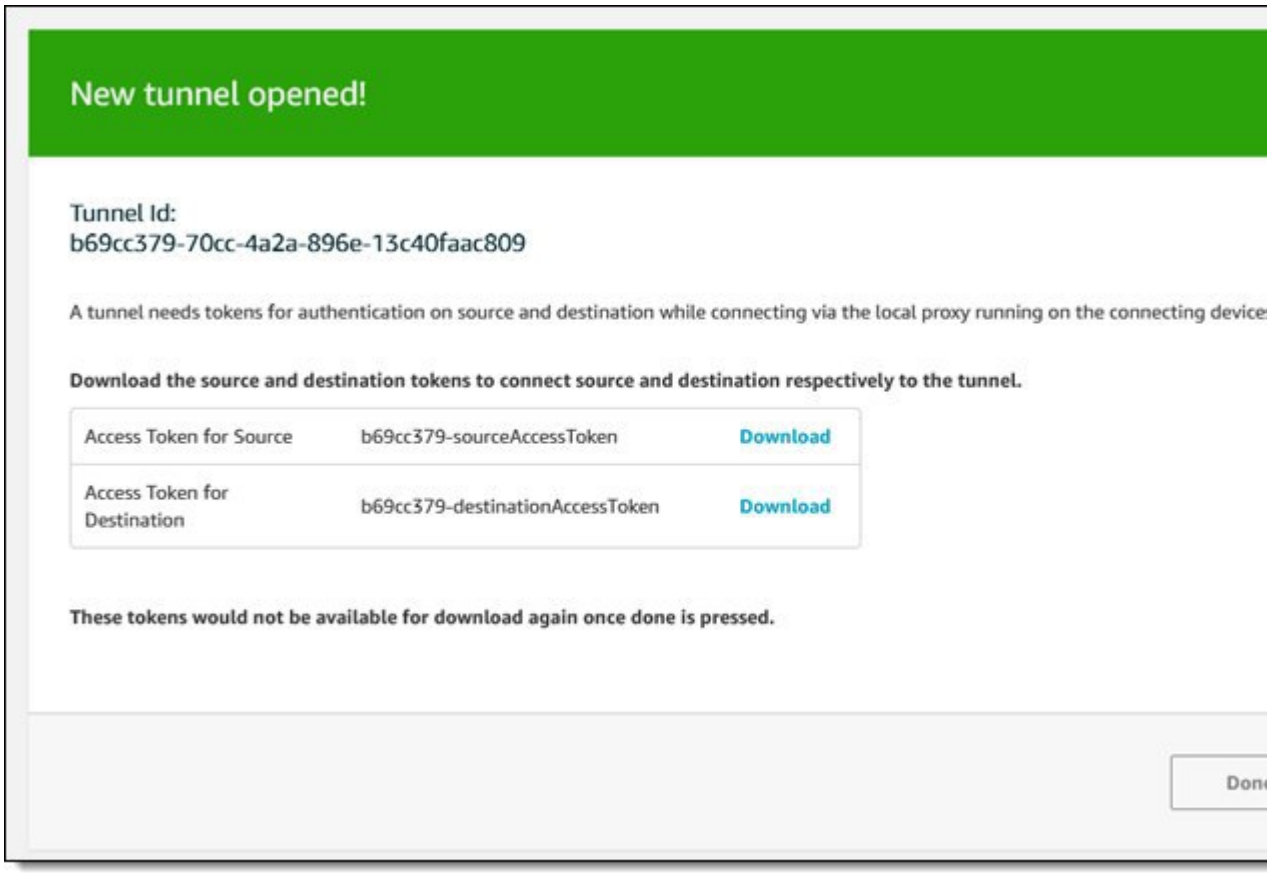
   c. Service : un service à utiliser sur la chose (par exemple ssh, ftp, etc.).

   d. Configuration du délai d'expiration du tunnel : spécifiez une durée d'expiration pour votre tunnel.

   e. Tages de ressources : appliquez des balises à vos ressources pour pouvoir les organiser et les identifier. Contient une paire clé-valeur sensible à la casse.


OPEN TUNNEL
Open a new secure tunnel
AWS IoT Secure Tunneling allows you to securely connect to a deployed device and debug its software. Open a new tunn
a remote device.
Description
SSH Debug Tunnel
Thing Name
Choose a thing to open a tunnel for:
Search things
Thing3
Thing2
Thing1
foo
Service
A service with maximum 8 characters (e.g. ssh, ftp etc) to be used on the thing.
SSH
Tunnel timeout configuration
Specify a timeout duration for your tunnel.
90 Minutes
Enter the timeout duration for your job execution. We support durations from 1 minute to 7 days.
Hours
Minutes
90

4. Téléchargez les jetons d'accès client pour la source et la destination.



5. Sélectionnez Done (Terminé).

Démarrer le proxy local

Ouvrez un terminal sur votre ordinateur portable, copiez le jeton d'accès client source et utilisez-le pour démarrer le proxy local en mode source. Dans la commande suivante, le proxy local est configuré pour écouter les nouvelles connexions sur le port 5555.

./localproxy -r us-east-1 -s 5555 -t ***source-client-access-token***

> **Note**
>
> La . Région AWS Dans cette commande doit être identique Région AWS où le tunnel a été créé.

-r

Spécifie le Région AWS où votre tunnel est créé.

-s

Spécifie le port auquel le proxy doit se connecter.

-t

Spécifie le texte du jeton client.

Note

Si vous recevez l'erreur suivante, configurez le chemin de l'autorité de certification. Pour de plus amples informations, veuillez consulter GitHub.
`Could not perform SSL handshake with proxy server: certificate verify failed`

Démarrer une session SSH

Ouvrez un autre terminal et utilisez la commande suivante pour démarrer une nouvelle session SSH en vous connectant au proxy local sur le port 5555.

ssh ***username***@localhost -p 5555

Vous pouvez être invité à entrer un mot de passe pour la session SSH. Lorsque vous avez terminé avec la session SSH, tapez **`exit`** pour fermer la session.

Fermeture du tunnel

1. Ouvrez la console AWS IoT.
2. Choisissez le tunnel, puis, dans Actions, choisissez Close (Fermer). La fermeture du tunnel entraîne la fermeture des deux instances proxy locales.

Cycle de vie des tunnels sécurisés

Démonstration du tunneling sécurisé AWS IoT

Pour consulter une démonstration rapide du tunneling sécurisé AWS IoT, utilisez notre démonstration du tunneling sécurisé AWS IoT sur GitHub.

L'état des tunnels peut être :

- `OPEN`
- `CLOSED`

L'état des connexions peut être :

- `CONNECTED`
- `DISCONNECTED`

Lorsque vous ouvrez un tunnel, son état est `OPEN`. L'état de connexion source et de destination du tunnel est défini sur `DISCONNECTED`. Lorsqu'un appareil (source ou de destination) se connecte au tunnel, l'état de connexion correspondant devient `CONNECTED`, alors que l'état du tunnel reste `OPEN`. Lorsqu'un appareil se déconnecte du tunnel, l'état de connexion correspondant redevient `DISCONNECTED`. Un appareil peut se connecter à un tunnel et s'en déconnecter à plusieurs reprises tant que l'état du tunnel reste `OPEN`.

L'état d'un tunnel devient `CLOSED` lorsque vous appelez `CloseTunnel` ou que l'état du tunnel reste `OPEN` pendant plus longtemps que la valeur `MaxLifetimeTimeout`. Vous pouvez configurer `MaxLifetimeTimeout` lorsque vous appelez `OpenTunnel`. Si vous ne spécifiez pas de valeur, `MaxLifetimeTimeout` est défini par défaut sur 12 heures.

Une fois que l'état d'un tunnel est `CLOSED`, le tunnel peut être visible dans la console AWS IoT pendant au moins trois heures avant d'être supprimé. Lorsque le tunnel est visible, vous pouvez appeler `DescribeTunnel` et `ListTunnels` pour afficher les métadonnées du tunnel. Un tunnel ne peut pas être

rouvert lorsque son état est `CLOSED`. Le service de tunneling sécurisé rejette les tentatives de connexion à un tunnel dont l'état est `CLOSED`.

Contrôle de l'accès aux tunnels

Le service de tunnel sécurisé fournit les actions, ressources et clés de contexte de condition spécifiques au service suivantes en vue de leur utilisation dans les stratégies d'autorisation IAM.

Conditions préalables à l'accès au tunnel

- Découvrez comment sécuriserAWSRessources en utilisantStratégies IAM.
- Découvrez comment créer et évaluer des conditions IAM.
- Découvrez comment sécuriserAWSRessources utilisantbalise de ressource.

iot:OpenTunnel

L'action de stratégie `iot:OpenTunnel` accorde à un mandataire l'autorisation d'appeler OpenTunnel. Vous devez spécifier l'ARN du tunnel générique`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`dans le`Resource`élément de la déclaration de politique IAM. Vous pouvez spécifier un ARN de chose (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:thing/ <`*`thing-name`*) dans la`Resource`élément de la déclaration de stratégie IAM à gérer`OpenTunnel`pour des objets IoT spécifiques.

Par exemple, la déclaration de stratégie suivante vous permet d'ouvrir un tunnel vers l'objet IoT nommé `TestDevice`.

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*",
        "arn:aws:iot:aws-region:aws-account-id:thing/TestDevice"
    ]
}
```

L'action de stratégie `iot:OpenTunnel` prend en charge les clés de condition suivantes :

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `aws:RequestTag/`*`clé-balise`*
- `aws:SecureTransport`
- `aws:TagKeys`

La déclaration de stratégie suivante vous permet d'ouvrir un tunnel à la chose si l'objet appartient à un groupe d'objets dont le nom commence par`TestGroup`et le service de destination configuré sur le tunnel est SSH.

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
```

```
    "Condition": {
        "ForAnyValue:StringLike": {
            "iot:ThingGroupArn": [
                "arn:aws:iot:aws-region:aws-account-id:thinggroup/TestGroup*"
            ]
        },
        "ForAllValues:StringEquals": {
            "iot:TunnelDestinationService": [
                "SSH"
            ]
        }
    }
}
```

Vous pouvez également utiliser des balises de ressources pour contrôler l'autorisation d'ouvrir des tunnels. Par exemple, la déclaration de stratégie suivante permet d'ouvrir un tunnel si la clé de balise `Owner` est présente et que sa valeur est `Admin` et qu'aucune autre balise n'est spécifiée.

```
{
    "Effect": "Allow",
    "Action": "iot:OpenTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:RequestTag/Owner": "Admin"
        },
        "ForAllValues:StringEquals": {
            "aws:TagKeys": "Owner"
        }
    }
}
```

iot:DescribeTunnel

L'action de stratégie `iot:DescribeTunnel` accorde à un mandataire l'autorisation d'appeler DescribeTunnel. Vous pouvez spécifier un ARN de tunnel entièrement qualifié (par exemple,`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`) ou utilisez l'ARN du tunnel générique (`arn:aws:iot:aws-region:aws-account-id:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:DescribeTunnel` prend en charge les clés de condition suivantes :

- `aws:ResourceTag/tag-key`
- `aws:SecureTransport`

La déclaration de stratégie suivante vous permet d'appeler `DescribeTunnel` si le tunnel demandé est marqué avec la clé `Owner` ayant la valeur `Admin`.

```
{
    "Effect": "Allow",
    "Action": "iot:DescribeTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/Owner": "Admin"
        }
```

```
    }
}
```

iot:ListTunnels

L'action de stratégie `iot:ListTunnels` accorde à un mandataire l'autorisation d'appeler ListTunnels. Vous devez spécifier l'ARN du tunnel générique (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM. Gestion d'`ListTunnels`autorisation sur des objets IoT sélectionnés, vous pouvez également spécifier l'ARN d'un objet (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:thing/`*`thing-name`*) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:ListTunnels` prend en charge la clé de condition suivante :

- `aws:SecureTransport`

La déclaration de stratégie suivante vous permet de répertorier les tunnels pour l'objet nommé `TestDevice`.

```
{
    "Effect": "Allow",
    "Action": "iot:ListTunnels",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*",
        "arn:aws:iot:aws-region:aws-account-id:thing/TestDevice"
    ]
}
```

iot:ListTagsForResource

L'action de stratégie `iot:ListTagsForResource` accorde à un mandataire l'autorisation d'appeler `ListTagsForResource`. Vous pouvez spécifier un ARN de tunnel entièrement qualifié (`arn:aws:iot:`*`aws-region`*`: `*`aws-account-id`*`:tunnel/`*`tunnel-id`*) ou utilisez l'ARN du tunnel générique (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:ListTagsForResource` prend en charge la clé de condition suivante :

- `aws:SecureTransport`

iot:CloseTunnel

L'action de stratégie `iot:CloseTunnel` accorde à un mandataire l'autorisation d'appeler CloseTunnel. Vous pouvez spécifier un ARN de tunnel entièrement qualifié (`arn:aws:iot:`*`aws-region`*`: `*`aws-account-id`*`:tunnel/`*`tunnel-id`*) ou utilisez l'ARN du tunnel générique (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:CloseTunnel` prend en charge les clés de condition suivantes :

- `iot:Delete`
- `aws:ResourceTag/`*`tag-key`*
- `aws:SecureTransport`

La déclaration de stratégie suivante vous permet d'appeler `CloseTunnel` si le paramètre `Delete` de la demande est `false` et si le tunnel demandé est balisé avec une clé `Owner` ayant la valeur `QATeam`.

```
{
    "Effect": "Allow",
    "Action": "iot:CloseTunnel",
    "Resource": [
        "arn:aws:iot:aws-region:aws-account-id:tunnel/*"
    ],
    "Condition": {
        "Bool": {
            "iot:Delete": "false"
        },
        "StringEquals": {
            "aws:ResourceTag/Owner": "QATeam"
        }
    }
}
```

iot:TagResource

L'action de stratégie `iot:TagResource` accorde à un mandataire l'autorisation d'appeler `TagResource`. Vous pouvez spécifier un ARN de tunnel entièrement qualifié (`arn:aws:iot:`*`aws-region`*`: `*`aws-account-id`*`:tunnel/`*`tunnel-id`*) ou utilisez l'ARN du tunnel générique (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:TagResource` prend en charge la clé de condition suivante :

- `aws:SecureTransport`

iot:UntagResource

L'action de stratégie `iot:UntagResource` accorde à un mandataire l'autorisation d'appeler `UntagResource`. Vous pouvez spécifier un ARN de tunnel entièrement qualifié (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/`*`tunnel-id`*) ou utilisez l'ARN du tunnel générique (`arn:aws:iot:`*`aws-region`*`:`*`aws-account-id`*`:tunnel/*`) dans la`Resource`élément de la déclaration de politique IAM.

L'action de stratégie `iot:UntagResource` prend en charge la clé de condition suivante :

- `aws:SecureTransport`

Pour de plus amples informations sur les groupes de sécurité AWS IoT, veuillez consulter Gestion des identités et des accès pour AWS IoT (p. 315).

Proxy local

Le proxy local est un processus qui agit en tant que destinataire ou expéditeur de connexions TCP entrantes. Il transmet les données envoyées par l'application de l'appareil via le service de tunneling sécurisé via une connexion sécurisée WebSocket. Vous pouvez télécharger la source du proxy local à partir de GitHub. Le proxy local peut s'exécuter en deux modes : `source` ou `destination`. En mode source, le proxy local s'exécute sur le même appareil ou réseau que l'application cliente qui initie la connexion TCP. En mode destination, le proxy local s'exécute sur l'appareil distant, avec l'application de destination. Actuellement, un même tunnel ne peut prendre en charge qu'une seule connexion TCP à la fois.

Le proxy local établit d'abord une connexion au service de tunneling sécurisé. Lorsque vous démarrez le proxy local, utilisez la commande`-r`pour spécifier l'argument Région AWS dans lequel le tunnel est

ouvert. Utilisez l'argument `-t` pour transmettre le jeton d'accès client source ou de destination renvoyé par le `OpenTunnel`. Deux proxy locaux utilisant la même valeur de jeton d'accès client ne peuvent pas être connectés en même temps.

Une fois la connexion WebSocket établie, le proxy local effectue des comportements en mode source ou en mode destination, en fonction de sa configuration.

Par défaut, le proxy local tente de se reconnecter au service de tunneling sécurisé si des erreurs d'E/S se produisent ou si la connexion WebSocket est se ferme de façon inattendue. Cela provoque la fermeture de la connexion TCP. Si des erreurs de socket TCP se produisent, le proxy local envoie un message via le tunnel pour avertir l'autre partie de fermer sa connexion TCP. Par défaut, le proxy local utilise toujours la communication SSL.

Après avoir utilisé le tunnel, vous pouvez arrêter le processus de proxy local en toute sécurité. Nous vous recommandons de fermer explicitement le tunnel en appelant `CloseTunnel`. Il est possible que les clients de tunnel actifs ne soient pas fermés immédiatement après l'appel `CloseTunnel`.

Meilleures pratiques en matière de sécurité du proxy local

Lors de l'exécution du proxy local, suivez les bonnes pratiques de sécurité suivantes :

- Évitez d'utiliser l'argument de proxy local `-t` pour transmettre un jeton d'accès. Nous vous recommandons d'utiliser la variable d'environnement `AWSIOT_TUNNEL_ACCESS_TOKEN` pour définir le jeton d'accès pour le proxy local.
- Exécutez l'exécutable du proxy local avec moindres privilèges dans le système d'exploitation ou l'environnement.
  - Évitez d'exécuter le proxy local en tant qu'administrateur sous Windows.
  - Évitez d'exécuter le proxy local en tant que racine sur Linux et macOS.
- Envisagez d'exécuter le proxy local sur des hôtes, conteneurs, environnements de test, environnements de test, chroot jail ou un environnement virtualisé distincts.
- Créez le proxy local avec les indicateurs de sécurité pertinents correspondants à votre chaîne d'outils.
- Sur les appareils dotés de plusieurs interfaces réseau, utilisez l'argument `-b` pour lier le socket TCP à l'interface réseau utilisée pour communiquer avec l'application de destination.

Flux de données multiplexes dans un tunnel sécurisé

Vous pouvez utiliser plusieurs flux de données par tunnel à l'aide de la fonction de multiplexage Secure Tunneling. Avec le multiplexage, vous pouvez dépanner les périphériques à l'aide de plusieurs connexions ou ports (par exemple, un navigateur Web qui nécessite l'envoi de plusieurs flux de données HTTP et SSH). Vous pouvez également réduire votre charge opérationnelle en éliminant la nécessité de créer, déployer et démarrer plusieurs serveurs proxy locaux ou d'ouvrir plusieurs tunnels sur le même périphérique.

Exemple de cas d'utilisation

Vous pouvez utiliser la fonctionnalité de multiplexage dans le cas où un périphérique sur le terrain nécessite plusieurs connexions au périphérique afin de le résoudre correctement. Par exemple, vous devrez peut-être vous connecter à une application Web sur l'appareil pour modifier certains paramètres réseau, tout en exécutant simultanément des commandes shell via le terminal pour vérifier que le

périphérique fonctionne correctement avec les nouveaux paramètres réseau. Dans ce scénario, vous devrez peut-être vous connecter au périphérique via HTTP et SSH et transférer deux flux de données parallèles afin d'accéder simultanément à l'application Web et au terminal. Grâce à la fonction multiplexage, ces deux flux indépendants peuvent être transférés simultanément sur le même tunnel.

Comment configurer un tunnel multiplexé

La procédure suivante vous explique comment configurer un tunnel multiplexé pour le dépannage des périphériques à l'aide d'applications nécessitant des connexions à plusieurs ports. Vous allez configurer un tunnel avec deux flux multiplexés : un flux HTTP et un flux SSH.

1. Tout d'abord, configurez le périphérique de destination avec des fichiers de configuration. Des fichiers de configuration peuvent être fournis sur le périphérique si les mappages de port sont peu susceptibles de changer. Sur le périphérique de destination, créez un répertoire de configuration appelé`config`dans le même dossier où le proxy local est en cours d'exécution. Ensuite, créez un fichier nommé`SSHSource.ini`Dans ce répertoire. Le contenu de ce fichier est le suivant :

   ```
   HTTP1 = 5555
   SSH1 = 3333
   ```

   > **Note**
   >
   > Vous pouvez ignorer cette étape si vous préférez spécifier le mappage de port via l'interface de ligne de commande ou si vous n'avez pas besoin de démarrer le proxy local sur les ports d'écoute désignés.

2. Ensuite, configurez le périphérique source avec les fichiers de configuration. Dans le même dossier que celui où le proxy local est en cours d'exécution, créez un répertoire de configuration appelé`config`et donnez au proxy local l'autorisation de lecture à ce répertoire. Ensuite, créez un fichier nommé`SSHDestination.ini`Dans ce répertoire. Le contenu de ce fichier est le suivant :

   ```
   HTTP1 = 80
   SSH1 = 22
   ```

   > **Note**
   >
   > Vous pouvez ignorer cette étape si vous préférez spécifier le mappage de port via l'interface de commande. Si c'est le cas, vous devrez mettre à jour l'Agent de tunnel (p. 726)Pour utiliser les nouveaux paramètres.

3. Ouvrir un tunnel avec l'identifiant de service`HTTP1`and`SSH1`.`thingName`est facultative si votre appareil n'est pas enregistré avecAWS IoT.

   ```
   aws iotsecuretunneling open-tunnel \
   --destination-config thingName=foo,services=HTTP1,SSH1
   ```

   Un jeton d'accès client de destination et source sera donné après cet appel. Notez que l'`destination_client_access_token`and`source_client_access_token`Pour connaître les étapes suivantes. La sortie doit ressembler à ce qui suit :

   ```
   {
    "tunnelId": "b2de92a3-b8ff-46c0-b0f2-afa28b00cecd",
    "tunnelArn": "arn:aws:iot:us-west-2:431600097591:tunnel/b2de92a3-b8ff-46c0-b0f2-
   afa28b00cecd",
    "sourceAccessToken": source_client_access_token,
    "destinationAccessToken": destination_client_access_token
   }
   ```

4. Ensuite, démarrez le proxy local de destination. Vous serez connecté au service de tunnel sécurisé lors de la livraison du jeton. Un proxy local s'exécutant sur les périphériques de destination démarre en mode de destination. Vous avez deux options pour cela :

   1. Démarrez le proxy local de destination avec les fichiers de configuration à partir de l'étape 1.

      ```
      ./localproxy -r us-east-1 -m dst -t destination_client_access_token
      ```

   2. Démarrez le proxy local de destination avec le mappage spécifié via l'interface de ligne de commande.

      ```
      ./localproxy -r us-east-1 -d HTTP1=80,SSH1=22 -t destination_client_access_token
      ```

5. Maintenant, démarrez le proxy local source. Un proxy local s'exécutant sur les périphériques source démarre en mode source. Trois possibilités s'offrent à vous :

   1. Démarrez le proxy local source avec les fichiers de configuration à partir de l'étape 2.

      ```
      ./localproxy -r us-east-1 -m src -t source_client_access_token
      ```

   2. Démarrez le proxy local source avec le mappage spécifié via l'interface de ligne de commande.

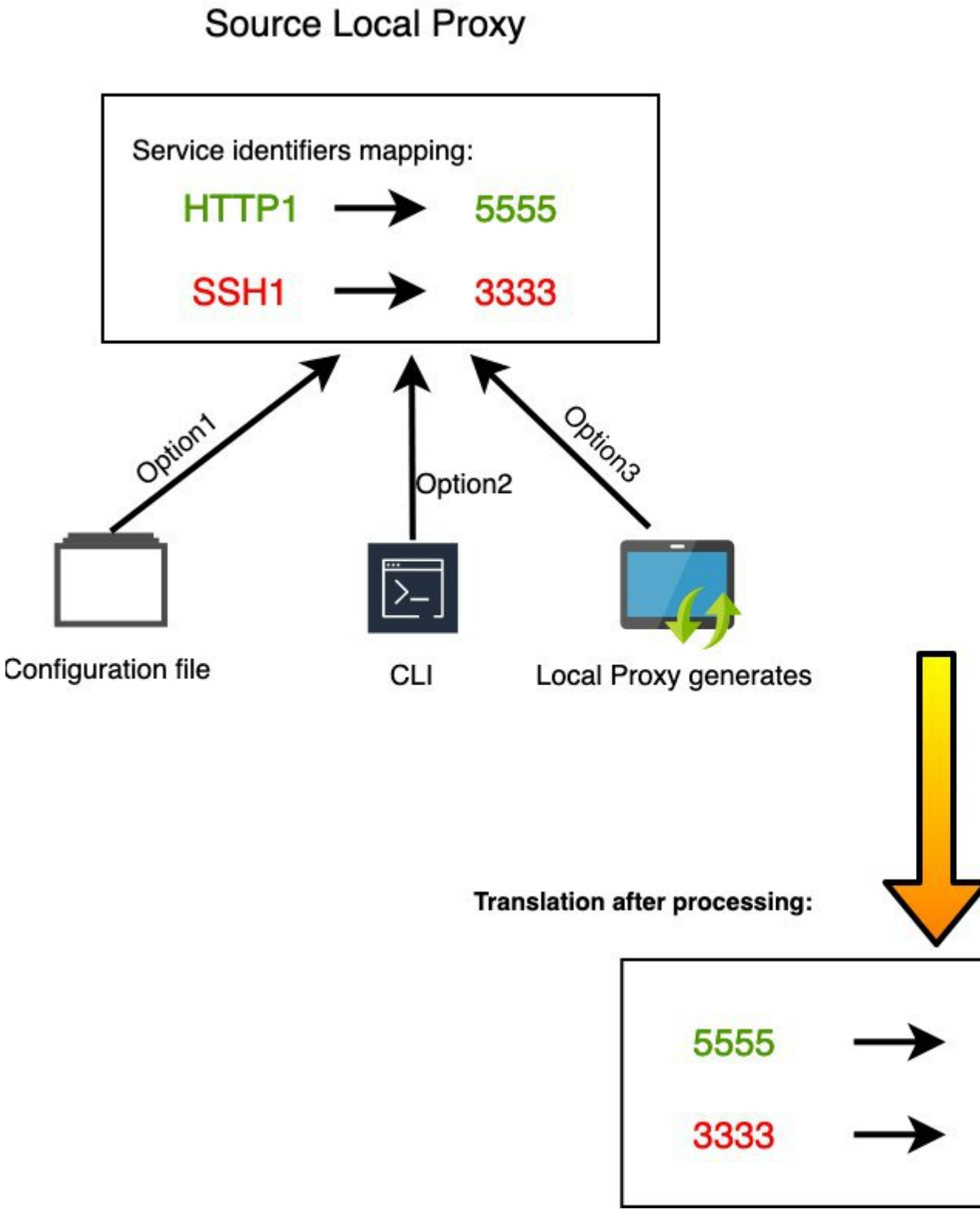
      ```
      ./localproxy -r us-east-1 -s HTTP1=5555,SSH1=3333 -t source_client_access_token
      ```

   3. Démarrez le proxy local source sans fichiers de configuration et aucun mappage spécifié à partir de l'interface de ligne de commande. Le proxy local récupère les ports disponibles pour utiliser et gérer les mappages pour vous.

      ```
      ./localproxy -r us-east-1 -m src -t source_client_access_token
      ```

6. Les données d'application de la connexion SSH et HTTP peuvent désormais être transférées simultanément sur le tunnel multiplexé. Comme le montre la carte ci-dessous, l'identificateur de service agit comme un format lisible pour traduire le mappage de port entre le périphérique source et le périphérique de destination. Avec cette configuration, le service de tunnel sécurisé :

   1. Transférer tout trafic HTTP entrant du port 5555 sur le périphérique source vers le port 80 sur le périphérique de destination.
   2. Transférer tout trafic SSH entrant du port 3333 du périphérique source vers le port 22 du périphérique de destination.


Source Local Proxy
Service identifiers mapping:
HTTP1
5555
SSH1
3333
Option1
Option2
Option3
Configuration file
CLI
Local Proxy generates
Translation after processing:
5555
3333

Extrait de l'agent IoT

L'agent IoT est utilisé pour recevoir le message MQTT qui inclut le jeton d'accès client et pour démarrer un proxy local sur l'appareil distant. Vous devez installer et exécuter l'agent IoT sur l'appareil distant si vous souhaitez que le service de tunneling sécurisé fournisse le jeton d'accès client. L'agent IoT doit s'abonner à la rubrique MQTT IoT réservée suivante :

`$aws/things/thing-name/tunnels/notify`

Où `thing-name` est le nom de l'objet IoT associé à l'appareil distant.

Voici un exemple de charge utile de message MQTT :

```
{
    "clientAccessToken": "destination-client-access-token",
    "clientMode": "destination",
    "region": "aws-region",
    "services": ["destination-service"]
}
```

Après avoir reçu un message MQTT, l'agent IoT doit démarrer un proxy local sur l'appareil distant avec les paramètres appropriés.

Le code Java suivant montre comment utiliser le kit SDK pour appareils AWS IoT et ProcessBuilder à partir de la bibliothèque Java pour créer un agent IoT simple pour fonctionner avec le service de tunneling sécurisé.

```
// Find the IoT device endpoint for your
    Compte AWS

final String endpoint = iotClient.describeEndpoint(new
 DescribeEndpointRequest().withEndpointType("iot:Data-ATS")).getEndpointAddress();

// Instantiate the IoT Agent with your AWS credentials
final String thingName = "RemoteDeviceA";
final String tunnelNotificationTopic = String.format("$aws/things/%s/tunnels/notify",
 thingName);
final AWSIotMqttClient mqttClient = new AWSIotMqttClient(endpoint, thingName,
                 "your_aws_access_key", "your_aws_secret_key");

try {
    mqttClient.connect();
    final TunnelNotificationListener listener = new
 TunnelNotificationListener(tunnelNotificationTopic);
    mqttClient.subscribe(listener, true);
}
finally {
    mqttClient.disconnect();
}

private static class TunnelNotificationListener extends AWSIotTopic {
    public TunnelNotificationListener(String topic) {
        super(topic);
    }

    @Override
    public void onMessage(AWSIotMessage message) {
        try {
            // Deserialize the MQTT message
            final JSONObject json = new JSONObject(message.getStringPayload());
```

```
            final String accessToken = json.getString("clientAccessToken");
            final String region = json.getString("region");

            final String clientMode = json.getString("clientMode");
            if (!clientMode.equals("destination")) {
                throw new RuntimeException("Client mode " + clientMode + " in the MQTT
 message is not expected");
            }

            final JSONArray servicesArray = json.getJSONArray("services");
            if (servicesArray.length() > 1) {
                throw new RuntimeException("Services in the MQTT message has more than 1
 service");
            }
            final String service = servicesArray.get(0).toString();
            if (!service.equals("SSH")) {
                throw new RuntimeException("Service " + service + " is not supported");
            }

            // Start the destination local proxy in a separate process to connect to the
 SSH Daemon listening port 22
            final ProcessBuilder pb = new ProcessBuilder("localproxy",
                        "-t", accessToken,
                        "-r", region,
                        "-d", "localhost:22");
            pb.start();
        }
        catch (Exception e) {
            log.error("Failed to start the local proxy", e);
        }
    }
}
```

Configuration d'un appareil distant

Si vous souhaitez remettre le jeton d'accès client de destination à l'appareil autrement que par l'abonnement à la rubrique MQTT IoT réservée, vous pouvez avoir besoin de deux composants sur l'appareil distant :

- Un écouteur de jeton d'accès client de destination.
- Un proxy local.

L'écouteur de jeton d'accès client de destination doit fonctionner avec le mécanisme de remise de jeton d'accès client de votre choix. Il doit pouvoir démarrer un proxy local en mode destination.

Mise en service des appareils

AWSfournit plusieurs façons différentes de provisionner un périphérique et d'y installer des certificats client uniques. Cette section décrit chaque méthode et comment sélectionner la meilleure solution IoT pour votre solution IoT. Nous proposons également desun outil interactif pour guider votre décision. Ces options sont détaillées dans le livre blanc intitulé*Fabrication et provisionnement de périphériques avec les certificats X.509 dans AWS IoT Core* .

Sélectionnez l'option qui correspond le mieux à votre situation

- Vous pouvez installer des certificats sur des appareils IoT avant qu'ils ne soient livrés

  Si vous pouvez installer en toute sécurité des certificats client uniques sur vos appareils IoT avant qu'ils ne soient remis à l'utilisateur final, vous souhaitez utiliserjuste-à-tempsprovisioning (JITP) (p. 737)oujuste-à-tempsInscription (JITR) (p. 247).

  En utilisant JITP et JITR, l'autorité de certification utilisée pour signer le certificat d'appareil est enregistrée avecAWS IoTet est reconnu parAWS IoTlorsque l'appareil se connecte pour la première fois. Le périphérique est provisionné dansAWS IoTsur sa première connexion en utilisant les détails de son modèle de provisioning.

  Pour plus d'informations sur une seule chose, JITP, JITR et le provisionnement en bloc des périphériques disposant de certificats uniques, consultezthe section called “Mise en service d'appareils disposant de certificats d'appareils” (p. 736).
- Les utilisateurs finaux ou les installateurs peuvent utiliser une application pour installer des certificats sur leurs appareils IoT

  Si vous ne pouvez pas installer en toute sécurité des certificats client uniques sur votre appareil IoT avant qu'ils ne soient livrés à l'utilisateur final, mais que l'utilisateur final ou un programme d'installation peut utiliser une application pour enregistrer les appareils et installer les certificats de périphérique uniques, vous souhaitez utiliser leMise en service par utilisateur approuvé (p. 732)processus.

  L'utilisation d'un utilisateur approuvé, tel qu'un utilisateur final ou un programme d'installation disposant d'un compte connu, peut simplifier le processus de fabrication de l'appareil. Au lieu d'un certificat client unique, les périphériques disposent d'un certificat temporaire qui permet au périphérique de se connecter àAWS IoTpendant seulement 5 minutes. Pendant cette fenêtre de 5 minutes, l'utilisateur approuvé obtient un certificat client unique avec une durée de vie plus longue et l'installe sur l'appareil. La durée de vie limitée du certificat de réclamation minimise le risque de compromettre un certificat.

  Pour plus d'informations, consultez the section called “Allocation par utilisateur approuvé” (p. 732).
- Les utilisateurs finaux NE PEUVENT PAS utiliser une application pour installer des certificats sur leurs appareils IoT

  Si aucune des options précédentes ne fonctionne dans votre solution IoT, leprovisionnement par revendication (p. 730)est une option. Grâce à ce processus, vos appareils IoT disposent d'un certificat de réclamation qui est partagé par d'autres appareils de la flotte. La première fois qu'un appareil se connecte à un certificat de réclamation,AWS IoTenregistre le périphérique à l'aide de son modèle de provisioning et émet son certificat client unique pour un accès ultérieur àAWS IoT

  Cette option permet le provisionnement automatique d'un périphérique lorsqu'il se connecte àAWS IoT, mais pourrait présenter un risque plus important dans le cas d'un certificat de réclamation compromis. Si un certificat de réclamation est compromis, vous pouvez le désactiver. La désactivation du certificat de revendication empêche l'enregistrement futur de tous les appareils avec ce certificat de revendication.

Toutefois, la désactivation du certificat de revendication ne bloque pas les périphériques qui ont déjà été provisionnés.

Pour plus d'informations, consultez the section called “Allocation par revendication” (p. 730).

Mise en service des appareils dansAWS IoT

Lorsque vous allouez un appareil avec AWS IoT, vous devez créer des ressources afin que vos appareils et AWS IoT puissent communiquer en toute sécurité. D'autres ressources peuvent être créées pour vous aider à gérer votre flotte d'appareils. Les ressources suivantes peuvent être créées au cours du processus de mise en service :

- Un objet IoT.

  Les objets IoT sont des entrées dans le registre d'appareils AWS IoT. Chaque objet a un nom et un ensemble d'attributs uniques, et est associé à un appareil physique. Les objets peuvent être définis à l'aide d'un type d'objet ou regroupées en groupes d'objets. Pour plus d'informations, consultez Gestion des appareils avec AWS IoT (p. 203).

  Bien qu'elle ne soit pas nécessaire, la création d'un objet vous permet de gérer votre parc d'appareils plus efficacement en recherchant les appareils par type d'objet, groupe d'objets et attributs d'objet. Pour plus d'informations, consultez Service d'indexation de flotte (p. 758).
- Un certificat X.509.

  Les appareils utilisent des certificats X.509 pour effectuer une authentification mutuelle avec AWS IoT. Vous pouvez enregistrer un certificat existant ou demander à AWS IoT de générer et d'enregistrer un nouveau certificat pour vous. Vous associez un certificat à un appareil en l'attachant à l'objet qui représente l'appareil. Vous devez également copier le certificat et la clé privée associée sur l'appareil. Les appareils présentent le certificat lors de la connexion à AWS IoT. Pour plus d'informations, consultez Authentication (p. 232).
- Une stratégie IoT.

  Les stratégies IoT définissent les opérations qu'un appareil peut effectuer dans AWS IoT. Les stratégies IoT sont attachées aux certificats des appareils. Lorsqu'un appareil présente le certificat à AWS IoT, les autorisations spécifiées dans la stratégie lui sont octroyées. Pour plus d'informations, consultez Authorization (p. 268). Chaque appareil a besoin d'un certificat pour communiquer avec AWS IoT.

AWS IoT prend en charge la mise en service automatisée de flotte à l'aide de modèles de mise en service. Les modèles d'allocation décrivent les ressources requises par AWS IoT pour allouer votre appareil. Les modèles contiennent des variables qui vous permettent d'utiliser un modèle pour mettre en service plusieurs appareils. Lorsque vous allouez un appareil, vous spécifiez des valeurs pour les variables spécifiques à l'appareil à l'aide d'un dictionnaire ou d'une carte. Pour mettre en service un autre appareil, spécifiez de nouvelles valeurs dans le dictionnaire.

Vous pouvez utiliser l'allocation automatisée, que vos appareils disposent de certificats uniques (et de leur clé privée associée) ou non.

API de mise en service de flotte

Il existe plusieurs catégories d'API utilisées dans le provisionnement de flotte :

- Ces fonctions de plan de contrôle créent et gèrent les modèles de provisionnement de flotte et configurent les stratégies des utilisateurs approuvés.

  - CreateProvisioningTemplate
  - CreateProvisioningTemplateVersion
  - DeleteProvisioningTemplate
  - DeleteProvisioningTemplateVersion
  - DescribeProvisioningTemplate
  - DescribeProvisioningTemplateVersion
  - ListProvisioningTemplates
  - ListProvisioningTemplateVersions
  - UpdateProvisioningTemplate
- Les utilisateurs approuvés peuvent utiliser cette fonction de plan de contrôle pour générer une demande d'intégration temporaire. Cette demande temporaire est transmise à l'appareil lors de la configuration Wi-Fi ou d'une méthode similaire.
  - CreateProvisioningClaim.
- API MQTT utilisée au cours du processus de provisionnement par des périphériques avec un certificat de demande de provisionnement incorporé dans un périphérique ou transmis par un utilisateur approuvé.
  - the section called “CreateCertificateFromCsr” (p. 752)
  - the section called “CreateKeysAndCertificate” (p. 753)
  - the section called “RegisterThing” (p. 755)

Mise en service d'appareils qui ne disposent pas de certificats d'appareils à l'aide de la mise en service de flotte

En utilisantAWS IoTMise en service de flotte,AWS IoTpeut générer et fournir en toute sécurité des certificats d'appareil et des clés privées à vos appareils lorsqu'ils se connectent àAWS IoTPour la première fois.AWS IoTfournit des certificats clients signés par l'autorité de certification racine Amazon.

Il existe deux façons d'utiliser la mise en service de flotte :

- Par revendication
- Par un utilisateur de confiance

Allocation par revendication

Les appareils peuvent être fabriqués avec un certificat de revendication de mise en service et une clé privée (qui sont des informations d'identification à usage spécial) intégrés. Si ces certificats sont enregistrés avec AWS IoT, le service peut les échanger contre des certificats d'appareil uniques que l'appareil peut utiliser pour des opérations effectuées régulièrement. Le processus comprend les étapes suivantes :

Avant de livrer l'appareil

1. Appelez`CreateProvisioningTemplate`Pour créer un modèle de mise en service. Cette API renvoie un ARN de modèle. Pour plus d'informations, consultez API MQTT de mise en service des appareils (p. 751).

   Vous pouvez également créer un modèle d'allocation de parc dans la console AWS IoT.

   a. Dans le panneau de navigation, choisissez Intégrer, puis Modèles de mise en service de flotte.

b. Choisissez Créer et suivez les invites.

2. Créez les certificats et les clés privées associées qui seront utilisés comme certificats de revendications de mise en service.
3. Enregistrez ces certificats auprès d'AWS IoT et associez une stratégie IoT qui restreint l'utilisation des certificats. L'exemple suivant de stratégie IoT limite l'utilisation du certificat associé à cette stratégie aux appareils mis en service.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Connect"],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": ["iot:Publish","iot:Receive"],
            "Resource": [
                "arn:aws:iot:aws-region:aws-account-id:topic/$aws/certificates/create/
*",
                "arn:aws:iot:aws-region:aws-account-id:topic/$aws/provisioning-
templates/templateName/provision/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": [
                "arn:aws:iot:aws-region:aws-account-id:topicfilter/$aws/certificates/
create/*",
                "arn:aws:iot:aws-region:aws-account-id:topicfilter/$aws/provisioning-
templates/templateName/provision/*"
            ]
        }
    ]
}
```

4. Accordez au service AWS IoT l'autorisation de créer ou de mettre à jour des ressources IoT telles que des objets et des certificats dans votre compte lors de la mise en service des appareils. Pour ce faire, connectez la stratégie gérée `AWSIoTThingsRegistration` à un rôle IAM (appelé le rôle de mise en service) qui approuve le mandataire du service AWS IoT.
5. Fabriquez l'appareil avec le certificat de revendication de mise en service intégré de manière sécurisée.

L'appareil est maintenant prêt à être livré là où il sera installé pour être utilisé.

> **Important**
>
> Les clés privées des revendications de mise en service doivent être sécurisées à tout moment, y compris sur l'appareil. Nous vous recommandons d'utiliserAWS IoTLes métriques et les journaux CloudWatch pour surveiller les utilisations abusives. Si vous détectez une utilisation abusive, désactivez le certificat de revendication de mise en service afin qu'il ne puisse pas être utilisé pour la mise en service des appareils.

Pour initialiser l'appareil à utiliser

1. L'appareil utilise le AWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil (p. 1140) pour se connecter et s'authentifier auprès d'AWS IoT à l'aide du certificat de demande de mise en service qu'il a installé.

Note

Pour des raisons de sécurité, le`certificateOwnershipToken`Retourné par`CreateCertificateFromCsr` (p. 752)and`CreateKeysAndCertificate` (p. 753)expire après une heure.`RegisterThing` (p. 755)doit être appelé avant la commande`certificateOwnershipToken`Expires. Si le jeton expire, le périphérique peut appeler`CreateCertificateFromCsr` (p. 752)ou`CreateKeysAndCertificate` (p. 753)Pour générer un nouveau certificat.

2. L'appareil obtient un certificat permanent et une clé privée en utilisant l'une de ces options. L'appareil utilisera le certificat et la clé pour toute authentification future auprès d'AWS IoT.

   a. Appelez `CreateKeysAndCertificate` (p. 753)Pour créer un nouveau certificat et une clé privée à l'aide de l'adresseAWSAutorité de certification.

   Ou

   b. Appelez `CreateCertificateFromCsr` (p. 752) pour générer un certificat à partir d'une demande de signature de certificat qui conserve sa clé privée sécurisée.

3. À partir de l'appareil, appelez `RegisterThing` (p. 755) pour enregistrer ce dernier auprès d'AWS IoT et créer des ressources cloud.

   Le service de mise en service de flotte crée des ressources cloud telles que des objets, des groupes d'objets et des attributs, tels que définis dans le modèle de mise en service.

4. Après avoir enregistré le certificat permanent sur l'appareil, celui-ci doit se déconnecter de la session qu'il a ouverte avec le certificat de revendication de mise en service et se reconnecter à l'aide du certificat permanent.

L'appareil est maintenant prêt à communiquer normalement avec AWS IoT.

Allocation par utilisateur approuvé

Dans de nombreux cas, un appareil se connecte à AWS IoT pour la première fois lorsqu'un utilisateur approuvé, comme un utilisateur final ou un technicien d'installation utilise une application mobile pour configurer l'appareil dans à l'endroit où il est déployé.

Important

Vous devez gérer l'accès et l'autorisation de l'utilisateur approuvé pour effectuer cette procédure. Une façon de le faire consiste à fournir et à gérer un compte pour l'utilisateur approuvé qui l'authentifie et lui accorde l'accès aux fonctionnalités d'AWS IoT et aux API requises pour effectuer cette procédure.

Avant de livrer l'appareil

1. Appelez `CreateProvisioningTemplate` pour créer un modèle de mise en service et renvoyer son *`TemplateArn`* et *`TemplateName`*.
2. Créez un rôle IAM qui permettra à un utilisateur de confiance de lancer le processus de mise en service. Le modèle de mise en service permet uniquement à cet utilisateur de mettre en service un appareil. Exemples :

```
{
    "Effect": "Allow",
    "Action": [
        "iot:CreateProvisioningClaim",
    ],
    "Resource": [
        "arn:aws:aws-region:aws-account-id:provisioningtemplate/templateName"
```

```
    ]
}
```

3. Accordez au service AWS IoT l'autorisation de créer ou de mettre à jour des ressources IoT telles que des objets et des certificats dans votre compte lors de la mise en service des appareils. Pour cela, vous attachez l'adresse`AWSIoTThingsRegistration`Stratégie gérée par un rôle IAM (appeléérôle de mise en service) qui fait confiance à laAWS IoTPrincipal du service.
4. Fournit les moyens d'identifier vos utilisateurs de confiance, notamment en leur fournissant un compte qui peut les authentifier et autoriser leurs interactions avec laAWSAPI nécessaires pour enregistrer leurs appareils.

Pour initialiser l’appareil à utiliser

1. Un utilisateur de confiance se connecte à votre application mobile ou service web de mise en service.
2. L'application mobile ou l'application web utilise le rôle IAM et appelle `CreateProvisioningClaim`pour obtenir un certificat de revendication de provisionnement temporaire auprès deAWS IoT.

   > **Note**
   >
   > Pour des raisons de sécurité, le certificat de revendication de mise en service temporaire qui`CreateProvisioningClaim`L'adresse renvoyée expire après cinq minutes. Les étapes suivantes doivent renvoyer un certificat valide avant l'expiration du certificat de revendication de mise en service temporaire. Les certificats de revendication de mise en service temporaire n'apparaissent pas dans la liste des certificats de votre compte.
3. L'application mobile ou l'application web fournit le certificat de revendication de mise en service à l'appareil ainsi que toutes les informations de configuration nécessaires, notamment les informations d'identification Wi-Fi.
4. L'appareil utilise le certificat de revendication de mise en service temporaire pour se connecter à AWS IoT à l'aide du AWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil (p. 1140).
5. L'appareil obtient un certificat permanent et une clé privée en utilisant l'une de ces options. L’appareil utilisera le certificat et la clé pour toute authentification future auprès d’AWS IoT.

   a. Appelez `CreateKeysAndCertificate` (p. 753)Pour créer un nouveau certificat et une clé privée à l'aide de l'adresseAWSAutorité de certification.

      Ou

   b. Appelez `CreateCertificateFromCsr` (p. 752) pour générer un certificat à partir d'une demande de signature de certificat qui conserve sa clé privée sécurisée.
6. L'appareil appelle `RegisterThing` (p. 755) pour enregistrer l'appareil auprès d'AWS IoT et créer des ressources cloud.

   > **Note**
   >
   > Gardez à l'esprit ce qui suit`RegisterThing`doit retourner un certificat valide dans les cinq minutes suivant la connexion àAWS IoTavec le certificat de revendication de mise en service temporaire.

   Le service de mise en service de flotte crée des ressources cloud telles que des objets IoT, des groupes d'objets et des attributs, tels que définis dans le modèle de mise en service.
7. Après avoir enregistré le certificat permanent sur l’appareil, celui-ci doit se déconnecter de la session qu'il a ouverte avec le certificat de revendication de mise en service temporaire et se reconnecter à l'aide du certificat permanent.

L'appareil est maintenant prêt à communiquer normalement avec AWS IoT.

Utilisation des hooks de pré-provisionnement avec l'interface de ligne de commande AWS

La procédure suivante crée un modèle de mise en service avec des hooks de mise en service en amont. La fonction Lambda utilisée ici est un exemple qui peut être modifié.

Pour créer et appliquer un hook de mise en service en amont à un modèle de mise en service

1. Créez une fonction Lambda qui a une entrée et une sortie définies. Les fonctions Lambda sont hautement personnalisables`allowProvisioning`and`parameterOverrides`sont nécessaires pour créer des hooks de mise en service en amont. Pour plus d'informations sur la création de fonctions Lambda, consultezUtiliserAWS Lambdaavec leAWSInterface de ligne de commande.

   Voici un exemple de sortie de fonction Lambda :

   ```
   {
     "allowProvisioning": True,
     "parameterOverrides": {
       "incomingKey0": "incomingValue0",
       "incomingKey1": "incomingValue1"
     }
   }
   ```

2. AWS IoTutilise des stratégies basées sur les ressources pour appeler Lambda, vous devez donc donnerAWS IoTAutorisation d'appeler votre fonction Lambda

   Voici un exemple de commande deautorisation d'extensiondonnez l'autorisation IoT à votre Lambda.

   ```
   aws lambda add-permission \
       --function-name myLambdaFunction \
       --statement-id iot-permission \
       --action lambda:InvokeFunction \
       --principal iot.amazonaws.com
   ```

3. Ajoutez un hook de mise en service en amont à un modèle à l'aide de la commande create-provisioning-template ou update-provisioning-template.

   L'exemple de commande CLI suivant utilise le modèle create-provisioning-template pour créer un modèle de mise en service avec des hooks de mise en service en amont :

   ```
   aws iot create-provisioning template \
       --template-name myTemplate \
       --provisioning-role-arn arn:aws:iam:us-east-1:1234564789012:role/myRole \
       --template-body file://template.json \
       --pre-provisioning-hook file://hooks.json
   ```

   La sortie de cette commande ressemble à ce qui suit :

   ```
   {
       "templateArn": "arn:aws:iot:us-east-1:1234564789012:provisioningtemplate/
   myTemplate",
       "defaultVersionId": 1,
       "templateName": myTemplate
   }
   ```

   Pour gagner du temps, vous pouvez également charger un paramètre à partir d'un fichier au lieu de le taper en tant que valeur de paramètre de ligne de commande. Pour de plus amples informations,

veuillez consulterchargement en coursAWS CLIParamètres d'à partir d'un fichier. Le code suivant illustre le paramètre `template` au format JSON étendu :

```
{
    "Parameters" : {
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingTypeName" : {"Fn::Join":["",["ThingTypePrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["widgets", "WA"],
                "BillingGroup": "BillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:504350838278:topic/foo/
bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"},
 "LocationUrl"]}
    }
```

```
}
```

Le code suivant illustre le paramètre `pre-provisioning-hook` au format JSON étendu :

```
{
     "targetArn" : "arn:aws:lambda:us-
east-1:765219403047:function:pre_provisioning_test",
     "payloadVersion" : "2020-04-01"
}
```

Mise en service d'appareils disposant de certificats d'appareils

AWS IoT propose trois méthodes pour mettre en service les appareils lorsqu'ils disposent déjà d'un certificat d'appareil (et d'une clé privée associée) :

- Mise en service d'un objet unique avec un modèle de mise en service. Cette option convient si vous devez uniquement mettre en service des appareils un par un.
- Mise en service juste-à-temps (Just-in-time provisioning, JITP) avec un modèle qui met en service un appareil lors de sa première connexion à AWS IoT. Cette option convient si vous devez enregistrer un grand nombre d'appareils, mais que vous n'avez pas d'informations sur ceux-ci que vous pouvez assembler dans une liste de mise en service en bloc.
- Enregistrement en bloc. Cette option vous permet de spécifier une liste de valeurs de modèle de mise en service d'objet unique qui sont stockées dans un fichier au sein d'un compartiment S3. Cette approche fonctionne bien si vous avez un grand nombre d'appareils connus dont vous pouvez assembler les caractéristiques souhaitées dans une liste.

Rubriques

Mise en service d'un seul objet

Pour mettre en service un objet, utilisez l'API RegisterThing ou la commande de l'interface de ligne de commande `register-thing`. La commande de l'interface de ligne de commande `register-thing` accepte les arguments suivants :

--template-body

Modèle de mise en service.

--parameters

Liste de paires nom-valeur pour les paramètres utilisés dans le modèle de mise en service, au format JSON (par exemple, `{"ThingName" : "MyProvisionedThing", "CSR" : "csr-text"}`).

Voir Mise en service des modèles (p. 740).

RegisterThing ou `register-thing` renvoie les ARN des ressources et le texte du certificat créé :

```
{
    "certificatePem": "certificate-text",
    "resourceArns": {
    "PolicyLogicalName": "arn:aws:iot:us-
west-2:123456789012:policy/2A6577675B7CD1823E271C7AAD8184F44630FFD7",
    "certificate": "arn:aws:iot:us-west-2:123456789012:cert/
cd82bb924d4c6ccbb14986dcb4f40f30d892cc6b3ce7ad5008ed6542eea2b049",
    "thing": "arn:aws:iot:us-west-2:123456789012:thing/MyProvisionedThing"
    }
}
```

Si un paramètre est omis du dictionnaire, la valeur par défaut est utilisée. Si aucune valeur par défaut n'est spécifiée, le paramètre n'est pas remplacé par une valeur.

Mise en service juste-à-temps

Vous pouvez demander que les appareils soient mis en service lors de leur première tentative de connexion à AWS IoT. Les paramètres de la mise en service juste-à-temps (JITP) sont définis sur les certificats CA. Pour mettre en service l'appareil, vous devez activer l'enregistrement automatique et associer un modèle de mise en service au certificat CA utilisé pour signer le certificat d'appareil. Les succès et les erreurs de provisionnement sont consignés en tant queMétriques de mise en service d'appareils (p. 371)dans Amazon CloudWatch.

Vous pouvez définir ces paramètres lorsque vous enregistrez un nouveau certificat de CA avec l'API RegisterCACertificate ou la commande de l'interface de ligne de commande `register-ca-certificate` :

```
aws iot register-ca-certificate --ca-certificate file://your-ca-cert --verification-cert
                file://your-verification-cert --set-as-active --allow-auto-registration --
registration-config file://your-template
```

Pour en savoir plus, consultez Enregistrement d'un certificat CA.

Vous pouvez également utiliser l'API UpdateCACertificate ou la commande de l'interface de ligne de commande `update-ca-certificate` afin de mettre à jour les paramètres d'un certificat CA :

```
aws iot update-ca-certificate --certificate-id caCertificateId --new-auto-registration-
status ENABLE --registration-config file://your-template
```

> **Note**
>
> JITP appelle d'autres API du plan de contrôle AWS IoT pendant le processus de mise en service. Ces appels peuvent dépasser les quotas de limitation AWS IoT définis pour votre compte et entraîner des appels limités. ContactAWSsupport clientPour augmenter vos quotas de limitation, si nécessaire.

Lorsqu'un appareil tente de se connecter à AWS IoT en utilisant un certificat signé par un certificat CA enregistré, AWS IoT charge le modèle à partir du certificat CA et l'utilise pour appeler RegisterThing (p. 755). Le flux de travail de mise en service JITP enregistre d'abord un certificat avec une valeur de statut PENDING_ACTIVATION. Lorsque le flux de mise en service d'appareil est terminé, le statut du certificat est modifié en ACTIVE.

AWS IoT définit les paramètres suivants que vous pouvez déclarer et référencer dans les modèles de mise en service :

- `AWS::IoT::Certificate::Country`
- `AWS::IoT::Certificate::Organization`

- `AWS::IoT::Certificate::OrganizationalUnit`
- `AWS::IoT::Certificate::DistinguishedNameQualifier`
- `AWS::IoT::Certificate::StateName`
- `AWS::IoT::Certificate::CommonName`
- `AWS::IoT::Certificate::SerialNumber`
- `AWS::IoT::Certificate::Id`

Les valeurs de ces paramètres de modèle de mise en service sont limitées à ce que la mise en service JITP peut extraire du champ d'objet du certificat de l'appareil qui est mis en service. Le certificat doit contenir des valeurs pour tous les paramètres du corps du modèle. Le paramètre `AWS::IoT::Certificate::Id` fait référence à un ID généré en interne, et non un ID qui est contenu dans le certificat. Vous pouvez obtenir la valeur de cet ID à l'aide de la fonction `principal()` à l'intérieur d'une règle AWS IoT.

Le fichier JSON suivant est l'exemple d'un modèle de mise en service JITP complet. La valeur du champ `templateBody` doit être un objet JSON spécifié comme une chaîne placée dans une séquence d'échappement et peut utiliser uniquement les valeurs de la liste précédente. Vous pouvez utiliser différents outils pour créer l'objet JSON requis, par exemple `json.dumps` (Python) ou `JSON.stringify` (Node). La valeur du champ `roleARN` doit être l'ARN d'un rôle qui est le `AWSIoTThingsRegistration` attaché à celui-ci. De plus, votre modèle peut utiliser un `PolicyName` au lieu du `PolicyDocument` en ligne dans l'exemple. (Des sauts de ligne sont ajoutés au premier exemple pour plus de lisibilité, mais vous pouvez copier et coller le modèle qui s'affiche directement ci-dessous.)

```
       {
           "templateBody" : "{
               \r\n    \"Parameters\" : {\r\n
                           \"AWS::IoT::Certificate::CommonName\": {\r\n    \"Type\":
 \"String\"\r\n        },\r\n
                           \"AWS::IoT::Certificate::SerialNumber\": {\r\n  \"Type\":
 \"String\"\r\n        },\r\n
                           \"AWS::IoT::Certificate::Country\": {\r\n   \"Type\": \"String
\"\r\n        },\r\n
                           \"AWS::IoT::Certificate::Id\": {\r\n    \"Type\": \"String\"\r
\n        }\r\n    },\r\n
                      \"Resources\": {\r\n
                           \"thing\": {\r\n
                               \"Type\": \"AWS::IoT::Thing\",\r\n
                               \"Properties\": {\r\n
                                   \"ThingName\": {\r\n    \"Ref\":
 \"AWS::IoT::Certificate::CommonName\"\r\n  },\r\n
                                   \"AttributePayload\": {\r\n
                                       \"version\": \"v1\",\r\n
                                       \"serialNumber\": {\r\n
                                           \"Ref\": \"AWS::IoT::Certificate::SerialNumber
\"\r\n    }\r\n   },\r\n
                                   \"ThingTypeName\": \"lightBulb-versionA\",\r\n

                                   \"ThingGroups\": [\r\n
                                       \"v1-lightbulbs\",\r\n  {\r\n

                                           \"Ref\": \"AWS::IoT::Certificate::Country\"\r
\n }\r\n   ]\r\n   },\r\n
                           \"OverrideSettings\": {\r\n
                               \"AttributePayload\": \"MERGE\",\r\n
                               \"ThingTypeName\": \"REPLACE\",\r\n
                               \"ThingGroups\": \"DO_NOTHING\"\r\n }\r\n   },\r\n
                       \"certificate\": {\r\n
                           \"Type\": \"AWS::IoT::Certificate\",\r\n
                           \"Properties\": {\r\n
```

```
                                \"CertificateId\": {\r\n    \"Ref\":
 \"AWS::IoT::Certificate::Id\"\r\n  },\r\n
                                \"Status\": \"ACTIVE\"\r\n  },\r\n
                            \"OverrideSettings\": {\r\n
                                \"Status\": \"DO_NOTHING\"\r\n  }\r\n   },\r\n
                        \"policy\": {\r\n
                            \"Type\": \"AWS::IoT::Policy\",\r\n
                            \"Properties\": {\r\n
                                \"PolicyDocument\": \"{
                                    \\\"Version\\\": \\\"2012-10-17\\\",
                                    \\\"Statement\\\": [{
                                    \\\"Effect\\\": \\\"Allow\\\",
                                    \\\"Action\\\":[\\\"iot:Publish\\\"],
                                    \\\"Resource\\\": [\\\"arn:aws:iot:us-
east-1:123456789012:topic\/sample\/topic\\\"] }] }\"\r\n   }\r\n   }\r\n
                            }\r\n}",
            "roleArn" : "arn:aws:iam::123456789012:role/Provisioning-JITP"
        }
```

Voici une version que vous pouvez copier et coller :

```
    {
      "templateBody" : "{\r\n    \"Parameters\" : {\r\n
 \"AWS::IoT::Certificate::CommonName\": {\r\n            \"Type\": \"String\"\r\n        },
\r\n        \"AWS::IoT::Certificate::SerialNumber\": {\r\n            \"Type\": \"String
\"\r\n        },\r\n        \"AWS::IoT::Certificate::Country\": {\r\n            \"Type\":
 \"String\"\r\n        },\r\n        \"AWS::IoT::Certificate::Id\": {\r\n            \"Type
\": \"String\"\r\n        }\r\n    },\r\n    \"Resources\": {\r\n        \"thing\": {\r\n
            \"Type\": \"AWS::IoT::Thing\",\r\n            \"Properties\": {\r\n
      \"ThingName\": {\r\n                    \"Ref\": \"AWS::IoT::Certificate::CommonName
\"\r\n                },\r\n                \"AttributePayload\": {\r\n
   \"version\": \"v1\",\r\n                    \"serialNumber\": {\r\n
      \"Ref\": \"AWS::IoT::Certificate::SerialNumber\"\r\n                    }\r\n
          },\r\n                \"ThingTypeName\": \"lightBulb-versionA\",\r\n
      \"ThingGroups\": [\r\n                    \"v1-lightbulbs\",\r\n
 {\r\n                        \"Ref\": \"AWS::IoT::Certificate::Country\"\r\n
        }\r\n                ]\r\n            },\r\n            \"OverrideSettings\": {\r
\n                \"AttributePayload\": \"MERGE\",\r\n                \"ThingTypeName\":
 \"REPLACE\",\r\n                \"ThingGroups\": \"DO_NOTHING\"\r\n            }\r\n
   },\r\n        \"certificate\": {\r\n            \"Type\": \"AWS::IoT::Certificate\",\r\n
            \"Properties\": {\r\n                \"CertificateId\": {\r\n
   \"Ref\": \"AWS::IoT::Certificate::Id\"\r\n                },\r\n                \"Status
\": \"ACTIVE\"\r\n            },\r\n            \"OverrideSettings\": {\r\n
  \"Status\": \"DO_NOTHING\"\r\n            }\r\n        },\r\n        \"policy\": {\r\n
          \"Type\": \"AWS::IoT::Policy\",\r\n            \"Properties\": {\r\n
      \"PolicyDocument\": \"{ \\\"Version\\\": \\\"2012-10-17\\\", \\\"Statement\\\": [{ \
\\\"Effect\\\": \\\"Allow\\\", \\\"Action\\\":[\\\"iot:Publish\\\"], \\\"Resource\\\": [\\
\"arn:aws:iot:us-east-1:123456789012:topic\/foo\/bar\\\"] }] }\"\r\n            }\r\n
  }\r\n    }\r\n}",
      "roleArn" : "arn:aws:iam::123456789012:role/JITPRole"
    }
```

Cet exemple de modèle déclare des valeurs pour les paramètres de mise en service `AWS::IoT::Certificate::CommonName`, `AWS::IoT::Certificate::SerialNumber`, `AWS::IoT::Certificate::Country` et `AWS::IoT::Certificate::Id` qui sont extraits du certificat et utilisés dans la section `Resources`. Le flux de travail JITP utilise ensuite ce modèle pour effectuer les actions suivantes :

- Enregistrer un certificat et définir son état sur PENDING_ACTIVE.
- Créer une ressource d'objet.

- Créer une ressource de stratégie.
- Attacher la stratégie au certificat.
- Attacher le certificat à l'objet.
- Mettre à jour le statut du certificat en ACTIVE.

Notez que le provisionnement des périphériques échoue si le certificat ne possède pas toutes les propriétés mentionnées dans la`Parameters`Section du`templateBody`. Par exemple, si`AWS::IoT::Certificate::Country`est inclus dans le modèle, mais le certificat n'a pas de`Country`, le provisioning des périphériques échoue.

Vous pouvez également utiliser CloudTrail pour résoudre les problèmes liés à votre modèle JITP. Pour plus d'informations sur les mesures enregistrées dans Amazon CloudWatch, consultezMétriques de mise en service d'appareils (p. 371).

Enregistrement en bloc

Vous pouvez utiliser la commande `start-thing-registration-task` pour enregistrer les objets en bloc. Cette commande utilise un modèle d'enregistrement, un nom de compartiment S3, un nom de clé et un ARN de rôle qui autorise l'accès au fichier dans le compartiment S3. Le fichier du compartiment S3 contient les valeurs utilisées pour remplacer les paramètres dans le modèle. Le fichier doit être au format JSON délimité par une nouvelle ligne. Chaque ligne contient toutes les valeurs des paramètres pour l'enregistrement d'un seul appareil. Exemples :

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}
{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

Les API suivantes liées à l'enregistrement en bloc peuvent être utiles :

- ListThingRegistrationTasks : Liste les tâches de mise en service des objets en bloc qui sont en cours
- DescribeThingRegistrationTask : Fournit des informations sur une tâche spécifique d'enregistrement des objets en bloc.
- StopThingRegistrationTask : Arrête une tâche d'enregistrement d'objets en bloc.
- ListThingRegistrationTaskReports : Utilisé pour vérifier les résultats et/ou les défaillances d'une tâche d'enregistrement d'objets en bloc.

Note

- Vous ne pouvez exécuter qu'une seule tâche d'enregistrement en bloc à la fois (par compte).
- Les opérations d'enregistrement en bloc appellent d'autres API de plan de contrôle AWS IoT. Ces appels peuvent dépasser les quotas de limitation AWS IoT définis dans votre compte et provoquer des erreurs de limitation. ContactAWSsupport clientpour augmenter votreAWS IoTQuotons de limitation, si nécessaire.

Mise en service des modèles

Un modèle de mise en service est un document JSON qui utilise des paramètres pour décrire les ressources que votre appareil doit utiliser afin d'interagir avec AWS IoT. Un modèle contient deux sections : `Parameters` et `Resources`. Il existe deux types de modèles de mise en service dans AWS IoT. L'un est utilisé pour le provisionnement juste-à-temps (JITP) et l'enregistrement en bloc, et le second est utilisé pour l'allocation de parc.

Section Parameters

La section `Parameters` déclare les paramètres utilisés dans la section `Resources`. Chaque paramètre déclare un nom, un type et une valeur facultative par défaut. La valeur par défaut est utilisée lorsque le dictionnaire transmis avec le modèle ne contient pas de valeur pour le paramètre. La section `Parameters` d'un modèle de document ressemble à ce qui suit :

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CSR" : {
            "Type" : "String"
        }
    }
}
```

Cet extrait de modèle déclare quatre paramètres : `ThingName`, `SerialNumber`, `Location` et `CSR`. Tous ces paramètres sont de type `String`. Le paramètre `Location` déclare la valeur par défaut `"WA"`.

Section Resources

La section `Resources` du modèle déclare les ressources dont votre appareil a besoin pour pouvoir communiquer avec AWS IoT : un objet, un certificat et une ou plusieurs stratégies IoT. Chaque ressource spécifie un nom logique, un type et un ensemble de propriétés.

Un nom logique vous permet de faire référence à une ressource à un autre endroit dans le modèle.

Le type spécifie le type de ressource que vous déclarez. Les types valides sont :

- `AWS::IoT::Thing`
- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`

Les propriétés que vous spécifiez dépendent du type de ressource que vous déclarez.

Ressources d'objet

Les ressources d'objet sont déclarées à l'aide des propriétés suivantes :

- `ThingName`: String.
- `AttributePayload` - Facultatif. Liste de paires nom-valeur.
- `ThingTypeName` - Facultatif. Chaîne d'un type d'objet associé pour l'objet.
- `ThingGroups` - Facultatif. Liste des groupes auxquels l'objet appartient.

Ressources de certificat

Les certificats peuvent être spécifiés de l'une des façons suivantes :

- Demande de signature du certificat (CSR).
- ID d'un certificat d'appareil existant. (Seuls les ID de certificat peuvent être utilisés avec un modèle d'allocation de parc.)
- Certificat d'appareil créé avec un certificat CA enregistré auprès d'AWS IoT. Si vous avez plusieurs certificats CA enregistrés avec le même champ d'objet, vous devez également transmettre le certificat CA utilisé pour signer le certificat de l'appareil.

> **Note**
>
> Lorsque vous déclarez un certificat dans un modèle, vous devez uniquement utiliser ces méthodes. Par exemple, si vous utilisez une CSR, vous ne pouvez pas spécifier d'ID de certificat ou de certificat d'appareil. Pour plus d'informations, consultez Certificats client X.509 (p. 236).

Pour plus d'informations, consultez Présentation des certificats X.509 (p. 233).

Les ressources de certificat sont déclarées à l'aide des propriétés suivantes :

- `CertificateSigningRequest`: String.
- `CertificateID`: String.
- `CertificatePem`: String.
- `CACertificatePem`: String.
- `Status` - Facultatif. Chaîne qui peut être `ACTIVE` ou `INACTIVE`. ACTIVE est l'option sélectionnée par défaut.

Exemples :

- Certificat spécifié avec une CSR :

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CertificateSigningRequest": {"Ref" : "CSR"},
            "Status" : "ACTIVE"
        }
    }
}
```

- Certificat spécifié avec un ID de certificat existant :

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CertificateId": {"Ref" : "CertificateId"}
        }
    }
}
```

- Certificat spécifié avec un fichier .pem de certificat et un fichier .pem de certificat de CA existants :

```
{
    "certificate" : {
        "Type" : "AWS::IoT::Certificate",
        "Properties" : {
            "CACertificatePem": {"Ref" : "CACertificatePem"},
            "CertificatePem": {"Ref" : "CertificatePem"}
        }
```

```
    }
}
```

Ressources de stratégie

Les ressources de stratégie sont déclarées à l'aide de l'une des propriétés suivantes :

- `PolicyName` - Facultatif. String. Un hachage du document de stratégie est utilisé par défaut. Si vous utilisez une stratégie AWS IoT existante, pour la propriété `PolicyName`, saisissez le nom de la stratégie. N'incluez pas la propriété `PolicyDocument`.
- `PolicyDocument` - Facultatif. Un objet JSON spécifié comme une chaîne placée dans une séquence d'échappement. Si `PolicyDocument` n'est pas fourni, la stratégie doit déjà être créée.

> **Note**
>
> Si une section `Policy` est présente, `PolicyName` ou `PolicyDocument`, mais pas les deux, doit être spécifié.

Remplacer les paramètres

Si un modèle spécifie une ressource qui existe déjà, la section `OverrideSettings` vous permet de spécifier l'action à effectuer :

`DO_NOTHING`

Conserver la ressource telle qu'elle est.

`REPLACE`

Remplacer la ressource par celle qui est spécifiée dans le modèle.

`FAIL`

Entraîner l'échec de la demande avec `ResourceConflictsException`.

`MERGE`

Valide uniquement pour les propriétés `ThingGroups` et `AttributePayload` d'un `thing`. Fusionnez les attributs existants ou les appartenances aux groupes de l'objet avec ceux spécifiés dans le modèle.

Lorsque vous déclarez une ressource d'objet, vous pouvez spécifier `OverrideSettings` pour les propriétés suivantes :

- `ATTRIBUTE_PAYLOAD`
- `THING_TYPE_NAME`
- `THING_GROUPS`

Lorsque vous déclarez une ressource de certificat, vous pouvez spécifier `OverrideSettings` pour la propriété `Status`.

`OverrideSettings` n'est pas disponible pour les ressources de stratégie.

Exemple de ressource

L'extrait de modèle suivant déclare un objet, un certificat et une stratégie :

```
{
```

```
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateSigningRequest": {"Ref" : "CSR"},
                "Status" : "ACTIVE"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

L'objet est déclaré avec :

- Le nom logique `"thing"`.
- Le type `AWS::IoT::Thing`.
- Un ensemble de propriétés d'objet.

  Les propriétés d'objet comprennent le nom de l'objet, un ensemble d'attributs, un nom de type d'objet facultatif et une liste facultative de groupes d'objets auxquels l'objet appartient.

Les paramètres sont référencés par `{"Ref":"`*parameter-name*`"}`. Lorsque le modèle est évalué, les paramètres sont remplacés par la valeur du paramètre à partir du dictionnaire transmis avec le modèle.

Le certificat est déclaré avec :

- Le nom logique `"certificate"`.
- Le type `AWS::IoT::Certificate`.
- Un ensemble de propriétés.

  Les propriétés incluent la CSR pour le certificat et la définition de l'état sur `ACTIVE`. Le texte de la CSR est transmis en tant que paramètre dans le dictionnaire transmis avec le modèle.

La stratégie est déclarée avec :

- Le nom logique `"policy"`.
- Le type `AWS::IoT::Policy`.

- Le nom d'une stratégie existante ou un document de stratégie.

Exemple de modèle pour JITP et l'enregistrement en bloc

Le fichier JSON suivant est un exemple de modèle de mise en service complet qui spécifie le certificat avec une CSR :

(La valeur du champ `PolicyDocument` doit être un objet JSON spécifié comme une chaîne placée dans une séquence d'échappement.)

```
{
    "Parameters" : {
        "ThingName " : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CSR" : {
            "Type" : "String"
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateSigningRequest": {"Ref" : "CSR"},
                "Status" : "ACTIVE"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

Le fichier JSON suivant est un exemple de modèle de mise en service complet qui spécifie un certificat existant avec un ID de certificat :

```
{
```

```
    "Parameters" : {
        "ThingName " : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CertificateId" : {
            "Type" : "String"
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" :  {"Ref" :
 "SerialNumber"}},
                "ThingTypeName" :  "lightBulb-versionA",
                "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref" : "CertificateId"}
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\":
 [{ \"Effect\": \"Allow\", \"Action\":[\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-
east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

Mise en service de flotte

Les modèles de mise en service de flotte sont utilisés par AWS IoT pour définir la configuration du cloud et des appareils. Ces modèles utilisent les mêmes paramètres et ressources que le JITP et les modèles d'enregistrement en bloc. Pour plus d'informations, consultez Mise en service des modèles (p. 740).
Les modèles de mise en service de flotte peuvent contenir une section `Mapping` et une section `DeviceConfiguration`. Vous pouvez utiliser des fonctions intrinsèques à l'intérieur d'un modèle de mise en service de flotte pour générer une configuration spécifique à l'appareil. Les modèles d'allocation de parc sont des ressources nommées et sont identifiés par des noms ARN (par exemple, `arn:aws:iot:us-west-2:1234568788:provisioningtemplate/templateName`).

Mappings

La section `Mappings` facultative associe à une clé à un ensemble de valeurs correspondantes portant un nom. Par exemple, si vous souhaitez définir des valeurs basées sur unAWS, vous pouvez créer un mappage qui utilise la Région AWS en tant que clé et contient les valeurs que vous souhaitez spécifier pour chaque région spécifique. Pour récupérer les valeurs d'un mappage, utilisez la fonction intrinsèque `Fn::FindInMap`.

Vous ne pouvez pas inclure des paramètres, des pseudo-paramètres ou des fonctions d'appel intrinsèques dans la section `Mappings`.

Configuration de l'appareil

La section Configuration des appareils contient des données arbitraires que vous souhaitez envoyer à vos appareils lors de la mise en service. Exemples :

```
{
    "DeviceConfiguration": {
        "Foo":"Bar"
    }
}
```

Si vous envoyez des messages à vos appareils à l'aide du format JSON (JavaScript Object Notation), AWS IoT Core met en forme ces données en JSON. Si vous utilisez le format de charge utile CBOR (Concise Binary Object Representation, représentation concise d'objets binaires), AWS IoT Core formate ces données en tant que CBOR. La `.DeviceConfiguration`ne prend pas en charge les objets JSON imbriqués.

Fonctions intrinsèques

Les fonctions intrinsèques sont utilisées dans n'importe quelle section du modèle de mise en service, à l'exception de la section `Mappings`.

`Fn::Join`

Ajoute un ensemble de valeurs dans une seule valeur, séparées par le délimiteur spécifié. Si un délimiteur est une chaîne vide, l'ensemble de valeurs est concaténé avec aucun délimiteur.

`Fn::Select`

Renvoie un seul objet à partir d'une liste d'objets en fonction de son index.

> **Important**
>
> `Fn::Select` ne recherche pas les valeurs `null` ou ne vérifie pas si l'index sort des limites du tableau. Les deux conditions entraînent une erreur de mise en service. Vous devez donc vérifier que vous avez choisi une valeur d'index valide et que la liste contient des valeurs non nulles.

`Fn::FindInMap`

Renvoie la valeur correspondant aux clés dans un mappage à deux niveaux déclaré dans la section `Mappings`.

`Fn::Split`

Divise une chaîne en liste de valeurs de chaîne afin que vous puissiez sélectionner un élément dans la liste de chaînes. Vous spécifiez un délimiteur qui détermine où la chaîne est fractionnée (par exemple, une virgule). Après avoir divisé une chaîne, utilisez `Fn::Select` pour sélectionner un élément.

Par exemple, si une chaîne d'ID de sous-réseaux délimités par des virgules est importée dans votre modèle de pile, vous pouvez la fractionner au niveau de chaque virgule. Dans la liste des ID de sous-réseau, utilisez `Fn::Select` pour spécifier un ID de sous-réseau pour une ressource.

`Fn::Sub`

Remplace les variables contenues dans une chaîne d'entrée par des valeurs que vous spécifiez. Vous pouvez utiliser cette fonction pour construire des commandes ou des sorties qui incluent des valeurs qui ne sont pas disponibles tant que vous n'avez pas créé ou mis à jour une pile.

Exemple de modèle de mise en service de flotte

```
{
    "Parameters" : {
        "ThingName " : {
            "Type" : "String"
        },
        "SerialNumber": {
            "Type": "String"
        },
        "DeviceLocation": {
            "Type": "String"
        }
    },
    "Mappings": {
        "LocationTable": {
            "Seattle": {
                "LocationUrl": "https://example.aws"
            }
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "AttributePayload" : {
                    "version" : "v1",
                    "serialNumber" : "serialNumber"
                },
                "ThingName" : {"Ref" : "ThingName"},
                "ThingTypeName" : {"Fn::Join":["",["ThingPrefix_",
{"Ref":"SerialNumber"}]]},
                "ThingGroups" : ["v1-lightbulbs", "WA"],
                "BillingGroup": "LightBulbBillingGroup"
            },
            "OverrideSettings" : {
                "AttributePayload" : "MERGE",
                "ThingTypeName" : "REPLACE",
                "ThingGroups" : "DO_NOTHING"
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateId": {"Ref": "AWS::IoT::Certificate::Id"},
                "Status" : "Active"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : {
                    "Version": "2012-10-17",
                    "Statement": [{
                        "Effect": "Allow",
                        "Action":["iot:Publish"],
                        "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/foo/bar"]
                    }]
                }
            }
        }
    },
    "DeviceConfiguration": {
        "FallbackUrl": "https://www.example.com/test-site",
```

```
        "LocationUrl": {
            "Fn::FindInMap": ["LocationTable",{"Ref": "DeviceLocation"}, "LocationUrl"]}
        }
}
```

Note

Un modèle de mise en service existant peut être mis à jour afin d'ajouter un hook de mise en service en amont (p. 749).

Hooks de mise en service en amont

Lorsque vous utilisezAWS IoT, vous pouvez configurer une fonction Lambda pour valider les paramètres de l'appareil avant d'autoriser la mise en service de ce dernier. Cette fonction Lambda doit exister dans votre compte avant la mise en service d'un appareil, car elle est appelée chaque fois qu'un appareil envoie une demande viathe section called "RegisterThing" (p. 755). Pour que les appareils soient provisionnés, votre fonction Lambda doit accepter l'objet d'entrée et renvoyer l'objet de sortie décrit dans cette section. La mise en service ne se déroule que si la fonction Lambda renvoie un objet avec la valeur`"allowProvisioning": True`.

Important

AWS recommande d'utiliser des hooks de mise en service en amont lors de la création de modèles de mise en service pour permettre de contrôler davantage le nombre et l'identité des appareils intégrés à votre compte.

Entrée du hook de pré-provisionnement

AWS IoTenvoie cet objet à la fonction Lambda quand un périphérique s'enregistre avecAWS IoT.

```
{
    "claimCertificateId" : "string",
    "certificateId" : "string",
    "certificatePem" : "string",
    "templateArn" : "arn:aws:iot:us-east-1:1234567890:provisioningtemplate/MyTemplate",
    "clientId" : "221a6d10-9c7f-42f1-9153-e52e6fc869c1",
    "parameters" : {
        "string" : "string",
        ...
    }
}
```

La .`parameters`transmis à la fonction Lambda contient les propriétés de la propriété`parameters`passé dans l'argumentthe section called "RegisterThing" (p. 755)Charge utile de la demande.

Valeur de retour du hook de pré-provisionnement

La fonction Lambda doit renvoyer une réponse indiquant si elle a autorisé la demande de provisionnement et les valeurs de toutes les propriétés à remplacer.

Voici un exemple de réponse réussie de la fonction de pré-provisionnement.

```
{
    "allowProvisioning": true,
    "parameterOverrides" : {
        "Key": "newCustomValue",
```

```
        ...
    }
}
```

`"parameterOverrides"`Les valeurs seront ajoutées à`"parameters"`paramètre de l'the section called “RegisterThing” (p. 755)Charge utile de la demande.

Note

- Si la fonction Lambda échoue ou ne renvoie pas le`"allowProvisioning"`dans la réponse, la demande de mise en service échoue et l'erreur est renvoyée dans la réponse.
- La fonction Lambda doit procéder à l'exécution et renvoyer une réponse dans un délai de 5 secondes, sinon la demande de mise en service échoue.

Exemple de crochet de mise en service en amont Lambda

Python

Exemple de hook de mise en service en amont Lambda en Python.

```
import json

def pre_provisioning_hook(event, context):
    print(event)

    return {
        'allowProvisioning': True,
        'parameterOverrides': {
            'DeviceLocation': 'Seattle'
        }
    }
```

Java

Exemple de hook de mise en service en amont Lambda en Java.

Classe du gestionnaire :

```
package example;

import java.util.Map;
import java.util.HashMap;
import com.amazonaws.services.lambda.runtime.Context;
import com.amazonaws.services.lambda.runtime.RequestHandler;

public class PreProvisioningHook implements RequestHandler<PreProvisioningHookRequest,
 PreProvisioningHookResponse> {

    public PreProvisioningHookResponse handleRequest(PreProvisioningHookRequest object,
 Context context) {
        Map<String, String> parameterOverrides = new HashMap<String, String>();
        parameterOverrides.put("DeviceLocation", "Seattle");

        PreProvisioningHookResponse response = PreProvisioningHookResponse.builder()
                .allowProvisioning(true)
                .parameterOverrides(parameterOverrides)
                .build();
```

```
        return response;
    }

}
```

Classe de demande :

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;

@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class PreProvisioningHookRequest {
    private String claimCertificateId;
    private String certificateId;
    private String certificatePem;
    private String templateArn;
    private String clientId;
    private Map<String, String> parameters;
}
```

Classe de réponse :

```
package example;

import java.util.Map;
import lombok.Builder;
import lombok.Data;
import lombok.AllArgsConstructor;
import lombok.NoArgsConstructor;


@Data
@Builder
@AllArgsConstructor
@NoArgsConstructor
public class PreProvisioningHookResponse {
    private boolean allowProvisioning;
    private Map<String, String> parameterOverrides;
}
```

API MQTT de mise en service des appareils

Le service Fleet Provisioning prend en charge ces API MQTT :

- the section called “CreateCertificateFromCsr” (p. 752)
- the section called “CreateKeysAndCertificate” (p. 753)
- the section called “RegisterThing” (p. 755)

Cette API prend en charge les tampons de réponse au format CBOR (Concise Binary Object Représentation) et JSON (JavaScript Object Notation), selon le *payload-format* de la rubrique. Par

souci de clarté, cependant, les exemples de réponse et de demande de cette section sont présentés au format JSON.

payload-format	Type de données du format de réponse
CBOR	CBOR (Concise Binary Object Representation, représentation concise d'objets binaires)
json	JavaScript Object Notation (JSON)

> Important
>
> Avant de publier une rubrique de message de demande, abonnez-vous aux rubriques de réponse pour recevoir la réponse. Les messages utilisés par cette API utilisent le protocole de publication et d'abonnement de MQTT pour fournir une interaction de demande et réponse.
> Si vous ne vous abonnez pas aux rubriques de réponse avant de publier une demande, il se peut que vous ne receviez pas les résultats de cette demande.

CreateCertificateFromCsr

Crée un certificat à partir d'une demande de signature de certificat (CSR).AWS IoTfournit des certificats clients signés par l'autorité de certification racine Amazon. Le nouveau certificat a un statut `PENDING_ACTIVATION`. Lorsque vous appelez `RegisterThing` pour provisionner un objet avec ce certificat, l'état du certificat devient `ACTIVE` ou `INACTIVE` comme décrit dans le modèle.

> Note
>
> Pour des raisons de sécurité, le`certificateOwnershipToken`Retourné par`CreateCertificateFromCsr` (p. 752)expire après une heure.`RegisterThing` (p. 755)doit être appelé avant la commande`certificateOwnershipToken`Expires. Si le jeton expire, le périphérique peut appeler`CreateCertificateFromCsr` (p. 752)Pour générer un nouveau certificat.

Demande CreateCertificateFromCsr

Publiez un message avec la rubrique `$aws/certificates/create-from-csr/`*`payload-format`*.

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Charge utile de demande CreateCertificateFromCsr

```
{
    "certificateSigningRequest": "string"
}
```

`certificateSigningRequest`

La CSR, au format PEM.

Réponse CreateCertificateFromCsr

S'abonner à `$aws/certificates/create-from-csr/`*`payload-format`*`/accepted`

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Charge utile de réponse CreateCertificateFromCsr

```
{
    "certificateOwnershipToken": "string",
    "certificateId": "string",
    "certificatePem": "string"
}
```

`certificateOwnershipToken`

Le jeton pour prouver la propriété du certificat lors de la mise en service.

`certificateId`

ID du certificat. Les opérations de gestion de certificats prennent uniquement en compte un ID de certificat.

`certificatePem`

Données du certificat, au format PEM.

Erreur CreateCertificateFromCsr

Pour recevoir des réponses d'erreur, abonnez-vous à `$aws/certificates/create-from-csr/`*`payload-format`*`/rejected`.

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Charge utile d'erreur CreateCertificateFromCsr

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

Le code de statut.

`errorCode`

Code de l'erreur.

`errorMessage`

Message d'erreur.

CreateKeysAndCertificate

Crée de nouvelles clés et un certificat.AWS IoTfournit des certificats clients signés par l'autorité de certification racine Amazon. Le nouveau certificat a un statut `PENDING_ACTIVATION`. Lorsque vous

appelez `RegisterThing` pour provisionner un objet avec ce certificat, l'état du certificat devient `ACTIVE` ou `INACTIVE` comme décrit dans le modèle.

> Note
>
> Pour des raisons de sécurité, le`certificateOwnershipToken`Retourné par`CreateKeysAndCertificate` (p. 753)expire après une heure.`RegisterThing` (p. 755)doit être appelé avant la commande`certificateOwnershipToken`Expires. Si le jeton expire, le périphérique peut appeler`CreateKeysAndCertificate` (p. 753)Pour générer un nouveau certificat.

Demande CreateKeysAndCertificate

Publiez un message sur `$aws/certificates/create/`*`payload-format`* avec une charge utile de message vide.

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Réponse CreateKeysAndCertificate

S'abonner à `$aws/certificates/create/`*`payload-format`*`/accepted`

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Réponse CreateKeysAndCertificate

```
{
    "certificateId": "string",
    "certificatePem": "string",
    "privateKey": "string",
    "certificateOwnershipToken": "string"
}
```

`certificateId`

ID du certificat.

`certificatePem`

Données du certificat, au format PEM.

`privateKey`

Clé privée.

`certificateOwnershipToken`

Le jeton pour prouver la propriété du certificat lors de la mise en service.

Erreur CreateKeysAndCertificate

Pour recevoir des réponses d'erreur, abonnez-vous à `$aws/certificates/create/`*`payload-format`*`/rejected`.

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

Charge utile d'erreur CreateKeysAndCertificate

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

Le code de statut.

`errorCode`

Code de l'erreur.

`errorMessage`

Message d'erreur.

RegisterThing

Alloue un objet à l'aide d'un modèle prédéfini.

Demande RegisterThing

Publier un message sur `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

`templateName`

Nom du modèle de mise en service.

Charge utile de la demande RegisterThing

```
{
    "certificateOwnershipToken": "string",
    "parameters": {
        "string": "string",
        ...
    }
}
```

`certificateOwnershipToken`

Jeton pour prouver la propriété du certificat. Le jeton est généré par AWS IoT lorsque vous créez un certificat sur MQTT.

`parameters`

Facultatif. Paires clé-valeur du périphérique utilisées par les hooks de pré-provisionnement (p. 749) pour évaluer la demande d'enregistrement.

Réponse RegisterThing

S'abonner à `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*`/accepted`

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

`templateName`

Nom du modèle de mise en service.

Charge utile de la réponse RegisterThing

```
{
    "deviceConfiguration": {
        "string": "string",
        ...
    },
    "thingName": "string"
}
```

`deviceConfiguration`

Configuration de l'appareil définie dans le modèle.

`thingName`

Nom de l'objet IoT créé lors de la mise en service.

Réponse d'erreur RegisterThing

Pour recevoir des réponses d'erreur, abonnez-vous à `$aws/provisioning-templates/`*`templateName`*`/provision/`*`payload-format`*`/rejected`.

`payload-format`

Format de charge utile du message en tant que `cbor` ou `json`.

`templateName`

Nom du modèle de mise en service.

Charge utile de réponse d'erreur RegisterThing

```
{
    "statusCode": int,
    "errorCode": "string",
    "errorMessage": "string"
}
```

`statusCode`

Le code de statut.

`errorCode`

Code de l'erreur.

`errorMessage`

Message d'erreur.

Service d'indexation de flotte

Le service d'indexation de flotte est un service géré qui vous permet d'indexer, de rechercher et de regrouper des données de registre, des données de shadow et des données de connectivité (événements de cycle de vie d'appareil) dans le cloud. Après avoir configuré votre index de flotte, le service gère l'indexation des mises à jour apportées à vos groupes, vos registres et vos shadows d'appareils. Pour de plus amples informations sur les requêtes d'agrégation, veuillez consulter Interrogation des données agrégées (p. 770). Vous pouvez utiliser un langage de requête simple pour effectuer une recherche dans ces données. Vous pouvez également créer un groupe d'objets dynamique (p. 220) avec une requête de recherche.

> **Note**
>
> La mise à jour de l'index de flotte par le service d'indexation de flotte peut prendre jusqu'à 30 secondes après la création, la mise à jour ou la suppression d'un objet.

Lorsque vous activez l'indexation, AWS IoT crée un index pour vos objets ou groupes d'objets. Une fois qu'il est actif, vous pouvez exécuter des requêtes sur l'index : par exemple, trouver tous les appareils portatifs qui ont une durée de vie de batterie supérieure à 70 %. AWS IoT maintient l'index continuellement à jour avec vos dernières données.

`AWS_Things` est l'index créé pour tous vos objets. `AWS_ThingGroups` est l'index qui contient tous les groupes d'objets.

Vous pouvez utiliser la console AWS IoT pour gérer votre configuration d'indexation et exécuter vos requêtes de recherche. Choisissez les index que vous souhaitez utiliser dans la page des paramètres de la console. Si vous préférez un accès par programme, vous pouvez utiliser les kits SDK AWS ou l'AWS Command Line Interface (AWS CLI).

Pour plus de détails sur la tarification de ce service et des autres, consultez la page Tarification d'AWS IoT Device Management.

Rubriques

Gestion de l'indexation d'objet

`AWS_Things` est l'index créé pour l'ensemble de vos objets. Vous pouvez contrôler les éléments à indexer : données de registre, données de shadow et données de statut de connectivité d'appareil (pilotée par des événements de cycle de vie d'appareil).

Activation de l'indexation d'objet

Vous utilisez la commande de l'interface de commande update-indexing-configuration ou l'API UpdateIndexingConfiguration pour créer l'index `AWS_Things` et contrôler sa configuration. Le paramètre

`--thing-indexing-configuration` (`thingIndexingConfiguration`) vous permet de contrôler le type de données indexées (par exemple, données de registre, de shadow et de connectivité d'appareils).

Le paramètre `--thing-indexing-configuration` prend une chaîne avec la structure suivante :

```
{
  "thingIndexingMode": "OFF"|"REGISTRY"|"REGISTRY_AND_SHADOW",
  "thingConnectivityIndexingMode": "OFF"|"STATUS",
  "customFields": [
    { name: field-name, type: String | Number | Boolean },
    ...
  ]
}
```

L'attribut `thingIndexingMode` contrôle le type de données indexées. Les valeurs valides sont :

OFF

Aucune indexation.

REGISTRY

Indexation des données de registre.

REGISTRY_AND_SHADOW

Indexation des données de registre et des données de shadow d'objet.

L'attribut `thingConnectivityIndexingMode` spécifie si les données de connectivité d'objets sont indexées. Les valeurs valides sont :

OFF

Les données de connectivité d'objets ne sont pas indexées.

STATUS

Les données de connectivité d'objets sont indexées.

L'attribut `customFields` est une liste de paires de champs et de types de données. Les requêtes d'agrégation peuvent être effectuées sur ces champs en fonction du type de données. Le mode d'indexation que vous choisissez (REGISTRY ou REGISTRY_AND_SHADOW) affecte les champs qui peuvent être spécifiés dans `customFields`. Par exemple, si vous spécifiez le mode d'indexation `REGISTRY`, vous ne pouvez pas spécifier un champ à partir d'un shadow d'objet. Les champs personnalisés doivent être spécifiés dans `customFields` afin d'être indexés.

S'il existe une incohérence de type entre un champ personnalisé de votre configuration et la valeur indexée, le service d'indexation de flotte ignore la valeur incohérente pour les requêtes d'agrégation. Les journaux CloudWatch sont utiles pour résoudre les problèmes de requête d'agrégation. Pour plus d'informations, consultez Dépannage des requêtes d'agrégation pour le service d'indexation de parc (p. 1150).

Les champs gérés contiennent des données associées aux objets IoT, aux groupes d'objets et aux shadows d'appareils. Le type de données des champs gérés est défini par AWS IoT. Vous spécifiez les valeurs de chaque champ géré lorsque vous créez un objet IoT. Par exemple, les noms d'objets, les groupes d'objets et les descriptions d'objets sont tous des champs gérés. Le service d'indexation de flotte indexe les champs gérés en fonction du mode d'indexation que vous spécifiez :

- Champs gérés pour le registre

```
"managedFields" : [
  {name:thingId, type:String},
  {name:thingName, type:String},
  {name:registry.version, type:Number},
  {name:registry.thingTypeName, type:String},
  {name:registry.thingGroupNames, type:String},
]
```

- Champs gérés pour les shadows d'objets

```
"managedFields" : [
  {name:shadow.version, type:Number},
  {name:shadow.hasDelta, type:Boolean}
]
```

- Champs gérés pour la connectivité d'objets

```
"managedFields" : [
  {name:connectivity.timestamp, type:Number},
  {name:connectivity.version, type:Number},
  {name:connectivity.connected, type:Boolean},
  {name:connectivity.disconnectReason, type:String}
]
```

- Champs gérés pour les groupes d'objets

```
"managedFields" : [
  {name:description, type:String},
  {name:parentGroupNames, type:String},
  {name:thingGroupId, type:String},
  {name:thingGroupName, type:String},
  {name:version, type:Number},
]
```

Les champs gérés ne peuvent pas être modifiés ni s'afficher dans `customFields`.

Voici un exemple d'utilisation de update-indexing-configuration pour configurer l'indexation :

aws iot update-indexing-configuration --thing-indexing-configuration 'thingIndexingMode=REGISTRY_AND_SHADOW,customFields=[{name=attributes.version,type=Number}, {name=attributes.color, type=String},{name=shadow.desired.power, type=Boolean}}]

Cette commande active l'indexation pour les données de registre et de shadows. Les requêtes d'agrégation fonctionnent avec les champs gérés et le `customFields` fourni en fonction du type de données.

Vous pouvez utiliser la commande de l'interface de commande get-indexing-configuration ou l'API GetIndexingConfiguration pour récupérer la configuration d'indexation actuelle.

La commande suivante montre comment utiliser la commande de l'interface de commande get-indexing-configuration pour récupérer la configuration d'indexation des objets actuelle où cinq champs personnalisés (trois champs personnalisés de registre et deux champs personnalisés de shadow) sont définis.

aws iot get-indexing-configuration

```
{
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    },
```

```
    "thingIndexingConfiguration": {
        "thingConnectivityIndexingMode": "STATUS",
        "customFields": [
            {
                "name": "attributes.customField_NUM",
                "type": "Number"
            },
            {
                "name": "shadow.desired.customField_STR",
                "type": "String"
            },
            {
                "name": "shadow.desired.customField_NUM",
                "type": "Number"
            },
            {
                "name": "attributes.customField_STR",
                "type": "String"
            },
            {
                "name": "attributes.customField_BOOL",
                "type": "Boolean"
            }
        ],
        "thingIndexingMode": "REGISTRY_AND_SHADOW",
        "managedFields": [
            {
                "name": "shadow.hasDelta",
                "type": "Boolean"
            },
            {
                "name": "registry.thingGroupNames",
                "type": "String"
            },
            {
                "name": "connectivity.version",
                "type": "Number"
            },
            {
                "name": "connectivity.disconnectReason",
                "type": "String"
            },
            {
                "name": "registry.thingTypeName",
                "type": "String"
            },
            {
                "name": "connectivity.connected",
                "type": "Boolean"
            },
            {
                "name": "registry.version",
                "type": "Number"
            },
            {
                "name": "thingId",
                "type": "String"
            },
            {
                "name": "connectivity.timestamp",
                "type": "Number"
            },
            {
                "name": "thingName",
                "type": "String"
            },
```

```
            {
                "name": "shadow.version",
                "type": "Number"
            }
        ]
    }
}
```

Le tableau suivant fournit les combinaisons autorisées de `thingIndexingMode` et `thingConnectivityIndexingMode`, ainsi que les effets associés. Le paramètre obligatoire `thingIndexingMode` indique si l'index `AWS_Things` contient uniquement les données de registre, ou les données de registre et de shadows. Le paramètre facultatif `thingConnectivityIndexingMode` spécifie si l'index contient également les données de statut de connectivité (c'est-à-dire, les informations concernant la dernière connexion et déconnexion des appareils à AWS IoT).

`thingIndexingMode`	**`thingConnectivityIndexingM`**	Résultat
OFF	Non spécifié.	Aucun indexation ou suppression d'un index.
OFF	OFF	Équivalant à l'entrée précédente.
REGISTRY	Non spécifié.	Création ou configuration de l'index `AWS_Things` afin d'indexer uniquement les données du registre.
REGISTRY	OFF	Équivalant à l'entrée précédente. (Seuls les données de registre sont indexées.)
REGISTRY_AND_SHADOW	Non spécifié.	Création ou configuration de l'index `AWS_Things` afin d'indexer les données de registre et de shadows.
REGISTRY_AND_SHADOW	OFF	Équivalant à l'entrée précédente. (Les données de registre et les données de shadow sont indexées.)
REGISTRY	STATUS	Création ou configuration de l'index `AWS_Things` pour indexer les données de registre et les données de statut de connectivité d'objet (REGISTRY_AND_CONNECTIVITY_STATUS)
REGISTRY_AND_SHADOW	STATUS	Créez ou configurez l'index `AWS_Things` pour indexer les données de registre, les données de shadows et les données de statut de connectivité d'objet (REGISTRY_AND_SHADOW_AND_CONNECTI

Vous pouvez utiliser la commande de l'interface de ligne de commande AWS IoT update-indexing-configuration afin de mettre à jour la configuration d'indexation des objets. Les exemples suivants montrent comment utiliser la commande de l'interface de commande update-indexing-configuration.

Syntaxe courte :

aws iot update-indexing-configuration --thing-indexing-configuration
"thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS,customFields=[{name=attrib
{name=attributes.color,type=String},{name=shadow.desired.power,type=Boolean}]"

Syntaxe JSON :

aws iot update-indexing-configuration --cli-input-json \ '{ "thingIndexingConfiguration":
{ "thingIndexingMode": "REGISTRY_AND_SHADOW", "thingConnectivityIndexingMode": "STATUS",
"customFields": [{ "name": "shadow.desired.power", "type": "Boolean" }, { "name": "attributes.color", "type":
"String" }, { "name": "attributes.version", "type": "Number" }] } }'

La sortie de ces commandes est :


```
{
    "thingIndexingConfiguration": {
        "thingConnectivityIndexingMode": "STATUS",
        "customFields": [
            {
                "type": "String",
                "name": "attributes.color"
            },
            {
                "type": "Number",
                "name": "attributes.version"
            },
            {
                "type": "Boolean",
                "name": "shadow.desired.power"
            }
        ],
        "thingIndexingMode": "REGISTRY_AND_SHADOW",
        "managedFields": [
            {
                "type": "Boolean",
                "name": "connectivity.connected"
            },
            {
                "type": "String",
                "name": "registry.thingTypeName"
            },
            {
                "type": "String",
                "name": "thingName"
            },
            {
                "type": "Number",
                "name": "shadow.version"
            },
            {
                "type": "String",
                "name": "thingId"
            },
            {
                "type": "Boolean",
                "name": "shadow.hasDelta"
            },
            {
                "type": "Number",
                "name": "connectivity.timestamp"
            },
```

```
            {
                "type": "String",
                "name": "connectivity.disconnectReason"
            },
            {
                "type": "String",
                "name": "registry.thingGroupNames"
            },
            {
                "type": "Number",
                "name": "connectivity.version"
            },
            {
                "type": "Number",
                "name": "registry.version"
            }
        ]
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "OFF"
    }
}
```

Dans l'exemple suivant, un nouveau champ personnalisé est ajouté à la configuration :

aws iot update-indexing-configuration --thing-indexing-configuration 'thingIndexingMode=REGISTRY_AND_SHADOW,customFields=[{name=attributes.version,type=Number},{name=attributes.color,type=String},{name=shadow.desired.power,type=Boolean},{name=shadow.desired.intensity,type=Number}]'

Cette commande a ajouté `shadow.desired.intensity` à la configuration d'indexation.

> **Note**
>
> La mise à jour de la configuration d'indexation des champs personnalisés remplace tous les champs personnalisés existants. Assurez-vous de spécifier tous les champs personnalisés lorsque vous appelez update-indexing-configuration.

Une fois l'index régénéré, vous pouvez utiliser la requête d'agrégation sur les champs nouvellement ajoutés et faire des recherches dans les données de registre, les données de shadow et les données de statut de connectivité d'objet.

Lorsque vous modifiez le mode d'indexation, assurez-vous que tous vos champs personnalisés sont valides à l'aide du nouveau mode d'indexation. Par exemple, si vous commencez par le mode `REGISTRY_AND_SHADOW` avec un champ personnalisé appelé, `shadow.desired.temperature`, vous devez supprimer le champ personnalisé `shadow.desired.temperature` avant de remplacer le mode d'indexation par `REGISTRY`. Si votre configuration d'indexation contient des champs personnalisés qui ne sont pas indexés par le mode d'indexation, la mise à jour échoue.

Description d'un index d'objets

La commande suivante montre comment utiliser la commande d'interface de ligne de commande describe-index pour extraire l'état actuel de l'index d'objets.

```
aws iot describe-index --index-name "AWS_Things"
{
    "indexName": "AWS_Things",
    "indexStatus": "BUILDING",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

La première fois que vous activez l'indexation, AWS IoT crée votre index. Vous ne pouvez pas exécuter de requêtes sur l'index `indexStatus` si son état est `BUILDING`. Le `schema` de l'index d'objets indique le type de données (`REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS`) qui est indexé.

Si vous modifiez la configuration de votre index, ce dernier est recréé. Lors de ce processus, l'`indexStatus` est `REBUILDING`. Vous pouvez exécuter des requêtes sur les données de l'index d'objets pendant sa régénération. Par exemple, si vous faites passer la configuration d'index de `REGISTRY` à `REGISTRY_AND_SHADOW`, pendant sa régénération, vous pouvez interroger les données de registre, y compris les dernières mises à jour. Toutefois, vous ne pouvez pas interroger les données des shadows tant que la reconstruction n'est pas terminée. Le temps nécessaire pour créer ou recréer l'index dépend de la quantité de données.

Interrogation d'un index d'objets

Utilisez la commande de l'interface de ligne de commande search-index pour interroger les données dans l'index.

aws iot search-index --index-name "AWS_Things" --query-string "thingName:mything*"

```
{
    "things":[{
         "thingName":"mything1",
         "thingGroupNames":[
            "mygroup1"
         ],
         "thingId":"a4b9f759-b0f2-4857-8a4b-967745ed9f4e",
         "attributes":{
            "attribute1":"abc"
         },
         "connectivity": {
            "connected":false,
            "timestamp":1556649874716
         }
    },
    {
        "thingName":"mything2",
        "thingTypeName":"MyThingType",
        "thingGroupNames":[
            "mygroup1",
            "mygroup2"
        ],
        "thingId":"01014ef9-e97e-44c6-985a-d0b06924f2af",
        "attributes":{
            "model":"1.2",
            "country":"usa"
        },
        "shadow":{
            "desired":{
                "location":"new york",
                "myvalues":[3, 4, 5]
            },
            "reported":{
                "location":"new york",
                "myvalues":[1, 2, 3],
                "stats":{
                    "battery":78
                }
            },
            "metadata":{
                 "desired":{
                       "location":{
```

```
                        "timestamp":123456789
                    },
                   "myvalues":{
                        "timestamp":123456789
                   }
               },
               "reported":{
                    "location":{
                        "timestamp":34535454
                     },
                    "myvalues":{
                        "timestamp":34535454
                    },
                    "stats":{
                        "battery":{
                            "timestamp":34535454
                        }
                    }
                }
            },
            "version":10,
            "timestamp":34535454
        },
        "connectivity": {
            "connected":true,
            "timestamp":1556649855046
        }
    }],
    "nextToken":"AQFCuvk7zZ3D9pOYMbFCeHbdZ+h=G"
}
```


Dans la réponse JSON `"connectivity"` (activé par le paramètre `thingConnectivityIndexingMode=STATUS`) fournit une valeur booléenne et un horodatage indiquant si l'appareil est connecté à AWS IoT Core . L'appareil `"mything1"` déconnecté (`false`) à l'heure POSIX `1556649874716` :


```
"connectivity": {
    "connected":false,
    "timestamp":1556649874716
}
```


L'appareil `"mything2"` connecté (`true`) à l'heure POSIX `1556649855046` :


```
"connectivity": {
    "connected":true,
    "timestamp":1556649855046
}
```


Les horodatages sont indiqués en millisecondes depuis l'époque Unix, donc `1556649855046` représente 6:44:15.046 PM le mardi 30 avril 2019 (GMT).

> **Important**
>
> Si un appareil a été déconnecté pendant environ une heure, la valeur `"timestamp"` du statut de connectivité peut être manquante.

Limites et restrictions

Les restrictions et limitations pour `AWS_Things` sont les suivantes.

Champs Shadow de types complexes

Un champ shadow est indexé uniquement si la valeur du champ est un type simple, un objet JSON qui ne contient pas de tableau ou un tableau qui se compose entièrement de types simples. (Un type simple représente une chaîne, un nombre ou un des littéraux `true` ou `false`.) Par exemple, étant donné l'état du shadow suivant, la valeur du champ `"palette"` n'est pas indexée, car il s'agit d'un tableau dont les éléments ont des types complexes. La valeur du champ `"colors"` est indexée, car chaque valeur du tableau est une chaîne.

```
{
    "state": {
        "reported": {
            "switched": "ON",
            "colors": [ "RED", "GREEN", "BLUE" ],
            "palette": [
                {
                    "name": "RED",
                    "intensity": 124
                },
                {
                    "name": "GREEN",
                    "intensity": 68
                },
                {
                    "name": "BLUE",
                    "intensity": 201
                }
            ]
        }
    }
}
```

Noms de champs shadow imbriqués

Les noms des champs shadow imbriqués sont stockés sous la forme d'une chaîne délimitée par un point (.). Par exemple, soit un document shadow :

```
{
  "state": {
    "desired": {
      "one": {
        "two": {
          "three": "v2"
        }
      }
    }
  }
}
```

le nom du champ `three` est stocké sous la forme `desired.one.two.three`. Si vous avez également un document shadow similaire à ce qui suit :

```
{
  "state": {
    "desired": {
      "one.two.three": "v2"
    }
  }
}
```

ils correspondent tous deux à une requête pour `shadow.desired.one.two.three:v2`. Conformément aux bonnes pratiques, n'utilisez pas de points dans les noms de champs shadow.

Métadonnées de shadow

Un champ d'une section de métadonnées de shadow est indexé, à l'unique condition que le champ correspondant dans la section `"state"` du shadow soit indexé. (Dans l'exemple précédent, le champ `"palette"` de la section des métadonnées du shadow n'est pas indexé non plus.)

Shadows non enregistrés

Si vous utilisez UpdateThingShadow pour créer un shadow à l'aide d'un nom d'objet qui n'a pas été enregistré dans votre compte AWS IoT, les champs de ce shadow ne sont pas indexés.

Valeur numériques

Si des données de registre ou de shadow sont reconnues par le service en tant que valeurs numériques, elles sont indexées en tant que telles. Vous pouvez formuler des requêtes comportant des plages et des opérateurs de comparaison sur les valeurs numériques (par exemple, `"attribute.foo<5"` ou `"shadow.reported.foo:[75 TO 80]"`). Pour être considérée comme numérique, la valeur des données doit être un littéral JSON valide de type nombre (un entier dans la plage - 2^53... 2^53-1 ou une double précision à virgule flottante avec une notation exponentielle facultative) ou une partie d'un tableau contenant uniquement ces valeurs.

Valeurs null

Les valeurs null ne sont pas indexées.

Nombre maximal de champs personnalisés pour les requêtes d'agrégation

5

Nombre maximal de centiles demandés pour les requêtes d'agrégation.

100

Authorization

Vous pouvez spécifier l'index d'objets en tant qu'ARN de ressource dans une action de stratégie AWS IoT, de la manière suivante.

Action	Ressource
`iot:SearchIndex`	Un ARN d'index (par exemple, `arn:aws:iot:`*`your-aws-regionyour-aws-account`*`:index/AWS_Things`).
`iot:DescribeIndex`	Un ARN d'index (par exemple, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_Things`).

Note

Si vous disposez d'autorisations pour interroger l'index de la flotte, vous pouvez accéder aux données d'objets dans la totalité de la flotte.

Gestion de l'indexation de groupes d'objets

`AWS_ThingGroups` est l'index qui contient tous les groupes de votre objet. Cet index vous permet de rechercher des groupes en fonction de leur nom, de la description, des attributs et de tous les noms de groupes parents.

Activation de l'indexation de groupes d'objets

Vous pouvez utiliser le paramètre `thing-group-indexing-configuration` de l'API UpdateIndexingConfiguration pour créer l'index `AWS_ThingGroups` et en contrôler la configuration. Vous pouvez utiliser l'API GetIndexingConfiguration pour extraire la configuration d'indexation actuelle.

Utilisez la commande de l'interface de ligne de commande get-indexing-configuration pour récupérer l'objet actuel et les configurations d'indexation de groupes d'objets.

aws iot get-indexing-configuration

```
{
   "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "ON"
    }
}
```

Vous pouvez utiliser la commande d'interface de ligne de commande update-indexing-configuration pour mettre à jour les configurations d'indexation de groupes d'objets.

```
aws iot update-indexing-configuration --thing-group-indexing-configuration
 thingGroupIndexingMode=ON
```

Note

Vous pouvez également mettre à jour les configurations d'indexation d'objets et de groupes d'objets avec une seule commande, comme indiqué ci-dessous.

```
aws iot update-indexing-configuration --thing-indexing-configuration
 thingIndexingMode=REGISTRY --thing-group-indexing-configuration
 thingGroupIndexingMode=ON
```

Les valeurs suivantes sont valides pour `thingGroupIndexingMode`.

OFF

Pas d'indexation/suppression de l'index.

ON

Créez ou configurez l'index `AWS_ThingGroups`.

Description des index de groupes

Utilisez la commande de l'interface de ligne de commande describe-index pour récupérer le statut actuel de l'index `AWS_ThingGroups`.

```
aws iot describe-index --index-name "AWS_ThingGroups"
{
   "indexStatus": "ACTIVE",
   "indexName": "AWS_ThingGroups",
   "schema": "THING_GROUPS"
}
```

AWS IoT crée votre index la première fois que vous activez l'indexation. Vous ne pouvez pas interroger l'index si le `indexStatus` est `BUILDING`.

Interrogation d'un index de groupes d'objets

Utilisez la commande de l'interface de ligne de commande search-index pour interroger les données dans l'index :

```
aws iot search-index --index-name "AWS_ThingGroups" --query-string
 "thingGroupNames:mythinggroup*"
```

Authorization

Vous pouvez spécifier l'index de groupes d'objets en tant qu'ARN de ressource dans une action de stratégie AWS IoT, de la manière suivante.

Action	Ressource
`iot:SearchIndex`	Un ARN d'index (par exemple, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).
`iot:DescribeIndex`	Un ARN d'index (par exemple, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).

Interrogation des données agrégées

AWS IoT fournit trois API (`GetStatistics`, `GetCardinality` et `GetPercentiles`) qui vous permettent de rechercher des données agrégées dans votre flotte d'appareils.

GetStatistics

L'API GetStatistics et la commande get-statistics de l'interface de ligne de commande renvoient le nombre, la moyenne, la somme, le minimum, le maximum, la somme des carrés, la variance et l'écart type du champ agrégé spécifié.

La commande get-statistics de l'interface de ligne de commande (CLI) utilise les paramètres suivants :

`index-name`

Nom de l'index dans lequel effectuer la recherche. La valeur par défaut est `AWS_Things`.

`query-string`

Nom de la requête utilisée pour recherche dans l'index. Vous pouvez spécifier`"*"`pour obtenir le nombre de tous les objets indexés dans votre Compte AWS .

`aggregationField`

Facultatif. Champ à agréger. Ce champ doit être un champ géré ou personnalisé défini lorsque vous appelez update-indexing-configuration. Si vous ne spécifiez pas de champ d'agrégation, `registry.version` est utilisé comme champ d'agrégation.

`query-version`

Version de la requête à utiliser. La valeur par défaut est `2017-09-30`.

Le type de champ d'agrégation peut affecter les statistiques renvoyées.

GetStatistics avec des valeurs de chaîne

Si vous regroupez les données en fonction d'un champ de chaîne, l'appel à `GetStatistics` renvoie un nombre d'appareils dont les attributs correspondent à la requête. Exemples :

aws iot get-statistics --aggregation-field 'attributes.stringAttribute' --query-string '*'

Cette commande renvoie le nombre d'appareils qui contiennent un attribut nommé `stringAttribute` :

```
{
  "statistics": {
    "count": 3
  }
}
```

GetStatistics avec des valeurs booléennes

Lorsque vous appelez `GetStatistics` avec un champ d'agrégation booléenne :

- AVERAGE est le pourcentage d'appareils qui correspondent à la requête.
- MINIMUM est 0 ou 1, d'après les règles suivantes :
  - Si toutes les valeurs du champ d'agrégation sont `false`, MINIMUM est 0.
  - Si toutes les valeurs du champ d'agrégation sont `true`, MINIMUM est 1.
  - Si des valeurs du champ d'agrégation sont `false` et d'autres `true`, MINIMUM est 0.
- MAXIMUM est 0 ou 1, d'après les règles suivantes :
  - Si toutes les valeurs du champ d'agrégation sont `false`, MAXIMUM est 0.
  - Si toutes les valeurs du champ d'agrégation sont `true`, MAXIMUM est 1.
  - Si des valeurs du champ d'agrégation sont `false` et d'autres `true`, MAXIMUM est 1.
- SUM est la somme de l'équivalent entier des valeurs booléennes.
- COUNT est le nombre d'objets qui correspondent à la requête.

GetStatistics avec des valeurs numériques

Lorsque vous appelez `GetStatistics` et spécifiez un champ d'agrégation de type `Number`, `GetStatistics` renvoie les valeurs suivantes :

count

Nombre d'appareils dont un champ correspond à la requête.

average

Moyenne des valeurs numériques qui correspondent à la requête.

sum

Somme des valeurs numériques qui correspondent à la requête.

minimum

La plus petite des valeurs numériques qui correspondent à la requête.

maximum

La plus grande des valeurs numériques qui correspondent à la requête.

sumOfSquares

Somme des carrés des valeurs numériques qui correspondent à la requête.

variance

Variance des valeurs numériques qui correspondent à la requête. La variance d'un ensemble de valeurs est la moyenne des carrés des différences de chaque valeur par rapport à la valeur moyenne de l'ensemble.

stdDeviation

Écart type des valeurs numériques qui correspondent à la requête. L'écart type d'un ensemble de valeurs est une mesure de la répartition des valeurs.

L'exemple suivant montre comment appeler get-statistics avec un champ numérique personnalisé.

aws iot get-statistics --aggregation-field 'attributes.numericAttribute2' --query-string '*'

```
{
  "statistics": {
    "count": 3,
    "average": 33.333333333333336,
    "sum": 100.0,
    "minimum": -125.0,
    "maximum": 150.0,
    "sumOfSquares": 43750.0,
    "variance": 13472.22222222222,
    "stdDeviation": 116.06990230986766
  }
}
```

Pour les champs d'agrégation numérique, si les valeurs des champs dépassent la valeur double maximale, les valeurs statistiques sont vides

GetCardinality

L'API GetCardinality et la commande get-cardinality de l'interface de ligne de commande renvoient le nombre approximatif de valeurs uniques qui correspondent à la requête. Par exemple, vous pouvez trouver le nombre d'appareils dont le niveau de batterie est inférieur à 50 % :

aws iot get-cardinality --index-name AWS_Things --query-string "batterylevel > 50" --aggregation-field "shadow.reported.batterylevel".

Cette commande renvoie le nombre d'objets dont le niveau de batterie est supérieur à 50 % :

```
{
    "cardinality": 100
}
```

`cardinality` est toujours renvoyé par get-cardinality, même s'il n'y a pas de champs correspondants. Exemples :

aws iot get-cardinality --query-string "thingName:Non-existent*" --aggregation-field "attributes.customField_STR"

```
{
    "cardinality": 0
}
```

La commande get-cardinality de l'interface de ligne de commande (CLI) utilise les paramètres suivants :

`index-name`

Nom de l'index dans lequel effectuer la recherche. La valeur par défaut est `AWS_Things`.

`query-string`

Nom de la requête utilisée pour recherche dans l'index. Vous pouvez spécifier`"*"`pour obtenir le nombre de tous les objets indexés dans votre Compte AWS .

`aggregationField`

Champ à agréger.

`query-version`

Version de la requête à utiliser. La valeur par défaut est `2017-09-30`.

GetPercentiles

L'API GetPercentiles et la commande de l'interface de commande get-percentiles regroupe les valeurs agrégées qui correspondent à la requête en groupes de centiles. Les groupes de centiles par défaut sont : 1,5,25,50,75,95,99, bien que vous puissiez spécifier les vôtres lorsque vous appelez `GetPercentiles`. Cette fonction renvoie une valeur pour chaque groupe de centiles spécifié (ou les groupes de centiles par défaut). Le groupe de centiles « 1 » contient la valeur de champ agrégée qui se produit dans environ 1 % des valeurs qui correspondent à la requête. Le groupe de centiles « 5 » contient la valeur de champ agrégée qui se produit dans environ 5 % des valeurs qui correspondent à la requête, etc. Le résultat est une approximation. Plus les valeurs correspondent à la requête, plus les valeurs de centile sont précises.

L'exemple suivant montre comment appeler la commande de l'interface de ligne de commande get-percentiles.

aws iot get-percentiles --query-string "thingName:*" --aggregation-field "attributes.customField_NUM" --percents 10 20 30 40 50 60 70 80 90 99

```
{
    "percentiles": [
        {
            "value": 3.0,
            "percent": 80.0
        },
        {
            "value": 2.5999999999999996,
            "percent": 70.0
        },
        {
            "value": 3.0,
            "percent": 90.0
        },
        {
            "value": 2.0,
            "percent": 50.0
        },
        {
            "value": 2.0,
```

```
            "percent": 60.0
        },
        {
            "value": 1.0,
            "percent": 10.0
        },
        {
            "value": 2.0,
            "percent": 40.0
        },
        {
            "value": 1.0,
            "percent": 20.0
        },
        {
            "value": 1.4,
            "percent": 30.0
        },
        {
            "value": 3.0,
            "percent": 99.0
        }
    ]
}
```

La commande suivante affiche la sortie renvoyée par get-percentiles lorsqu'il n'y a pas de documents correspondants.

aws iot get-percentiles --query-string "thingName:Non-existent*" --aggregation-field "attributes.customField_NUM"

```
{
    "percentiles": []
}
```

La commande get-percentile de l'interface de ligne de commande (CLI) utilise les paramètres suivants :

`index-name`

Nom de l'index dans lequel effectuer la recherche. La valeur par défaut est `AWS_Things`.

`query-string`

Nom de la requête utilisée pour recherche dans l'index. Vous pouvez spécifier`"*"`pour obtenir le nombre de tous les objets indexés dans votre Compte AWS .

`aggregationField`

Champ à agréger, dont le type doit être `Number`.

`query-version`

Version de la requête à utiliser. La valeur par défaut est `2017-09-30`.

`percents`

Facultatif. Vous pouvez utiliser ce paramètre pour spécifier des groupes de centiles personnalisés.

Authorization

Vous pouvez spécifier l'index de groupes d'objets en tant qu'ARN de ressource dans une action de stratégie AWS IoT, de la manière suivante.

Action	Ressource
`iot:GetStatistics`	Un ARN d'index (par exemple, `arn:aws:iot:`*`your-aws-region`*`:index/AWS_Things` ou `arn:aws:iot:`*`your-aws-region`*`:index/AWS_ThingGroups`).

Syntaxe de requête

Les requêtes sont spécifiées à l'aide d'une syntaxe de requête.

Cette syntaxe de requête prend en charge les fonctions ci-dessous.

- Termes et expressions
- Champs de recherche
- Recherche de préfixe
- Recherche de plage
- Opérateurs booléens `AND`, `OR`, `NOT` et `-`. Le trait d'union est utilisé pour exclure quelque chose des résultats de recherche (par exemple, `thingName:(tv* AND -plasma)`).
- Regroupement
- Regroupement de champs
- Échappement de caractères spéciaux (comme avec \)

Cette syntaxe de requête ne prend pas en charge les fonctions suivantes :

- Recherche avec caractère générique en préfixe (par exemple, « *xyz »), mais la recherche de « * » donne un résultat de recherche contenant tous les objets
- Expressions régulières
- Promotion
- Classement
- Recherches approximatives
- Recherche de proximité
- Tri
- Agrégation

Quelques remarques concernant le langage de requête :

- L'opérateur par défaut est AND. Une requête pour `"thingName:abc thingType:xyz"` équivaut à `"thingName:abc AND thingType:xyz"`.
- Si aucun champ n'est spécifié, AWS IoT recherche le terme dans tous les champs.
- Tous les noms de champs sont sensibles à la casse.
- La recherche est insensible à la casse. Les mots sont séparés par des caractères d'espace vide, comme défini par l'élément Java `Character.isWhitespace(int)`.
- L'indexation des données de shadows d'appareil inclut les sections reported, desired, delta et metadata.
- Il n'est pas possible d'effectuer une recherche sur les versions du registre et des shadows d'appareil, mais elle sont présentes dans la réponse.

- Le nombre maximum de termes dans une requête est de 5.

Exemples de requêtes sur des objets

Les requêtes sont spécifiées dans une chaîne de requête à l'aide d'une syntaxe de requête, puis transmises à l'API `SearchIndex`. Le tableau ci-après répertorie quelques exemples de chaînes de requête.

Chaîne de requête	Résultat
abc	Requêtes portant sur la chaîne « abc » dans n'importe quel champ du registre ou d'un shadow.
thingName:myThingName	Requêtes concernant un objet nommé « myThingName ».
thingName:my*	Requêtes concernant les objets dont le nom commence par « my ».
thingName:ab?	Requêtes concernant les objets dont le nom contient la chaîne « ab » suivie d'un caractère supplémentaire, par exemple : « aba », « abb », « abc », etc.
thingTypeName:aa	Requêtes pour les objets qui sont associés au type aa.
attributes.myAttribute:75	Requêtes concernant les objets qui comportent un attribut nommé « myAttribute » ayant pour valeur 75.
attributes.myAttribute:[75 TO 80]	Requêtes concernant les objets qui comprennent un attribut nommé « myAttribute », dont la valeur est comprise dans la plage numérique 75—80 inclus.
attributes.myAttribute:{75 TO 80]	Requêtes concernant les objets qui comportent un attribut appelé « myAttribute », dont la valeur est comprise dans la plage numérique >75 et <=80.
attributes.serialNumber:["abcd" TO "abcf"]	Requêtes concernant les objets qui comportent un attribut nommé « serialNumber » dont la valeur est comprise dans une plage de chaînes alphanumériques. Cette requête renvoie les objets dont l'attribut « serialNumber » a la valeur « abcd », « abce » ou « abcf ».
attributes.myAttribute:i*t	Requêtes concernant les objets qui comportent un attribut nommé « myAttribute » dont la valeur est « i », suivi de n'importe quel nombre de caractères, puis du caractère « t ».
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Requêtes concernant les objets qui combinent des termes en utilisant des expressions booléennes. Cette requête renvoie les objets qui comportent un attribut nommé « attr1 » de valeur « abc », un attribut nommé « attr2 » qui est inférieur à 5 et un

Chaîne de requête	Résultat
	attribut nommé « attr3 » qui n'est pas supérieur à 10.
shadow.hasDelta:true	Requêtes concernant les objets dont le shadow comporte un élément delta.
NOT attributes.model:legacy	Requêtes concernant les objets dont l'attribut nommé « model » n'est pas défini sur « legacy ».
shadow.reported.stats.battery:{70 TO 100} (v2 OR v3) NOT attributes.model:legacy	Requêtes concernant les objets possédant les caractéristiques suivantes : • L'attribut `stats.battery` du shadow de l'objet possède une valeur comprise entre 70 et 100. • Le texte « v2 » ou « v3 » se retrouve dans le nom, le nom de type ou les valeurs d'attribut de l'objet. • L'attribut `model` de l'objet n'est pas défini sur « legacy ».
shadow.reported.myvalues:2	Requêtes concernant les objets dont la plage `myvalues` dans la section reported du shadow contient une valeur 2.
shadow.reported.location:* NOT shadow.desired.stats.battery:*	Requêtes concernant les objets possédant les caractéristiques suivantes : • L'attribut `location` figure dans la section `reported` du shadow. • L'attribut `stats.battery` ne figure pas dans la section `desired` du shadow.
connectivity.connected:true	Requêtes pour tous les appareils connectés.
connectivity.connected:false	Requêtes pour tous les appareils déconnectés.
connectivity.connected:true and connectivity.timestamp : [1557651600000 À 1557867600000]	Requêtes pour tous les appareils connectés avec un horodatage de connexion >= 1557651600000 and <= 1557867600000. Les horodatages sont indiqués en millisecondes depuis l’époque Unix.
connectivity.connected:false and connectivity.timestamp : [1557651600000 À 1557867600000]	Requêtes pour tous les appareils déconnectés avec un horodatage de déconnexion >= 1557651600000 and <= 1557867600000. Les horodatages sont indiqués en millisecondes depuis l’époque Unix.
connectivity.connected:true AND connectivity.timestamp > 1557651600000	Requêtes pour tous les appareils connectés avec un horodatage de connexion > 1557651600000. Les horodatages sont indiqués en millisecondes depuis l’époque Unix.
connectivity.connected:*	Requêtes pour tous les appareils comportant des informations de connectivité.

Exemples de requêtes sur des groupes d'objets

Les requêtes sont spécifiées dans une chaîne de requête à l'aide d'une syntaxe de requête, puis transmises à l'API `SearchIndex`. Le tableau ci-après répertorie quelques exemples de chaînes de requête.

Chaîne de requête	Résultat
abc	Requêtes pour « abc » dans n'importe quel champ.
ThingGroupNames:myGroupThingName	Requêtes concernant un groupe d'objets nommé « myGroupThingName ».
NomGroupNames:mon*	Requêtes concernant les groupes d'objets dont le nom commence par « my ».
Noms de groupes:AB ?	Requêtes concernant les groupes d'objets dont le nom contient la chaîne « ab » suivie d'un caractère supplémentaire, par exemple : « aba », « abb », « abc », etc.
attributes.myAttribute:75	Requêtes concernant les groupes d'objets qui comportent un attribut nommé « myAttribute » ayant pour valeur 75.
attributes.myAttribute:[75 TO 80]	Requêtes concernant les groupes d'objets qui comprennent un attribut nommé « myAttribute », dont la valeur est comprise dans la plage numérique 75—80 inclus.
attributes.myAttribute:[75 TO 80]	Requêtes concernant les groupes d'objets qui comportent un attribut nommé « myAttribute », dont la valeur est comprise dans la plage numérique >75 et <=80.
attributes.myAttribute:["abcd" TO "abcf"]	Requêtes concernant les groupes d'objets qui comportent un attribut nommé « myAttribute », dont la valeur est comprise dans une plage de chaînes alphanumériques. Cette requête renvoie les groupes d'objets dont l'attribut « serialNumber » a la valeur « abcd », « abce » ou « abcf ».
attributes.myAttribute:i*t	Requêtes concernant les groupes d'objets qui comportent un attribut nommé « myAttribute » dont la valeur est « i », suivi de n'importe quel nombre de caractères, puis du caractère « t ».
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Requêtes concernant les groupes d'objets qui combinent des termes en utilisant des expressions booléennes. Cette requête renvoie les groupes d'objets qui comportent un attribut nommé « attr1 » de valeur « abc », un attribut nommé « attr2 » qui est inférieur à 5 et un attribut nommé « attr3 » qui n'est pas supérieur à 10.
NOT attributes.myAttribute:cde	Requêtes concernant les groupes d'objets dont l'attribut nommé « myAttribute » n'est pas « cde ».

Chaîne de requête	Résultat
parentGroupNames:myParentThingGroupName)	Requêtes concernant les groupes d'objets dont le nom de groupe parent correspond à « myParentThingGroupName ».
parentGroupNames:(myParentThingGroupName OR myRootThingGroupName)	Requêtes concernant les groupes d'objets dont le nom de groupe parent correspond à « myParentThingGroupName » ou « myRootThingGroupName ».
parentGroupNames:(myParentThingGroupNa*)	Requêtes concernant les groupes d'objets dont le nom de groupe parent commence par « myParentThingGroupNa ».

Livraison de fichiers basée sur MQT

Une option que vous pouvez utiliser pour gérer les fichiers et les transférer versAWS IoTde votre flotte est la livraison de fichiers basée sur MQT. Avec cette fonctionnalité dans la sectionAWSCloud, vous pouvez créer unstreamqui contient plusieurs fichiers, vous pouvez mettre à jour les données de flux (liste de fichiers et descriptions), obtenir les données de flux, etc.AWS IoT La livraison de fichiers basée sur MQT peut transférer des données en petits blocs vers vos périphériques IoT, en utilisant le protocole MQTT avec prise en charge des messages de demande et de réponse dans JSON ou CBOR.

Pour plus d'informations sur les moyens de transférer des données vers et depuis des appareils IoT à l'aide deAWS IoT, voirConnexion d'appareils àAWS IoT (p. 75).

Rubriques

Qu'est-ce qu'un flux ?

DansAWS IoT, unstreamest une ressource adressable publiquement qui est une abstraction pour une liste de fichiers qui peuvent être transférés vers un périphérique IoT. Un flux standard contient les informations suivantes :

- Un Amazon Resource Name (ARN)qui identifie de manière unique un flux à un moment donné. Cet ARN présente le modèle`arn:`*`partition`*`:iot:`*`region`*`:`*`account-ID`*`:stream/`*`stream ID`*.
- ID du fluxqui identifie votre flux et est utilisé (et généralement requis) dansAWS Command Line Interface(AWS CLI) ou des commandes SDK.
- Une description du fluxFournit une description de la ressource de flux.
- Une version de fluxqui identifie une version particulière du flux. Étant donné que les données de flux peuvent être modifiées immédiatement avant que les périphériques ne démarrent le transfert de données, la version de flux peut être utilisée par les périphériques pour appliquer une vérification de cohérence.
- Liste de fichiersqui peuvent être transférés sur des appareils. Pour chaque fichier de la liste, le flux enregistre un ID de fichier, la taille du fichier et les informations d'adresse du fichier, qui comprennent, par exemple, le nom du compartiment Amazon S3, la clé d'objet et la version de l'objet.
- UnAWS Identity and Access Management(IAM)qui accordeAWS IoTDélivrance de fichiers basés sur MQT l'autorisation de lire les fichiers de flux stockés dans le stockage de données.

AWS IoTLa remise de fichiers basée sur MQT fournit les fonctionnalités suivantes afin que les périphériques puissent transférer des données à partir duAWSCloud :

- Transfert de données à l'aide du protocole MQTT.
- Support des formats JSON ou CBOR.
- La possibilité de décrire un flux (`DescribeStream`) pour obtenir une liste de fichiers de flux, une version de flux et des informations connexes.
- La possibilité d'envoyer des données en petits blocs (`GetStream`) afin que les périphériques avec des contraintes matérielles puissent recevoir les blocs.

- Support d'une taille de bloc dynamique par requête, pour prendre en charge les périphériques ayant des capacités de mémoire différentes.
- Optimisation des demandes de streaming simultanées lorsque plusieurs périphériques demandent des blocs de données à partir du même fichier de flux.
- Amazon S3 comme stockage de données pour les fichiers de flux.
- Support de la publication des journaux de transfert de donnéesAWS IoTRemise de fichiers basée sur MQT à CloudWatch.

Pour les quotas de remise de fichiers basés sur MQT, consultez AWS IoT Core Quotas de servicedans leAWSRéférence générale.

Gestion d'un flux dans leAWSCloud

AWS IoTfournitAWSSDK etAWS CLIque vous pouvez utiliser pour gérer un flux dansAWSCloud. Vous pouvez utiliser pour effectuer les tâches suivantes :

- Créer un flux. CLI/KIT DE DÉVELOPPEMENT LOGICIEL
- Décrivez un flux pour obtenir ses informations. CLI/KIT DE DÉVELOPPEMENT LOGICIEL
- Afficher la liste des flux dans votre Compte AWS . CLI/KIT DE DÉVELOPPEMENT LOGICIEL
- Mettez à jour la liste des fichiers ou la description du flux dans un flux. CLI/KIT DE DÉVELOPPEMENT LOGICIEL
- Supprimez un flux. CLI/KIT DE DÉVELOPPEMENT LOGICIEL

> Note
>
> À l'heure actuelle, les flux ne sont pas visibles dans leAWS Management Console. Vous devez utiliser leAWS CLIouAWSSDK pour gérer un flux dansAWS IoT.

Avant d'utiliserAWS IoTDiffusion de fichiers basée sur MQT à partir de vos appareils, vous devez suivre les étapes décrites dans les sections suivantes pour vous assurer que vos appareils sont correctement autorisés et peuvent se connecter auAWS IoTPériphérique de passerelle.

Accordez des autorisations à vos appareils

Vous pouvez suivre les étapes dansCréation d'unAWS IoTStratégiepour créer une stratégie de périphérique ou utiliser une stratégie de périphérique existante. Attachez la stratégie aux certificats associés à vos appareils et ajoutez les autorisations suivantes à la stratégie de périphérique.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {   "Effect": "Allow",
            "Action": [ "iot:Connect" ],
            "Resource": [
                "arn:partition:iot:region:accountID:client/
${iot:Connection.Thing.ThingName}"
            ]
        }
        {
            "Effect": "Allow",
            "Action": [ "iot:Receive", "iot:Publish" ],
            "Resource": [
                "arn:partition:iot:region:accountID:topic/$aws/things/
${iot:Connection.Thing.ThingName}/streams/*"
```

```
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": [
                "arn:partition:iot:region:accountID:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/streams/*"
            ]
        }
    ]
}
```

Connect de vos appareils àAWS IoT

Périphériques qui utilisentAWS IoTLa remise de fichiers basée sur MQT est nécessaire pour se connecter àAWS IoT.AWS IoT La livraison de fichiers basée sur MQT s'intègre àAWS IoTdans leAWSCloud, de sorte que vos appareils doivent se connecter directement àle point de terminaison duAWS IoTPlan de données.

> Note
>
> Point de terminaison de la méthodeAWS IoTest spécifique au plan de données Compte AWS et de la région. Vous devez utiliser le point de terminaison pour le Compte AWS et la région dans laquelle vos appareils sont enregistrés dansAWS IoT.

Pour plus d'informations, consultez Connexion à AWS IoT Core (p. 67).

UtiliserAWS IoTLivraison de fichiers basée sur MQT dans les périphériques

Pour lancer le processus de transfert de données, un périphérique doit recevoir unJeu de données initial, qui inclut au minimum un ID de flux. Vous pouvez utiliser unJobs (p. 595)pour planifier des tâches de transfert de données pour vos appareils en incluant le jeu de données initial dans le document de travail. Lorsqu'un périphérique reçoit l'ensemble de données initial, il doit alors commencer l'interaction avecAWS IoTLivraison de fichiers basée sur MQT. Pour échanger des données avecAWS IoTRemise de fichiers basée sur MQT, un périphérique doit :

- Utilisez le protocole MQTT pour vous abonner auRubriques de remise de fichiers basées sur MQT (p. 105).
- Envoyez des demandes, puis attendez de recevoir les réponses à l'aide des messages MQTT.

Vous pouvez éventuellement inclure un ID de fichier de flux et une version de flux dans l'ensemble de données initial. L'envoi d'un ID de fichier de flux vers un périphérique peut simplifier la programmation du micrologiciel/logiciel de l'appareil, car il élimine le besoin de faire un`DescribeStream`à partir de l'appareil pour obtenir cet ID. Le périphérique peut spécifier la version du flux dans un`GetStream`pour appliquer une vérification de cohérence au cas où le flux a été mis à jour de façon inattendue.

Utilisez DescribeStream pour obtenir des données de flux

AWS IoTLa remise de fichiers basée sur MQT fournit le`DescribeStream`pour envoyer des données de flux vers un périphérique. Les données de flux renvoyées par cette API incluent l'ID de flux, la version du

flux, la description du flux et une liste de fichiers de flux, dont chacun a un ID de fichier et la taille du fichier en octets. Avec ces informations, un périphérique peut sélectionner des fichiers arbitraires pour lancer le processus de transfert de données.

> **Note**
>
> Vous n'avez pas besoin d'utiliser la méthode`DescribeStream`si votre appareil reçoit tous les ID de fichier de flux requis dans le jeu de données initial.

Procédez comme suit :`DescribeStream`de la demande.

1. S'abonner au filtre de rubrique « accepté »`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/description/json`.
2. S'abonner au filtre de rubrique « rejeté »`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/rejected/json`.
3. Publiez un`DescribeStream`En envoyant un message à`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/describe/json`.
4. Si la demande a été acceptée, votre appareil reçoit un`DescribeStream`réponse sur le filtre de rubrique « accepté ».
5. Si la demande a été rejetée, votre appareil reçoit la réponse d'erreur sur le filtre de rubrique « rejeté ».

> **Note**
>
> Si vous remplacez`json`avec`cbor`dans les rubriques et les filtres de rubrique affichés, votre appareil reçoit des messages au format CBOR, qui est plus compact que JSON.

Requête DescribeStream

Un type`DescribeStream`En JSON se présente comme suit.

```
{
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039"
}
```

- (Facultatif) »`c`« est le champ de jeton client.

  Le jeton client ne peut pas dépasser 64 octets. Un jeton client d'une longueur supérieure à 64 octets génère une réponse d'erreur et un`InvalidRequest`Message d'erreur.

Réponse de DescribeStream

A`DescribeStream`Dans JSON se présente comme suit.

```
{
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039",
    "s": 1,
    "d": "This is the description of stream ABC.",
    "r": [
        {
            "f": 0,
            "z": 131072
        },
        {
            "f": 1,
            "z": 51200
```

```
        }
    ]
}
```

- "c« est le champ de jeton client. Ceci est retourné s'il a été donné dans le`DescribeStream`de la demande. Utilisez le jeton client pour associer la réponse à sa demande.
- "s« est la version du flux sous forme d'entier. Vous pouvez utiliser cette version pour effectuer une vérification de cohérence avec votre`GetStream`de la demande.
- "r« contient une liste des fichiers dans le flux.
  - "f« est l'ID du fichier de flux sous la forme d'un entier.
  - "z« est la taille du fichier de flux en nombre d'octets.
- "d« contient la description du flux.

Obtenir des blocs de données à partir d'un fichier de flux

Vous pouvez utiliser la méthode`GetStream`afin qu'un périphérique puisse recevoir des fichiers de flux dans de petits blocs de données, de sorte qu'il puisse être utilisé par les périphériques qui ont des contraintes sur le traitement de grandes tailles de blocs. Pour recevoir un fichier de données entier, un appareil peut avoir besoin d'envoyer ou de recevoir plusieurs demandes et réponses jusqu'à ce que tous les blocs de données soient reçus et traités.

Requête GetStream

Procédez comme suit :`GetStream`de la demande.

1. S'abonner au filtre de rubrique « accepté »`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/data/json`.
2. S'abonner au filtre de rubrique « rejeté »`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/rejected/json`.
3. Publiez un`GetStream`demande au sujet`$aws/things/`*`ThingName`*`/streams/`*`StreamId`*`/get/json`.
4. Si la demande a été acceptée, votre appareil recevra un ou plusieurs`GetStream`réponses sur le filtre de rubrique « accepté ». Chaque message de réponse contient des informations de base et une charge utile de données pour un seul bloc.
5. Répétez les étapes 3 et 4 pour recevoir tous les blocs de données. Vous devez répéter ces étapes si la quantité de données demandée est supérieure à 128 Ko. Vous devez programmer votre appareil pour utiliser plusieurs`GetStream`demande de recevoir toutes les données demandées.
6. Si la demande a été rejetée, votre appareil recevra la réponse d'erreur sur le filtre de rubrique « rejeté ».

   Note

   - Si vous remplacez « json » par « cbor » dans les rubriques et les filtres de rubrique affichés, votre appareil recevra des messages au format CBOR, qui est plus compact que JSON.
   - AWS IoTLa remise de fichiers basée sur MQT limite la taille d'un bloc à 128 Ko. Si vous effectuez une demande pour un bloc de plus de 128 Ko, la demande échouera.
   - Vous pouvez faire une demande pour plusieurs blocs dont la taille totale est supérieure à 128 Ko (par exemple, si vous faites une demande de 5 blocs de 32 Ko chacun pour un total de 160 Ko de données). Dans ce cas, la demande n'échoue pas, mais votre appareil doit

effectuer plusieurs requêtes afin de recevoir toutes les données demandées. Le service enverra des blocs supplémentaires au fur et à mesure que votre appareil effectue des demandes supplémentaires. Nous vous recommandons de continuer avec une nouvelle demande seulement une fois que la réponse précédente a été correctement reçue et traitée.

- Quelle que soit la taille totale des données demandées, vous devez programmer votre appareil pour lancer de nouvelles tentatives lorsque les blocs ne sont pas reçus ou ne sont pas reçus correctement.

Un type`GetStream`En JSON se présente comme suit.

```
{
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",
    "s": 1,
    "f": 0,
    "l": 4098,
    "o": 2,
    "n": 100,
    "b": "..."
}
```

- [facultatif] »`c`« est le champ de jeton client.

  La longueur du jeton client ne peut pas dépasser 64 octets. Un jeton client d'une longueur supérieure à 64 octets génère une réponse d'erreur et un`InvalidRequest`Message d'erreur.
- [facultatif] »`s`« est le champ de version du flux (un entier).

  La remise de fichiers basée sur MQT applique un contrôle de cohérence basé sur cette version demandée et la dernière version de flux dans le cloud. Si la version de flux envoyée à partir d'un périphérique dans un`GetStream`ne correspond pas à la dernière version de flux dans le cloud, le service envoie une réponse d'erreur et un`VersionMismatch`Message d'erreur. En règle générale, un périphérique reçoit la version de flux attendue (la plus récente) dans le jeu de données initial ou dans la réponse à`DescribeStream`.
- "`f`« est l'ID du fichier de flux (un entier compris entre 0 et 255).

  L'ID du fichier de flux est requis lorsque vous créez ou mettez à jour un flux à l'aide de la commandeAWS CLIou du kit SDK. Si un périphérique demande un fichier de flux avec un ID qui n'existe pas, le service envoie une réponse d'erreur et un`ResourceNotFound`Message d'erreur.
- "`l`« est la taille du bloc de données en octets (nombre entier compris entre 256 et 131 072).

  Reportez-vous àCréer un bitmap pour une requête GetStream (p. 786)pour obtenir des instructions sur l'utilisation des champs bitmap pour spécifier la partie du fichier de flux qui sera renvoyée dans la boîte de dialogue`GetStream`Réponse. Si un périphérique spécifie une taille de bloc hors plage, le service envoie une réponse d'erreur et un`BlockSizeOutOfBounds`Message d'erreur.
- [facultatif] »`o`« est le décalage du bloc dans le fichier stream (un entier compris entre 0 et 98 304).

  Reportez-vous àCréer un bitmap pour une requête GetStream (p. 786)pour obtenir des instructions sur l'utilisation des champs bitmap pour spécifier la partie du fichier de flux qui sera renvoyée dans la boîte de dialogue`GetStream`Réponse. La valeur maximale de 98 304 est basée sur une limite de taille de fichier de flux de 24 Mo et 256 octets pour la taille de bloc minimale. Si elle n'est pas spécifiée, la valeur par défaut est 0.
- [facultatif] »`n`« est le nombre de blocs demandés (nombre entier compris entre 0 et 98 304).

  Le champ « n » spécifie soit (1) le nombre de blocs demandés, soit (2) lorsque le champ bitmap (« b ») est utilisé, une limite sur le nombre de blocs qui seront renvoyés par la requête bitmap. Cette seconde utilisation est facultative. S'il n'est pas défini, il est par défaut 131072/*`DatablockSize`*.
- [facultatif] »`b`« est un bitmap qui représente les blocs demandés.

À l'aide d'un bitmap, votre appareil peut demander des blocs non consécutifs, ce qui rend la gestion des tentatives suite à une erreur plus pratique. Reportez-vous àCréer un bitmap pour une requête GetStream (p. 786)pour obtenir des instructions sur l'utilisation des champs bitmap pour spécifier la partie du fichier de flux qui sera renvoyée dans la boîte de dialogue`GetStream`Réponse. Pour ce champ, convertissez le bitmap en chaîne représentant la valeur du bitmap en notation hexadécimale. Le bitmap doit être inférieur à 12 288 octets.

Réponse GetStream

A`GetStream`dans JSON ressemble à cet exemple pour chaque bloc de données demandé.

```
{
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",
    "f": 0,
    "l": 4098,
    "i": 2,
    "p": "..."
}
```

- "`c`« est le champ de jeton client. Ceci est retourné s'il a été donné dans le`GetStream`de la demande. Utilisez le jeton client pour associer la réponse à sa demande.
- "`f`« est l'ID du fichier de flux auquel appartient la charge utile du bloc de données en cours.
- "`l`« est la taille de la charge utile du bloc de données en octets.
- "`i`« est l'ID du bloc de données contenu dans la charge utile. Les blocs de données sont numérotés à partir de 0.
- "`p`« contient la charge utile du bloc de données. Ce champ est une chaîne qui représente la valeur du bloc de données en notation hexadécimale.

Créer un bitmap pour une requête GetStream

Vous pouvez utiliser le champ bitmap (`b`) dans un`GetStream`pour obtenir des blocs non consécutifs à partir d'un fichier de flux. Cela permet aux appareils disposant d'une capacité de mémoire vive limitée de résoudre les problèmes de diffusion réseau. Un périphérique peut demander uniquement les blocs qui n'ont pas été reçus ou qui n'ont pas été reçus correctement. Le bitmap détermine quels blocs du fichier de flux seront renvoyés. Pour chaque bit qui est défini sur 1 dans le bitmap, un bloc correspondant du fichier stream sera retourné.

Voici un exemple de spécification d'un bitmap et de ses champs de support dans un`GetStream`de la demande. Par exemple, vous souhaitez recevoir un fichier de flux en morceaux de 256 octets (la taille du bloc). Pensez à chaque bloc de 256 octets comme ayant un nombre qui spécifie sa position dans le fichier, à partir de 0. Le bloc 0 correspond au premier bloc de 256 octets dans le fichier, le bloc 1 au second, et ainsi de suite. Vous voulez demander les blocs 20, 21, 24 et 43 du fichier.

Décalage de bloc

Étant donné que le premier bloc est le numéro 20, spécifiez le décalage (champ`o`) comme 20 pour économiser de l'espace dans le bitmap.

Nombre de blocs

Pour vous assurer que votre appareil ne reçoit pas plus de blocs qu'il ne peut gérer avec des ressources mémoire limitées, vous pouvez spécifier le nombre maximal de blocs qui doivent être retournés dans chaque message envoyé par la remise de fichiers basée sur MQT. Notez que cette valeur n'est pas prise en compte si le bitmap lui-même spécifie moins que ce nombre de blocs, ou si la

taille totale des messages de réponse envoyés par la remise de fichiers basée sur MQT est supérieure à la limite de service de 128 Ko par`GetStream`de la demande.

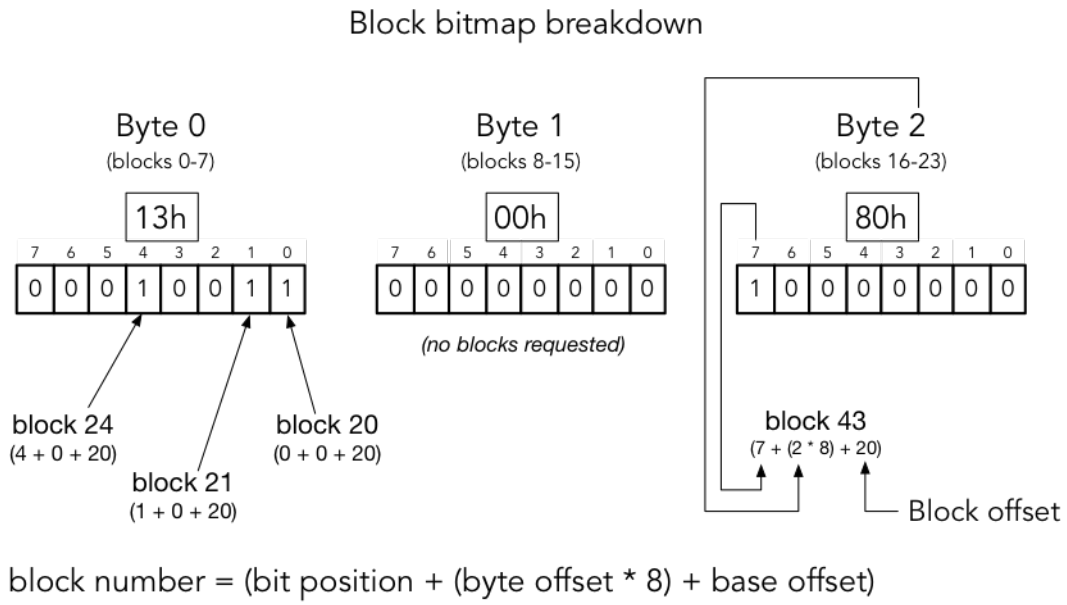
Bloc le bitmap

Le bitmap lui-même est un tableau d'octets non signés exprimés en notation hexadécimale, et inclus dans le fichier`GetStream`Demande sous forme d'une représentation sous forme de chaîne du nombre. Mais pour construire cette chaîne, commençons par penser au bitmap comme une longue séquence de bits (un nombre binaire). Si un bit dans cette séquence est défini sur 1, le bloc correspondant du fichier flux sera renvoyé au périphérique. Pour notre exemple, nous voulons recevoir les blocs 20, 21, 24 et 43, nous devons donc définir les bits 20, 21, 24 et 43 dans notre bitmap. Nous pouvons utiliser le décalage de bloc pour économiser de l'espace, donc après avoir soustrait le décalage de chaque numéro de bloc, nous voulons définir les bits 0, 1, 4 et 23, comme l'exemple suivant.

```
1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

En prenant un octet (8 bits) à la fois, cela est écrit conventionnellement comme : « 0b00010011", « 0b000000" et « 0b10000000". Le bit 0 apparaît dans notre représentation binaire à la fin du premier octet et le bit 23 au début du dernier. Cela peut être déroutant à moins que vous ne connaissiez les conventions. Le premier octet contient des bits 7-0 (dans cet ordre), le second octet contient des bits 15-8, le troisième octet contient des bits 23-16, et ainsi de suite. En notation hexadécimale, cela convertit en « 0x130080".

> **Tip**
>
> Vous pouvez convertir le binaire standard en notation hexadécimale. Prenez quatre chiffres binaires à la fois et convertissez-les en leur équivalent hexadécimal. Par exemple, « 0001" devient « 1", « 0011" devient « 3" et ainsi de suite.



En mettant tout cela ensemble, le JSON pour notre`GetStream`La demande se présente comme suit.

```
{
    "c" : "1",       // client token
    "s" : 1,         // expected stream version
    "l" : 256,       // block size
    "f" : 1,         // source file index id
    "o" : 20,        // block offset
```

```
    "n" : 32,        // number of blocks
    "b" : "0x130080" // A missing blockId bitmap starting from the offset
}
```

Gestion des erreurs deAWS IoTLivraison de fichiers basée sur MQT

Une réponse d'erreur envoyée à un périphérique pour`DescribeStream`and`GetStream`Les API contiennent un jeton client, un code d'erreur et un message d'erreur. Une réponse d'erreur type se présente comme suit.

```
{
    "o": "BlockSizeOutOfBounds",
    "m": "The block size is out of bounds",
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380"
}
```

- "o« correspond au code d'erreur qui indique la raison pour laquelle une erreur s'est produite. Pour plus d'informations, consultez les codes d'erreur plus loin dans cette section.
- "`m`« est le message d'erreur qui contient les détails de l'erreur.
- "c« est le champ de jeton client. Cela peut être retourné s'il a été donné dans le`DescribeStream`de la demande. Vous pouvez utiliser le jeton client pour associer la réponse à sa demande.

  Le champ de jeton client n'est pas toujours inclus dans une réponse d'erreur. Lorsque le jeton client donné dans la requête n'est pas valide ou est mal formé, il n'est pas renvoyé dans la réponse d'erreur.

  > **Note**
  >
  > Pour des raisons de compatibilité ascendante, les champs de la réponse d'erreur peuvent être sous forme non abrégée. Par exemple, le code d'erreur peut être désigné par des champs « code » ou « o » et le champ de jeton client peut être désigné par des champs « clientToken » ou « c ». Nous vous recommandons d'utiliser le formulaire d'abréviation ci-dessus.

InvalidTopic

La rubrique MQTT du flux n'est pas valide.

InvalidJson

La demande Stream n'est pas un document JSON valide.

InvalidCbor

La demande de flux n'est pas un document CBOR valide.

InvalidRequest

La demande est généralement identifiée comme étant mal formée. Pour plus d'informations, consultez le message d'erreur.

Non autorisé

La demande n'est pas autorisée à accéder aux fichiers de données de flux sur le support de stockage, tels qu'Amazon S3. Pour plus d'informations, consultez le message d'erreur.

BlockSizeOutOfBounds

La taille du bloc est hors limites. Reportez-vous à la section »Livraison de fichiers basée sur MQTdans AWS IoT Core Quotas de service.

OffsetOutOfBounds

Le décalage est hors limites. Reportez-vous à la section »Livraison de fichiers basée sur MQTTdans AWS IoT Core Quotas de service.

BlockCountLimitExcédé

Le nombre de blocs de requête est hors limites. Reportez-vous à la section »Livraison de fichiers basée sur MQTTdans AWS IoT Core Quotas de service.

BlockBitmapLimitdépassée

La taille du bitmap de la requête est hors limites. Reportez-vous à la section »Livraison de fichiers basée sur MQTTdans AWS IoT Core Quotas de service.

ResourceNotFound

Le flux demandé, les fichiers, les versions de fichiers ou les blocs n'ont pas été trouvés. Pour plus d'informations, consultez le message d'erreur.

VersionMismatch

La version du flux dans la requête ne correspond pas à la version du flux dans la fonctionnalité de remise de fichiers basée sur MQT. Cela indique que les données du flux ont été modifiées depuis que la version du flux a été initialement reçue par le périphérique.

EtagMismatch

L'ETag S3 dans le flux ne correspond pas à l'ETag de la dernière version d'objet S3.

InternalError

Une erreur interne s'est produite lors de la remise de fichiers basée sur MQT.

Un exemple de cas d'utilisation dans FreeRTOS OTA

L'agent OTA (OTA) de FreeRTOS utiliseAWS IoTLivraison de fichiers basée sur MQT pour transférer les images du firmware FreeRTOS vers les appareils FreeRTOS. Pour envoyer le jeu de données initial à un périphérique, il utilise la méthodeAWS IoTService de Job pour planifier une tâche de mise à jour OTA sur les appareils FreeRTOS.

Pour une implémentation de référence d'un client de remise de fichiers basé sur MQT, voirCodes de l'agent FreeRTOS OTAdans la documentation FreeRTOS.

AWS IoT Device Defender

AWS IoT Device Defender est un service de sécurité qui vous permet de vérifier la configuration de vos appareils et de surveiller les appareils connectés afin de détecter les comportements anormaux et d'atténuer les risques liés à la sécurité. Il vous offre la possibilité d'appliquer des stratégies de sécurité cohérentes pour l'ensemble de votre parc d'appareils AWS IoT et de réagir rapidement lorsque des appareils sont menacés.

Les parcs IoT sont composés d'un grand nombre d'appareils disposant de capacités diverses, d'une durée de vie longue et qui sont répartis géographiquement. Ces caractéristiques peuvent rendre la configuration des parcs complexe et source d'erreurs. Les appareils étant souvent limités en puissance de calcul, de mémoire et de capacités de stockage, l'utilisation du chiffrement et d'autres formes de sécurité sur les appareils eux-mêmes s'en trouve limitée. En outre, les appareils utilisent souvent des logiciels aux vulnérabilités connues. Ces facteurs font des parcs IoT une cible attractive pour les pirates informatiques et rend difficile la sécurisation de votre parc sur une base permanente.

AWS IoT Device Defenderrésout ces problèmes en fournissant des outils pour identifier les problèmes de sécurité, ainsi que les écarts par rapport aux bonnes pratiques.AWS IoT Device Defenderpeuvent auditer les parcs d'appareils afin de s'assurer qu'ils respectent les bonnes pratiques en matière de sécurité et de détecter les comportements anormaux sur les appareils.

Formation et certification AWS

Suivez le cours suivant pour démarrer avecAWS IoT Device Defender :AWS IoT Device Defender Primer.

Mise en route avec AWS IoT Device Defender

Vous pouvez utiliser les didacticiels suivants pour travailler avec AWS IoT Device Defender.

Rubriques

Configuration de

Avant d'utiliser AWS IoT Device Defender pour la première fois, exécutez les tâches suivantes :

Ces tâches créent un Compte AWS et un utilisateur IAM avec des privilèges d'administrateur pour le compte.

Inscrivez-vous à AWS

Lorsque vous vous inscrivez à AWS, votre compte est automatiquement inscrit pour tous les services dans AWS, y compris AWS IoT Device Defender. Si vous disposez d'un Compte AWS Déjà, passez à la prochaine étape. Si vous n'avez pas de Compte AWS , observez la procédure suivante pour en créer un.

Si vous n'avez pas de compte AWS, complétez les étapes suivantes pour en créer un.

Pour créer un compte AWS

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

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

Notez votre Compte AWS Numéro, car vous en aurez besoin pour la prochaine tâche.

Création d'un utilisateur IAM

Cette procédure décrit la façon de créer un utilisateur IAM pour vous-même et d'ajouter cet utilisateur à un groupe disposant des autorisations administratives d'une stratégie gérée attachée.

Pour créer un administrateur pour vous-même et ajouter l'utilisateur à un groupe d'administrateurs (console)

1. Connectez-vous à la console IAM en tant que propriétaire du compte en choisissant Utilisateur racine et en saisissant l'adresse e-mail de votre compte AWS. Sur la page suivante, saisissez votre mot de passe.

   Note

   Nous vous recommandons vivement de respecter la bonne pratique qui consiste à avoir recours à l'utilisateur IAM **`Administrator`** suivant et protéger les informations d'identification de l'utilisateur racine. Connectez-vous en tant qu'utilisateur racine pour effectuer certaines tâches de gestion des comptes et des services.
2. Dans le panneau de navigation, choisissez Utilisateurs, puis Add user (Ajouter un utilisateur).
3. Dans User name (Nom d'utilisateur), entrez **`Administrator`**.
4. Cochez la case à côté de AWS Management Console access (accès à la console). Puis, sélectionnez Custom password (Mot de passe personnalisé, et entrez votre nouveau mot de passe dans la zone de texte.
5. (Facultatif) Par défaut, AWS oblige le nouvel utilisateur à créer un nouveau mot de passe lors de sa première connexion. Décochez la case en regard de User must create a new password at next sign-in (L'utilisateur doit créer un nouveau mot de passe à sa prochaine connexion) pour autoriser le nouvel utilisateur à réinitialiser son mot de passe une fois qu'il s'est connecté.
6. Choisissez Next (Suivant) Permissions (Autorisations).
7. Sous Set permissions (Accorder des autorisations), choisissez Add user to group (Ajouter un utilisateur au groupe).
8. Choisissez Create group.
9. Dans la boîte de dialogue Create group (Créer un groupe), pour Group name (Nom du groupe), tapez **`Administrators`**.
10. Choisissez Filter policies (Filtrer les stratégies), puis sélectionnez AWS managed - job function (Fonction professionnelle gérée par AWS) pour filtrer le contenu de la table.
11. Dans la liste des stratégies, cochez la case AdministratorAccess. Choisissez ensuite Create group.

    Note

    Vous devez activer l'accès des rôles et utilisateurs IAM à la facturation avant de pouvoir utiliser les autorisations `AdministratorAccess` pour accéder à la console AWS Billing and

Cost Management. Pour ce faire, suivez les instructions de l'étape 1 du didacticiel portant sur comment déléguer l'accès à la console de facturation.

12. De retour dans la liste des groupes, activez la case à cocher du nouveau groupe. Choisissez Refresh si nécessaire pour afficher le groupe dans la liste.
13. Choisissez Next (Suivant) Tags (Balises).
14. (Facultatif) Ajoutez des métadonnées à l'utilisateur en associant les balises sous forme de paires clé-valeur. Pour de plus amples informations sur l'utilisation des balises dans IAM, veuillez consulter Balisage des utilisateurs et des rôles IAM dans le Guide de l'utilisateur IAM.
15. Choisissez Next (Suivant) VérificationPour afficher la liste des membres du groupe à ajouter au nouvel utilisateur. Une fois que vous êtes prêt à continuer, choisissez Create user.

Vous pouvez utiliser ce même processus pour créer d'autres groupes et utilisateurs et pour accorder à vos utilisateurs l'accès aux ressources de votre compte AWS. Pour en savoir plus sur l'utilisation des stratégies afin de limiter les autorisations d'accès des utilisateurs à certaines ressources AWS, consultez Gestion des accès et Exemples de stratégies.

Guide d'audit

Ce didacticiel fournit des instructions sur la configuration d'un audit récurrent, la configuration d'alarmes, l'examen des résultats d'audit et l'atténuation des problèmes d'audit.

Rubriques

Prerequisites

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

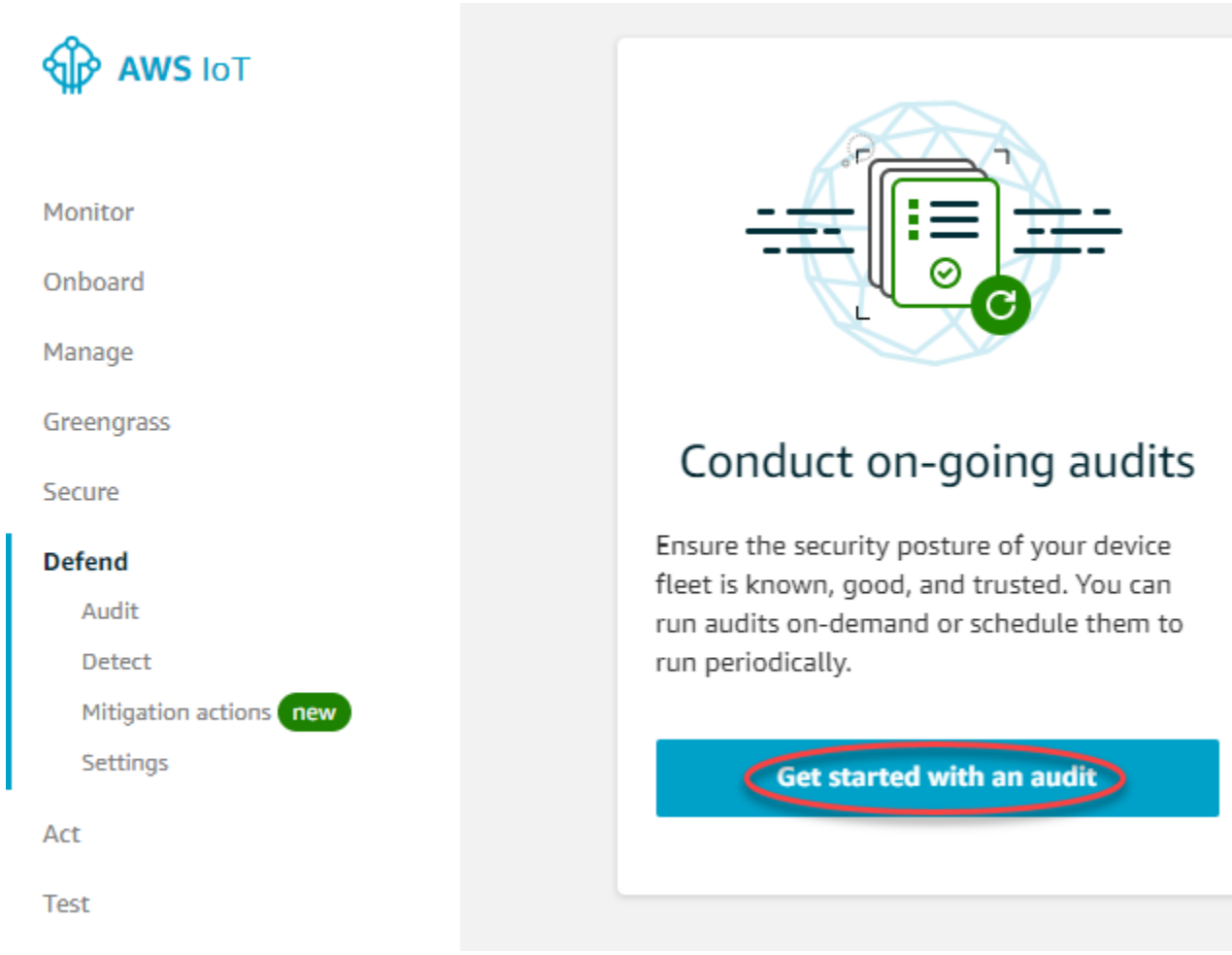
- Un Compte AWS . Si vous ne disposez pas de ces éléments, consultezConfiguration de.

Activation des contrôles d'audit

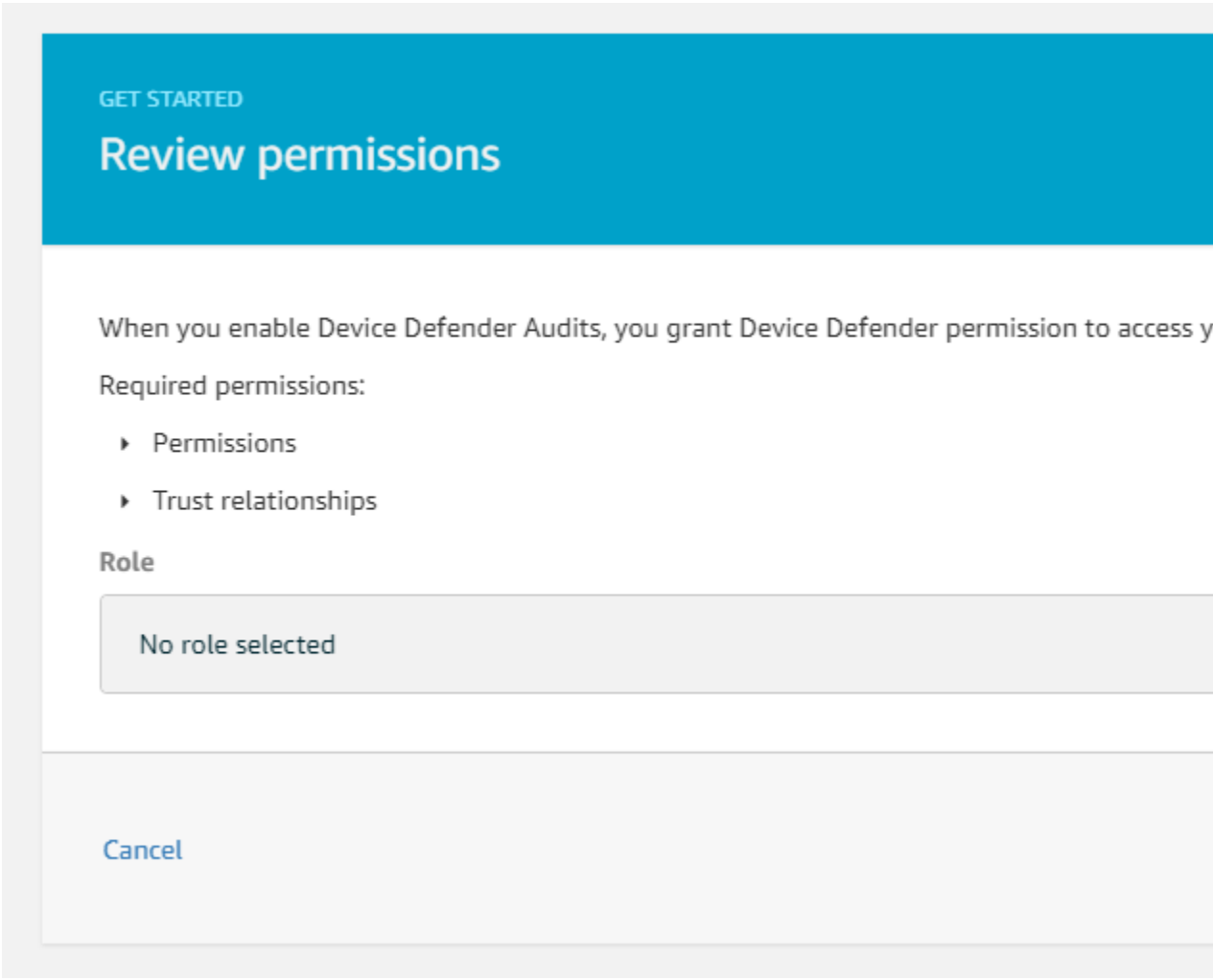
Dans la procédure suivante, vous activez les contrôles d'audit qui examinent les paramètres et les stratégies de compte et d'appareil afin de vérifier que les mesures de sécurité sont en place. Dans ce tutoriel, nous vous demandons d'activer toutes les vérifications d'audit, mais vous pouvez sélectionner les vérifications que vous souhaitez.

La tarification de l'audit est par nombre d'appareils et par mois (appareils de la flotte connectés àAWS IoT). Par conséquent, l'ajout ou la suppression de vérifications d'audit n'affecterait pas votre facture mensuelle lors de l'utilisation de cette fonctionnalité.

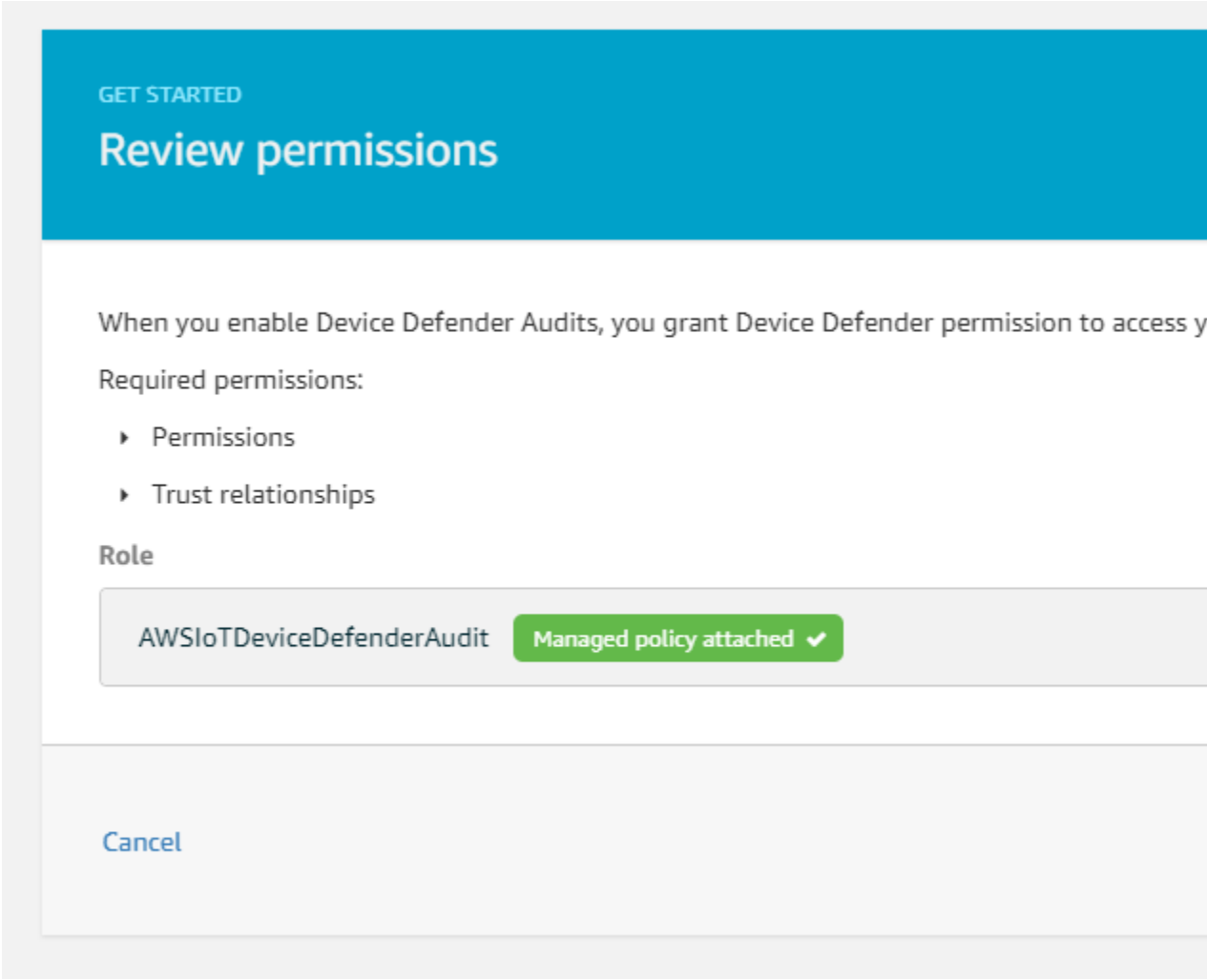
1. DansAWS IoTconsoleDans le volet de navigation, développezDéfendreet sélectionnezMise en route avec un audit.



2. La .Lancer l'audit Device Defenderdonne une vue d'ensemble des étapes requises pour activer les vérifications d'audit. Une fois que vous avez examiné l'écran, sélectionnezSuivant.
3. Si vous disposez déjà d'un rôle à utiliser, vous pouvez le sélectionner. Sinon, sélectionnezCréation d'un rôleet nommez-le*`AWSIOTDeviceDefenderAudit`*.

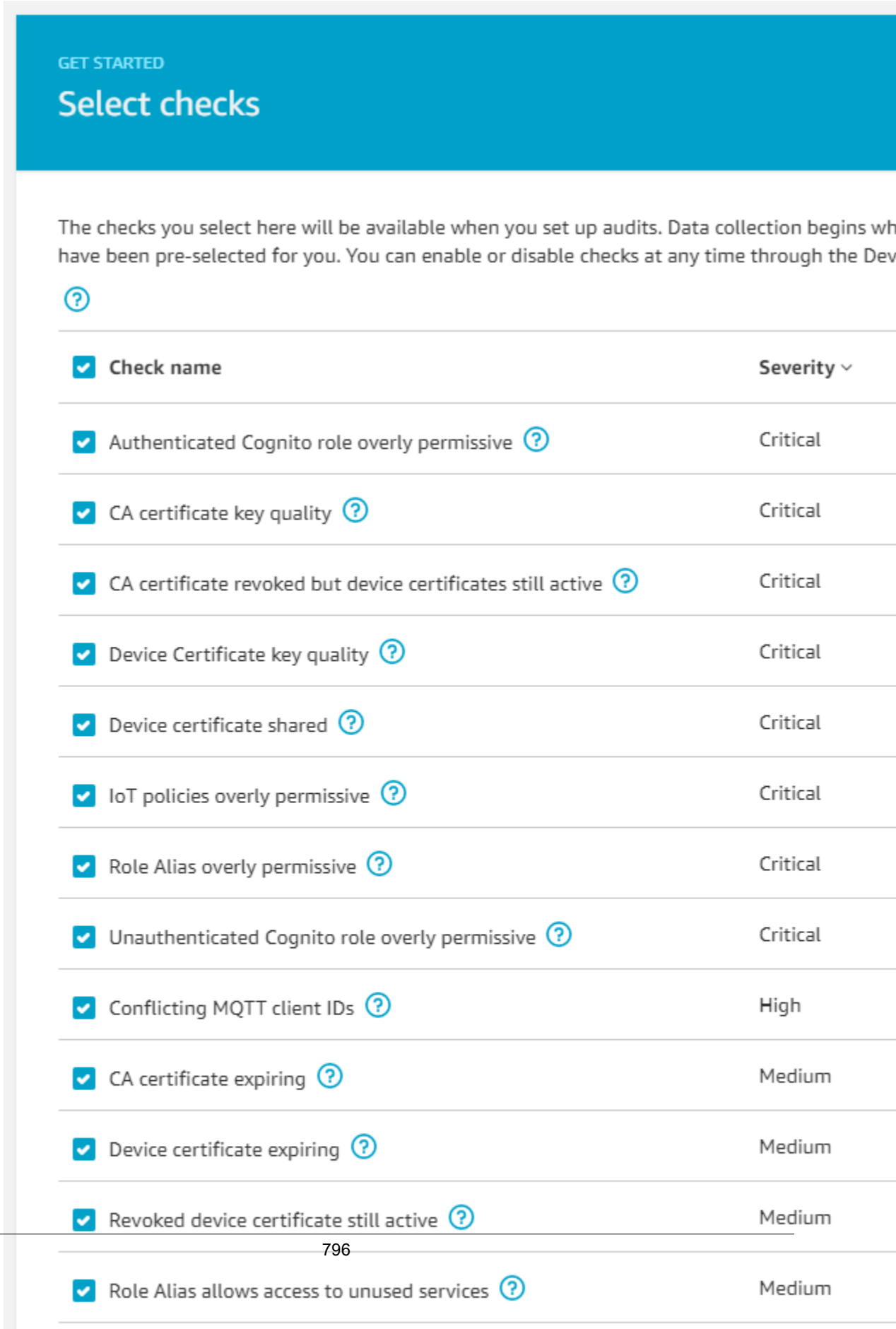


Vous devez voir les autorisations requises attachées automatiquement au rôle. Sélectionnez les triangles en regard deAutorisationsandRelations d'approbationpour voir quelles autorisations sont accordées. Tâche de sélectionSuivantLorsque vous êtes prêt à passer à autre chose.



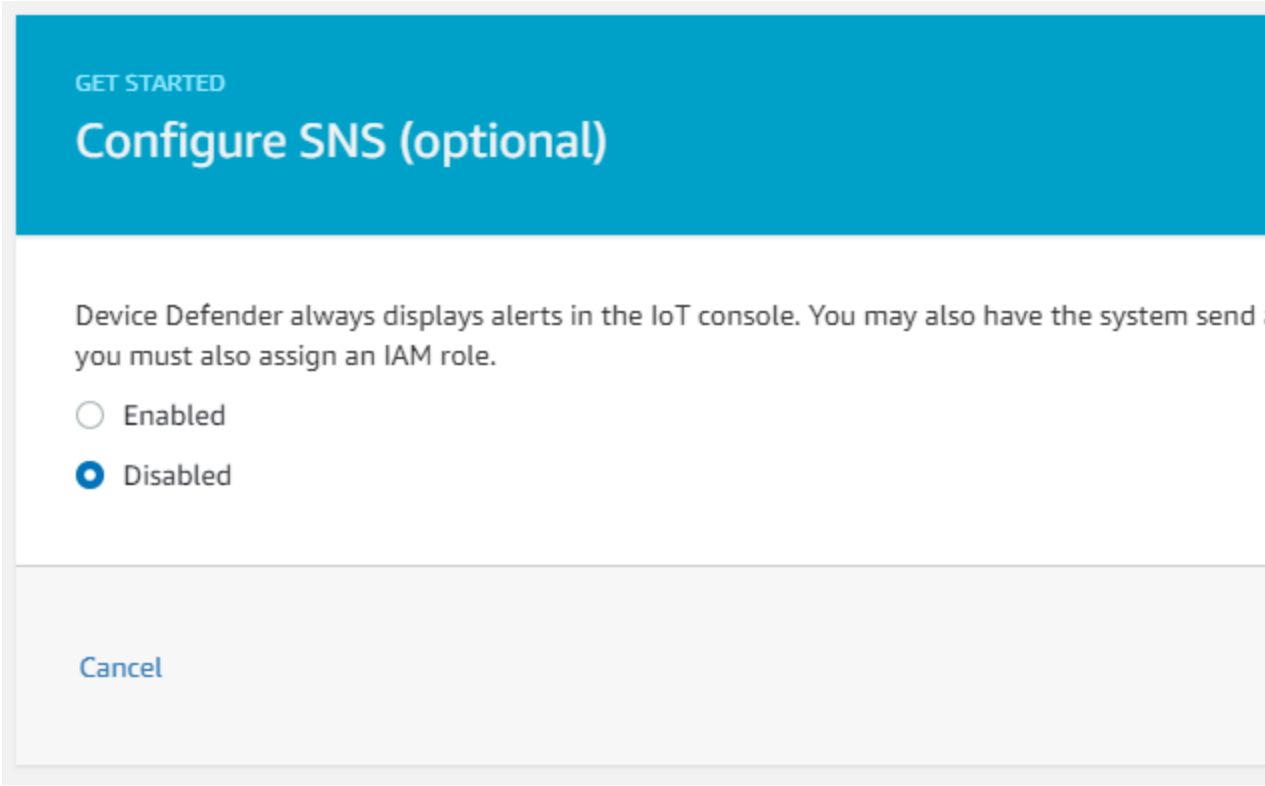
4. Dans la pageSélectionner les chèques, vous verrez toutes les vérifications d'audit que vous pouvez sélectionner. Pour ce tutoriel, nous vous demandons de sélectionner toutes les vérifications, mais vous pouvez sélectionner les vérifications que vous voulez. À côté de chaque vérification d'audit se trouve une icône d'aide qui décrit ce que fait la vérification d'audit. Pour plus d'informations sur les vérifications d'audit, consultezContrôles d'audit.

   Tâche de sélectionSuivantune fois que vous avez sélectionné vos chèques.

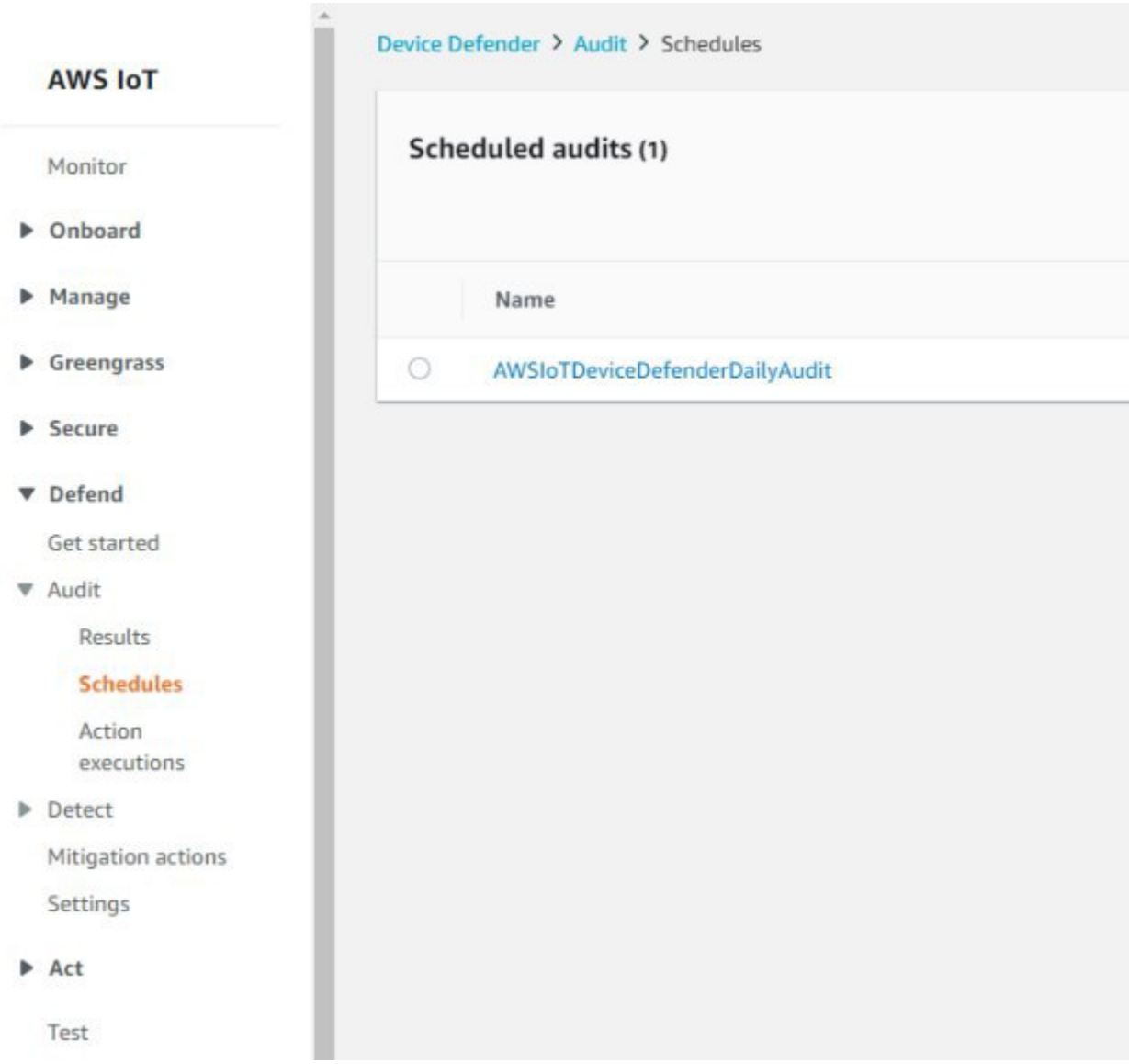

GET STARTED
Select checks
The checks you select here will be available when you set up audits. Data collection begins wh
have been pre-selected for you. You can enable or disable checks at any time through the Dev
Check name
Severity
Authenticated Cognito role overly permissive
Critical
CA certificate key quality
Critical
CA certificate revoked but device certificates still active
Critical
Device Certificate key quality
Critical
Device certificate shared
Critical
IoT policies overly permissive
Critical
Role Alias overly permissive
Critical
Unauthenticated Cognito role overly permissive
Critical
Conflicting MQTT client IDs
High
CA certificate expiring
Medium
Device certificate expiring
Medium
Revoked device certificate still active
Medium
Role Alias allows access to unused services
Medium

Vous pouvez toujours modifier vos vérifications d'audit configurées sousParamètres.

5. Dans la pageConfigurer SNS (facultatif)écran, sélectionnezActivez l'audit. Si vous souhaitez activer les notifications SNS, consultezActiver les notifications SNS (facultatif) (p. 804).



6. Vous serez redirigé versSchedulesUnderAudit.

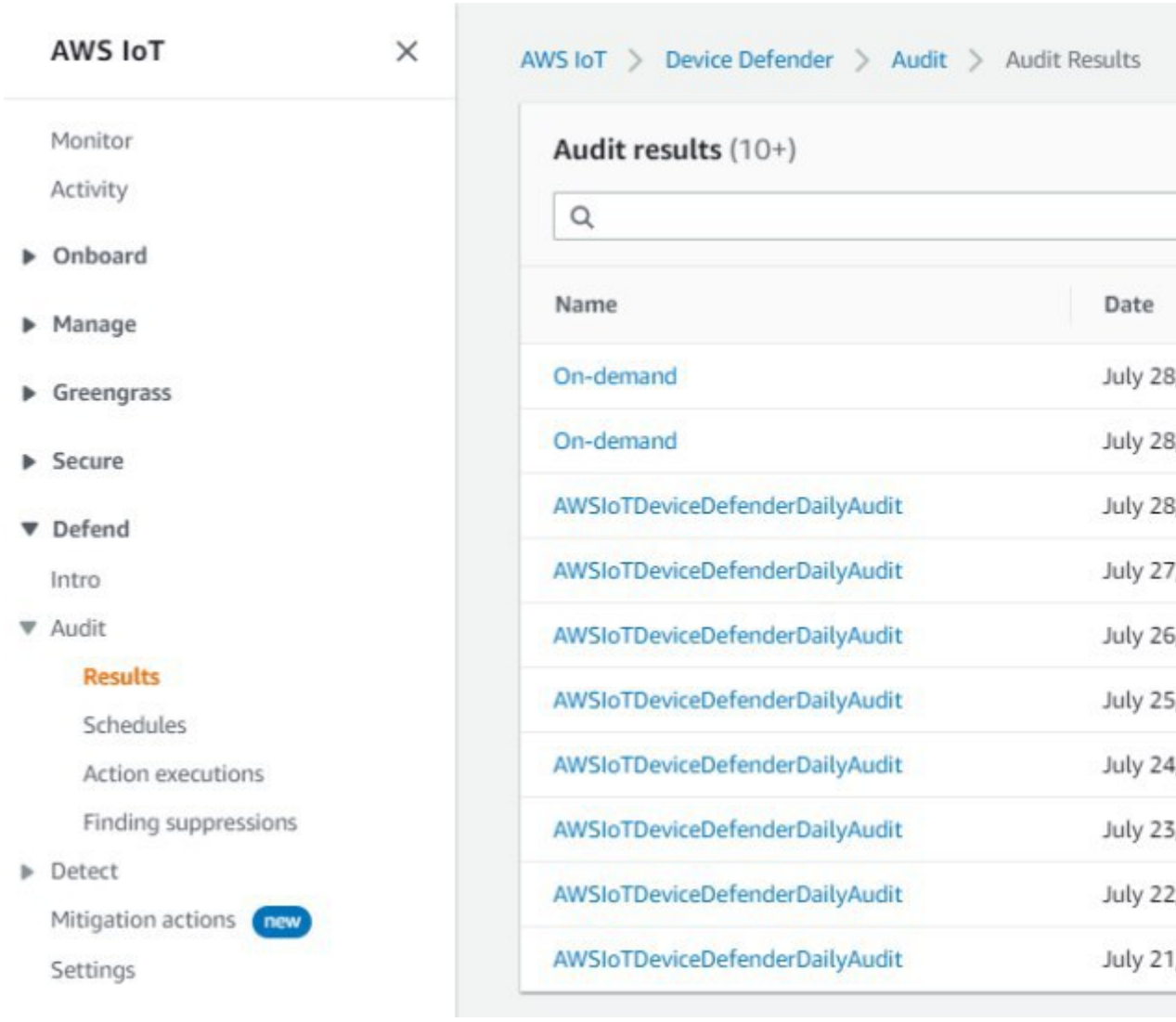


Afficher les résultats de l'audit

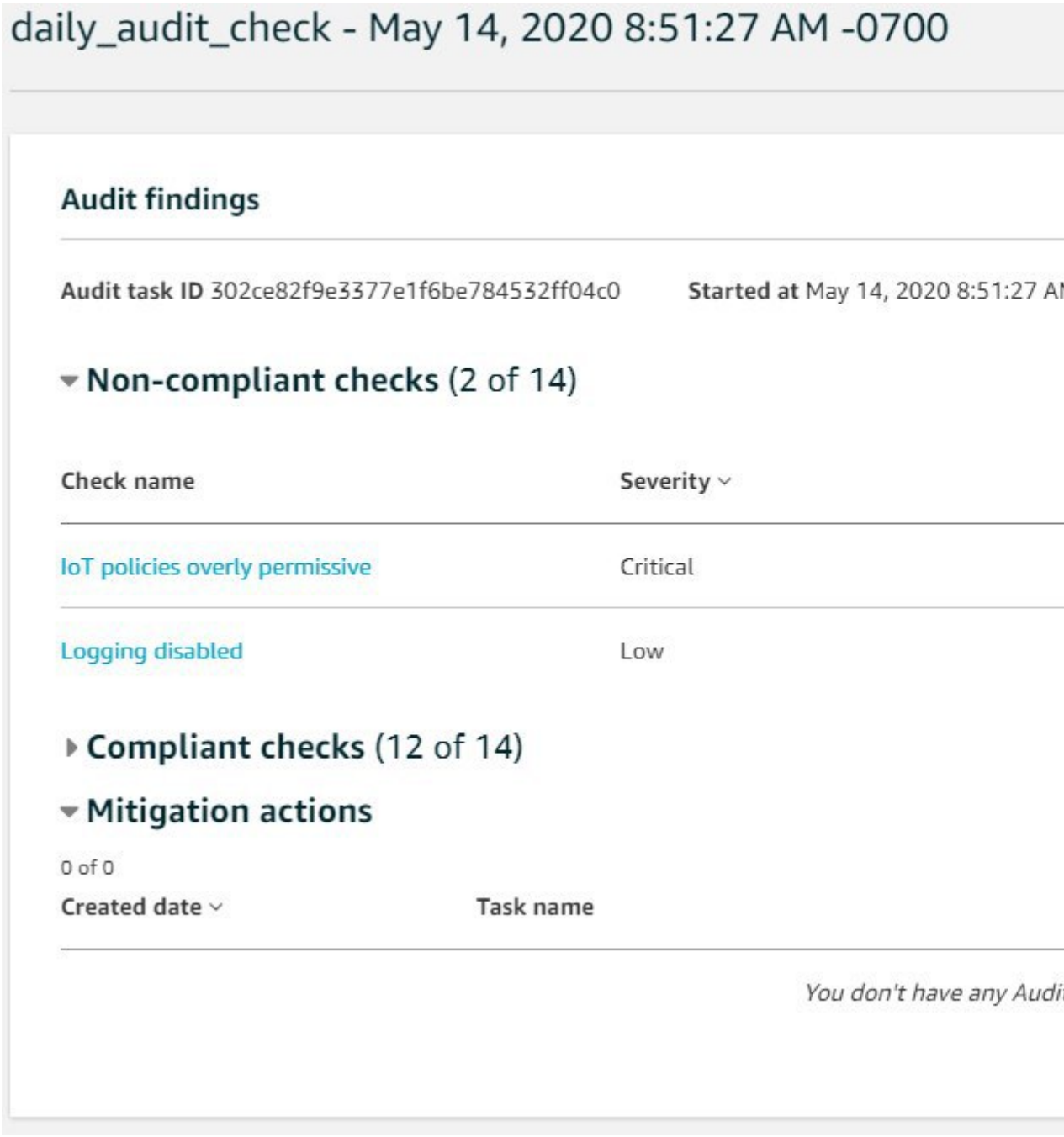
La procédure suivante vous montre comment afficher vos résultats d'audit. Dans ce didacticiel, vous voyez les résultats de l'audit des vérifications configurées dansActivation des contrôles d'audit (p. 792)Didacticiel de

Pour afficher les résultats de l'audit

1. DansAWS IoTconsoleDans le volet de navigation, développezDéfendre, sélectionnezAudit, puis sélectionnezRésultats.
2. La .Récapitulatifvous indiquera si vous avez des contrôles non conformes.



3. Sélectionnez laNomde la vérification d'audit que vous souhaitez examiner.



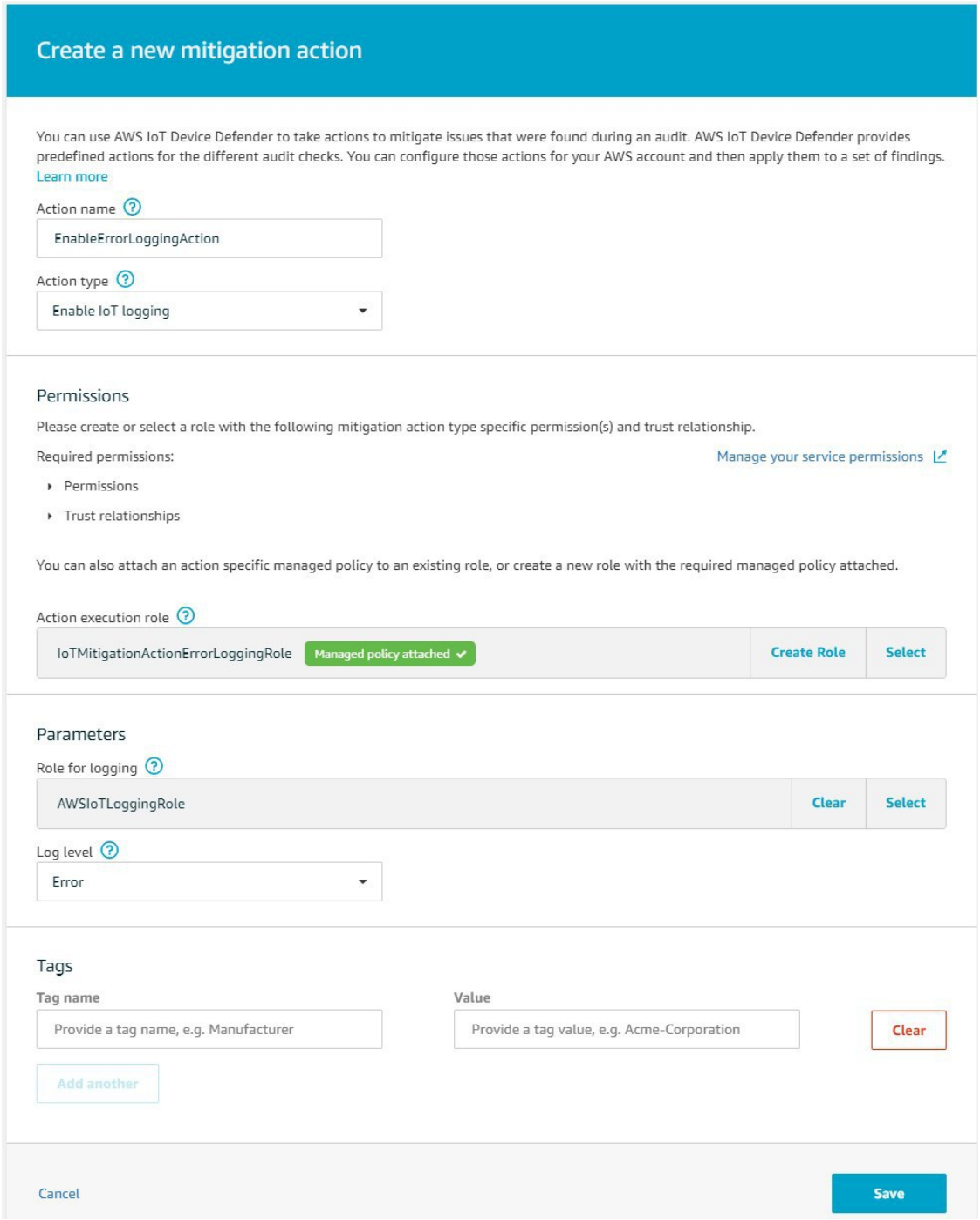
4. Utilisez les points d'interrogation pour obtenir des conseils sur la façon de rendre vos contrôles non conformes conformes. Par exemple, vous pouvez suivreActiver la journalisation (facultatif) (p. 807)pour rendre la vérification « Journalisation désactivée » conforme.

Création d'actions d'atténuation

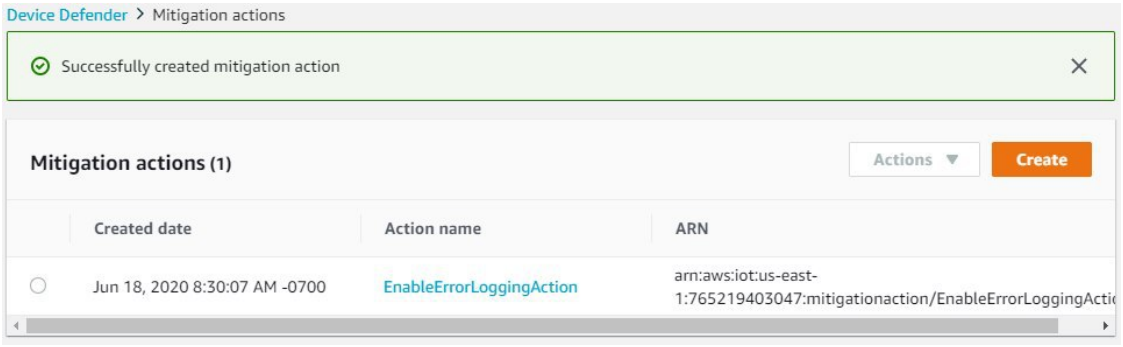
Dans la procédure suivante, vous allez créer unAWS IoT Device DefenderAction d'atténuation de l'audit pour activerAWS IoTjournalisation. Chaque vérification d'audit a défini des mesures d'atténuation qui affecteront lesType d'actionque vous choisissez pour la vérification d'audit que vous souhaitez corriger. Pour de plus amples informations, veuillez consulterActions d'atténuation.

Utiliser la console AWS IoT pour créer des actions d'atténuation

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Mitigation Actions (Actions d'atténuation).
3. Dans la page Mitigation Actions (Actions d'atténuation), choisissez Create (Créer).
4. Dans la pageCréez une action d'atténuation, dansAction name (Nom de l'action), saisissez un nom unique pour votre action d'atténuation tel que*`EnableErrorLoggingAction`*.
5. DansType d'action, choisissezActiver la IoT de l'.
6. DansRôle d'exécution d'action, sélectionnezCréation d'un rôle. PourNom, utilisez*`IotMitigationActionErrorLoggingRole`*. Ensuite, choisissezCréation d'un rôle.
7. DansParamètres, sousRôle pour la journalisation, sélectionnez`AWSIoTLoggingRole`. PourNiveau de journalisation, choisissez`Error`.



8. Choisissez Save (Enregistrer) pour enregistrer votre action d'atténuation pour votre compte AWS.
9. Une fois créé, vous verrez l'écran suivant indiquant que votre action d'atténuation a été créée avec succès.

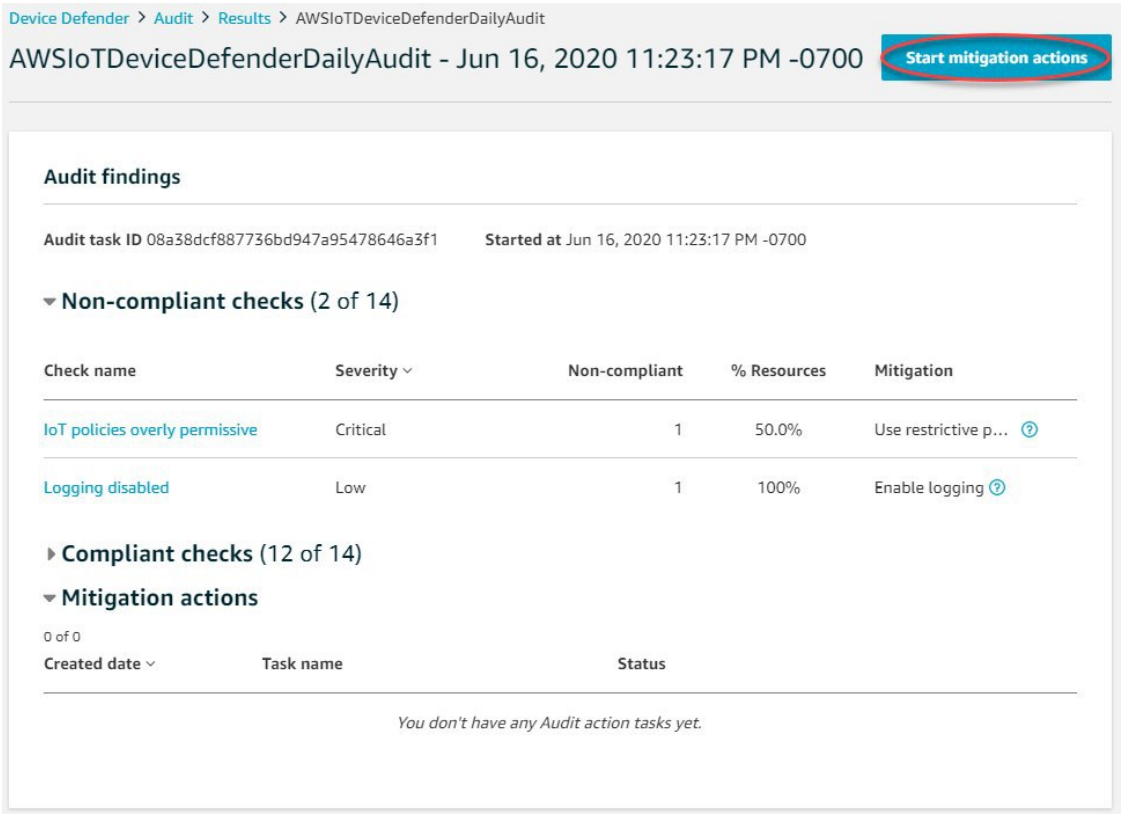


Appliquer des mesures d'atténuation aux constatations de votre audit

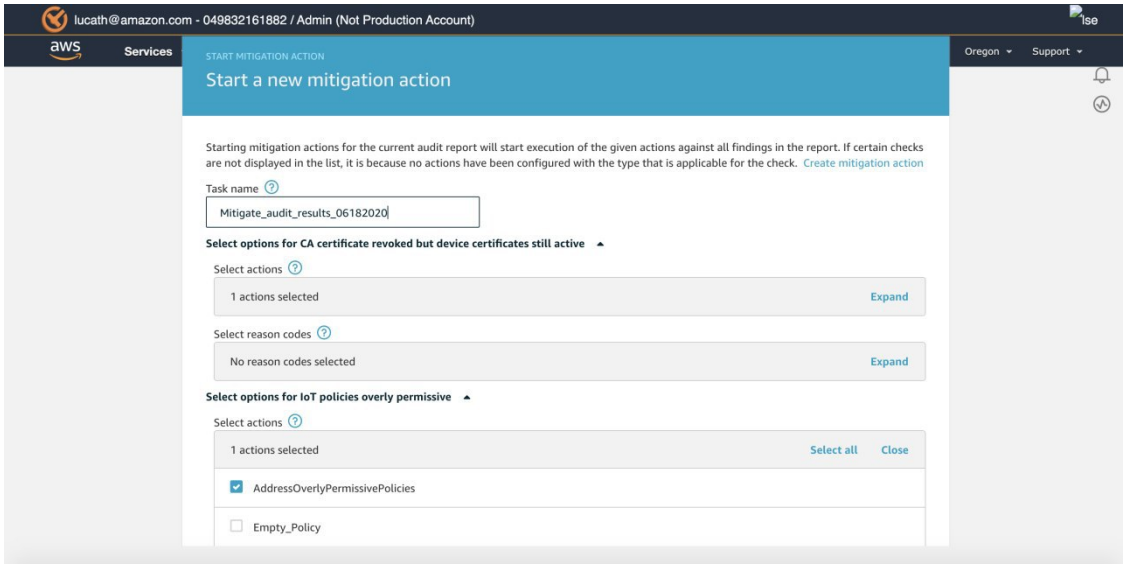
La procédure suivante vous montre comment appliquer des actions d'atténuation à vos résultats d'audit.

Pour atténuer les constatations d'audit non conformes

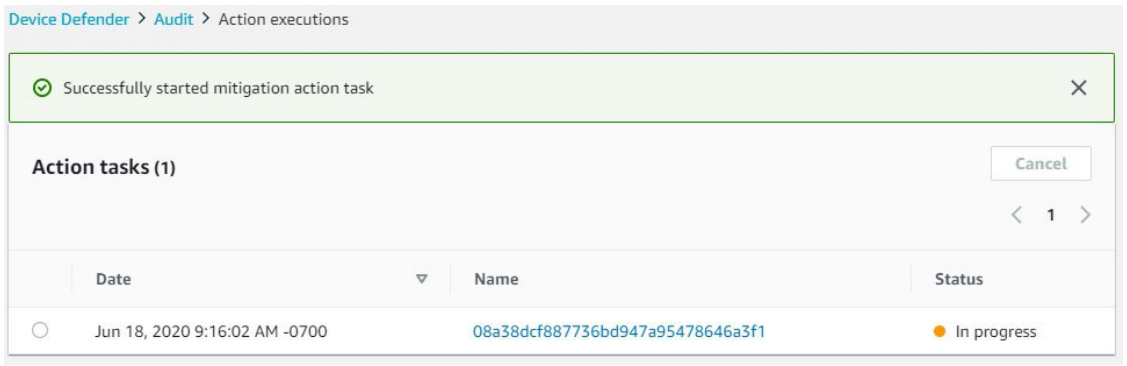
1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissezAudit, puis.Résultats. Sélectionnez le nom de l'audit auquel vous souhaitez répondre.
3. Vérifiez vos résultats. Notez que`Logging disabled`se trouve sousContrôles non conformes.
4. Tâche de sélectionDémarrer des actions d'atténuation.



5. UnderSélectionner des actions, sélectionnez les actions appropriées pour chaque constatation de non-conformité afin de résoudre les problèmes.



6. Tâche de sélectionConfirmer.
7. Une fois que l'action d'atténuation est lancée, la mise en œuvre peut prendre quelques minutes.



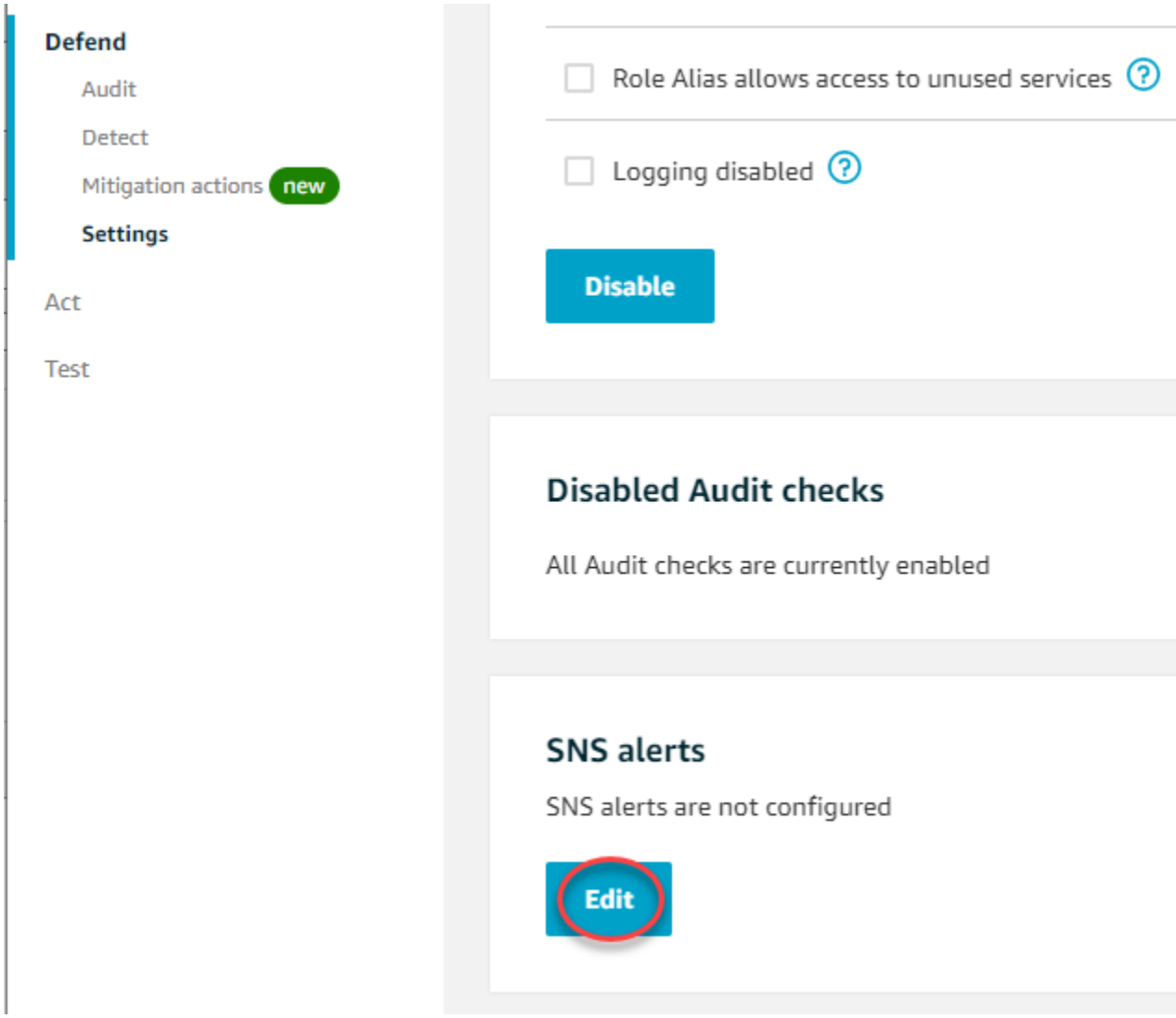
Pour vérifier que l'action d'atténuation a fonctionné

1. DansAWS, dans le volet de navigation, sélectionnezParamètres.
2. Confirmer queJournauxsont`Enabled`et l'Niveau de détailest`Error`.

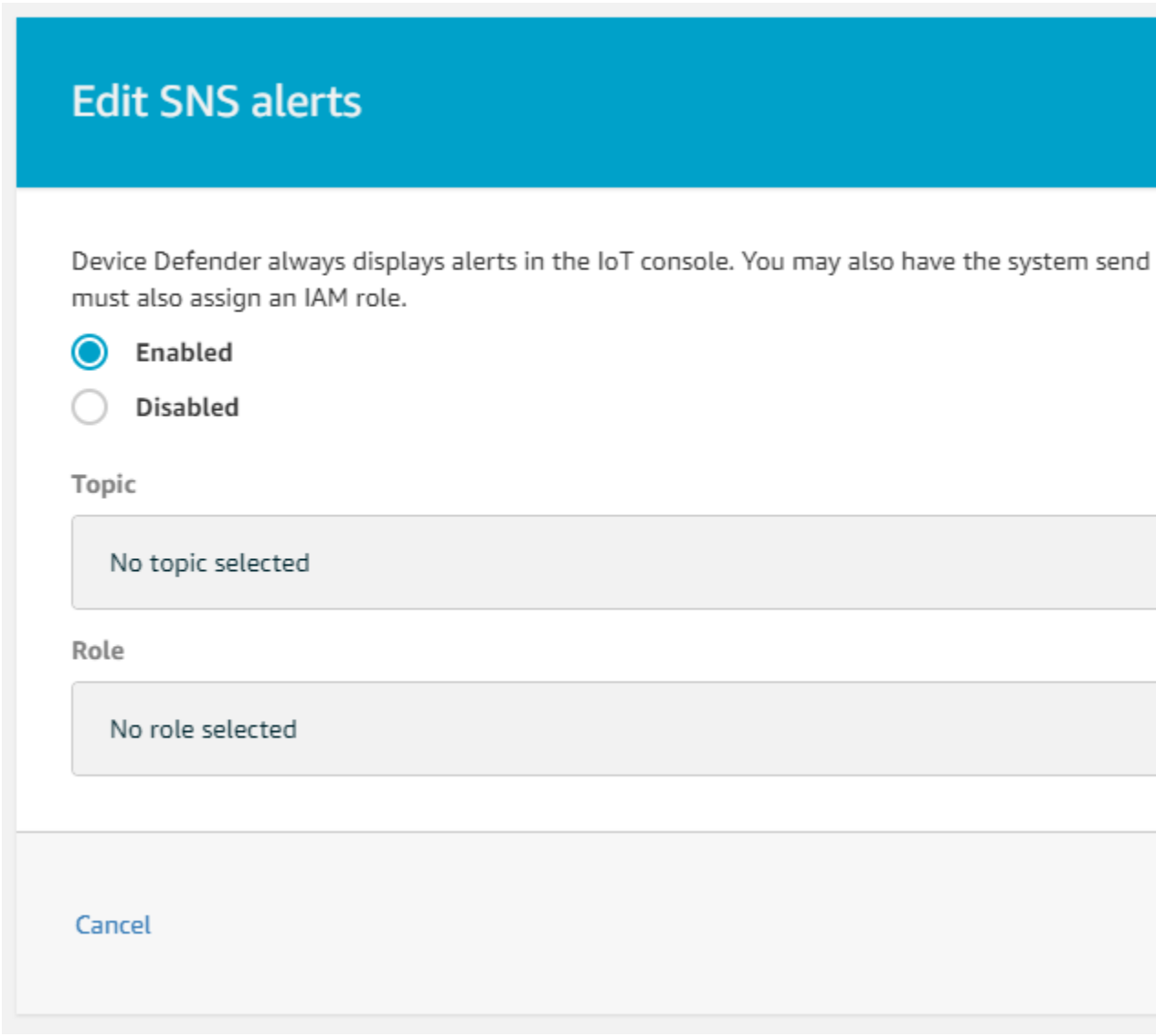
Activer les notifications SNS (facultatif)

Dans la procédure suivante, vous activez les notifications SNS (Simple Notifications Service) pour vous avertir lorsque vos audits identifient des ressources non conformes. Dans ce didacticiel, vous allez configurer des notifications pour les vérifications d'audit activées dans leActivation des contrôles d'audit (p. 792)Didacticiel de

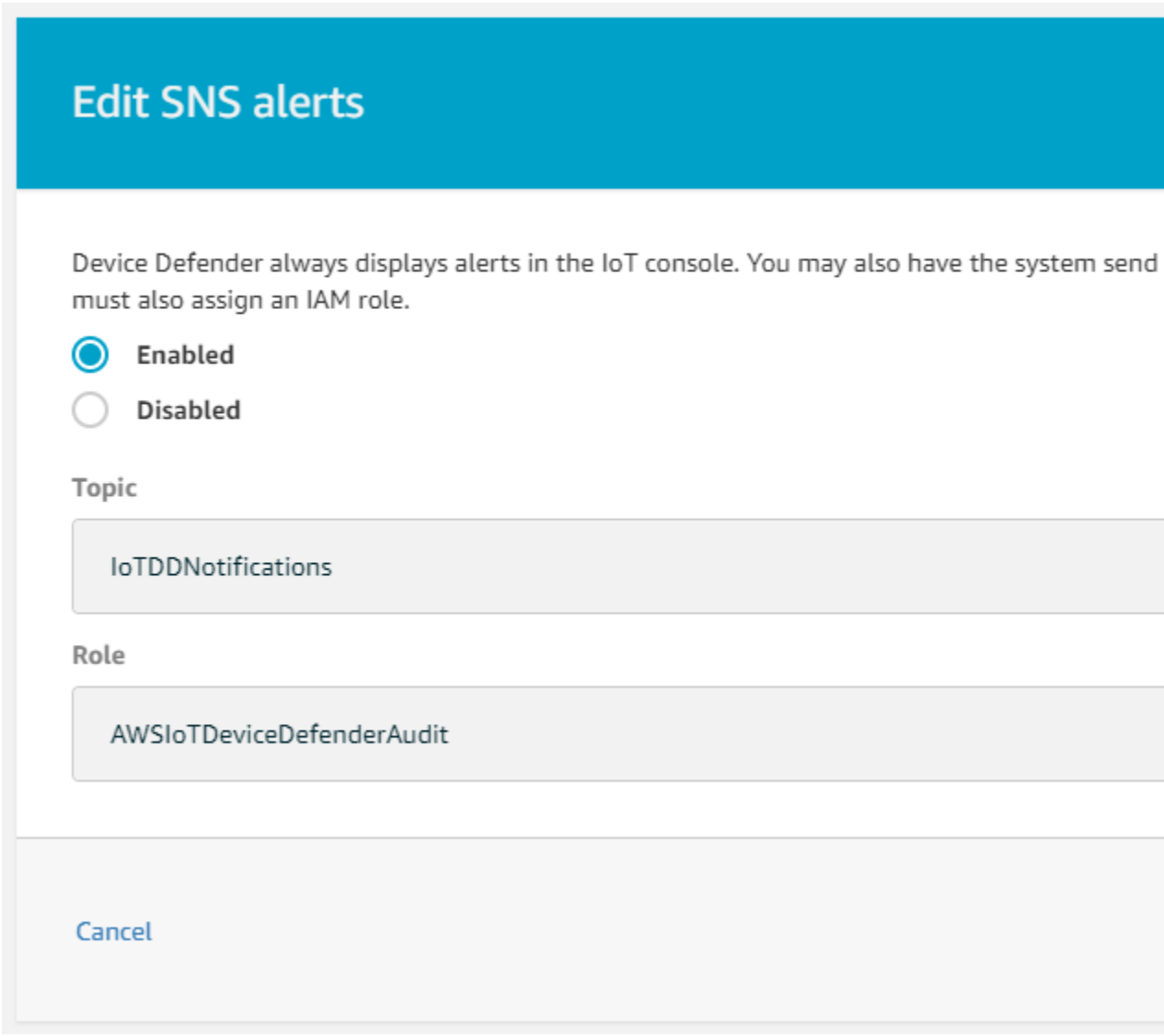
1. Tout d'abord, vous devez créer une stratégie IAM qui donne accès à Amazon SNS via l'outilAWSConsole de gestion. Pour ce faire, suivez l'interface de ligne de commandeCréation d'uneAWS IoT Device DefenderRôle d'audit IAM (facultatif) (p. 810), mais en sélectionnantAWSIOTDeviceDefenderPublishFindingStoSNMitigationActionÀ l'étape 8.
2. DansAWS IoTconsoleDans le volet de navigation, développezDéfendreet sélectionnezParamètres.
3. UnderAlertes SNS, sélectionnezModifier.



4. Dans la pageModifier les alertes SNSécran, sélectionnezActivé. UnderRubrique, sélectionnezCréer. Nommez le sujet*`Notifications IOTDDD`*et sélectionnezCréer. UnderRôle, sélectionnez le rôle que vous venez de créer, appelé*`AWSIOTDeviceDefenderAudit`*.



Tâche de sélectionMise à jour.



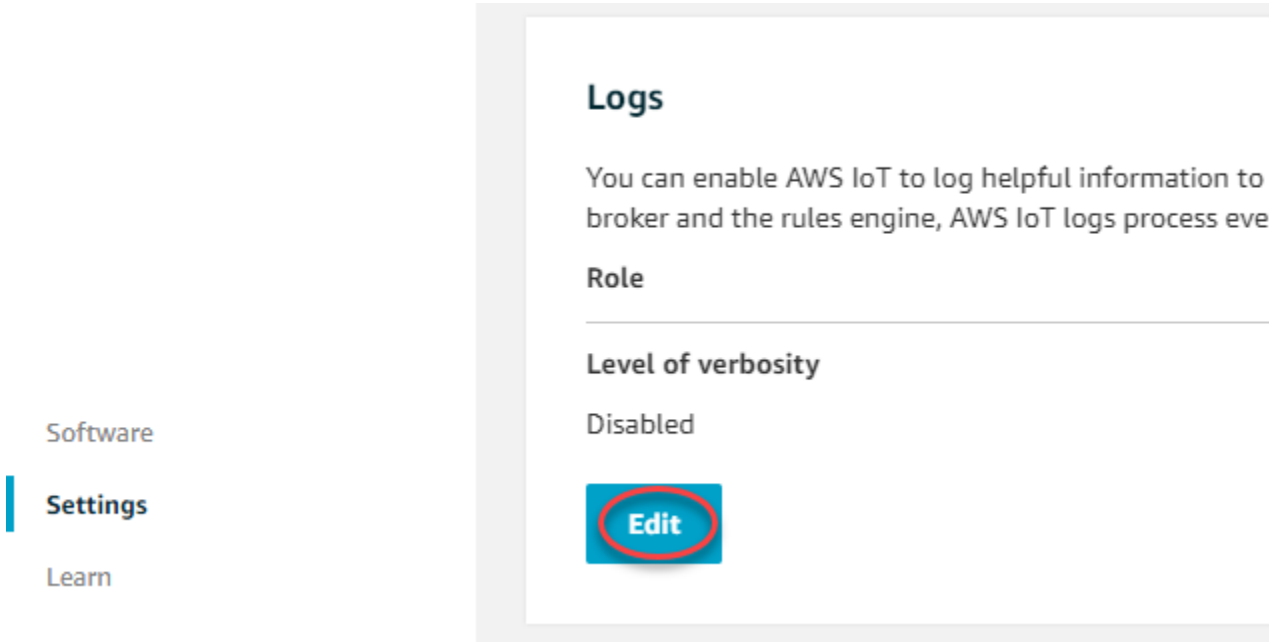
Si vous souhaitez recevoir des e-mails ou des SMS sur vos plateformes Ops via SNS, consultezUtilisation d'Amazon SNS pour les notifications utilisateur.

Activer la journalisation (facultatif)

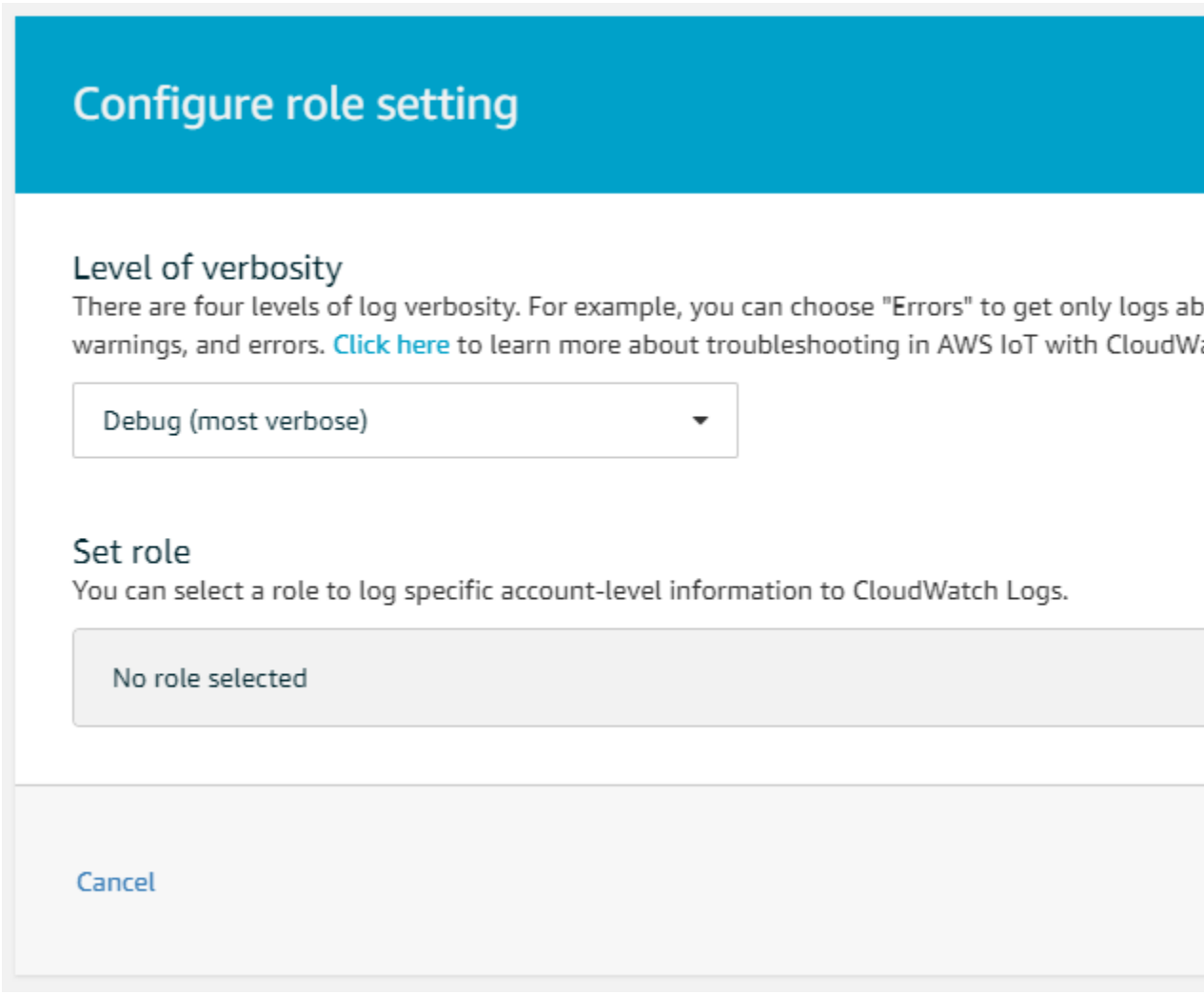
Cette procédure explique comment activer l'activationAWS IoTpour consigner les informations dans CloudWatch Logs. Cela vous permettra d'afficher les résultats de vos audits. L'activation de la journalisation peut entraîner des frais.

Pour activer la journalisation

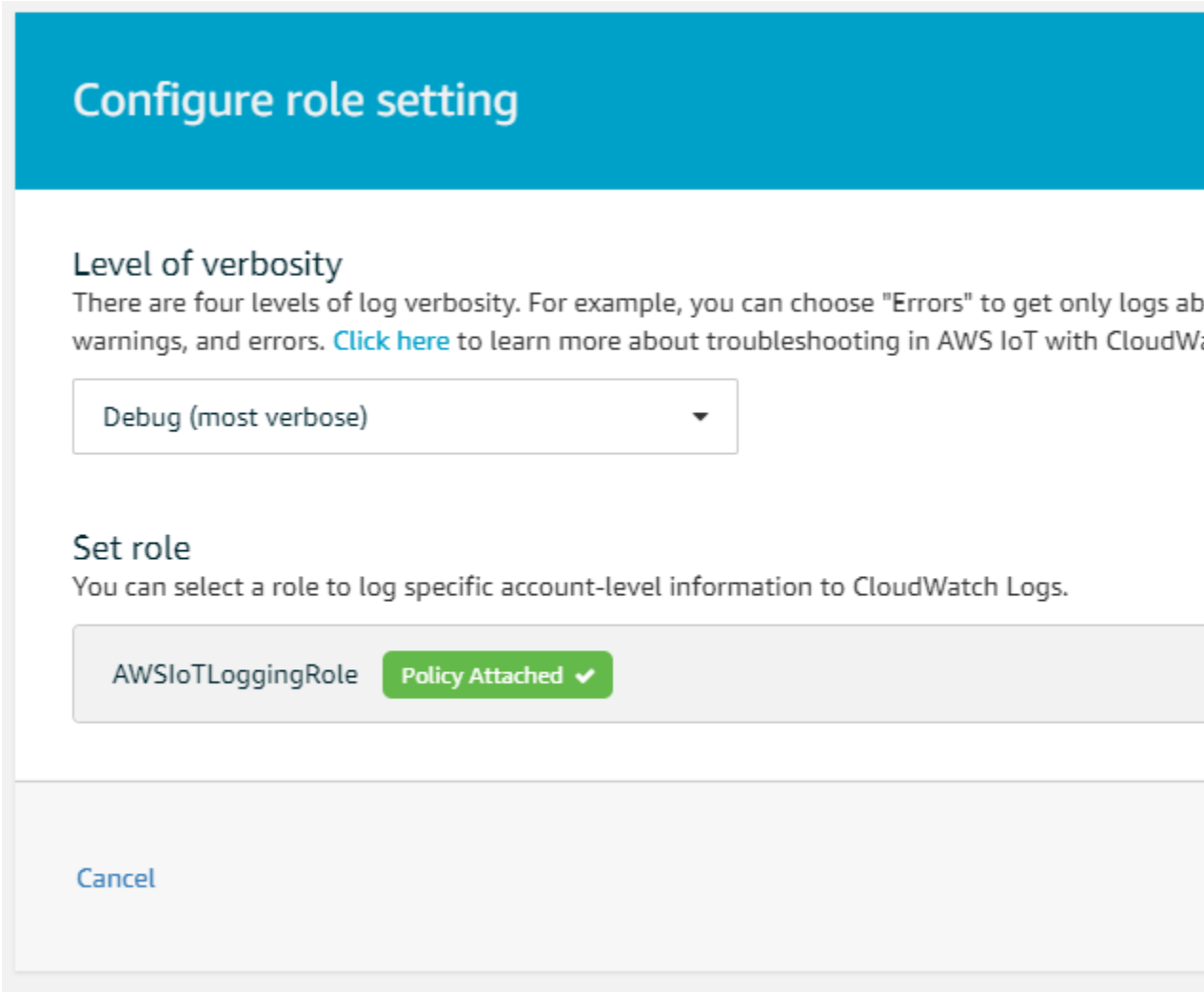
1. DansAWS, dans le volet de navigation, sélectionnezParamètres.
2. UnderJournaux, sélectionnezModifier.



3. UnderNiveau de détail, sélectionnezDébogage (plus verbeux).
4. UnderDéfinir le rôle, sélectionnezCréation d'un rôleet`AWSIOTLoggingRole`. Une stratégie sera automatiquement jointe.



Tâche de sélectionMise à jour.



Création d'uneAWS IoT Device DefenderRôle d'audit IAM (facultatif)

Dans la procédure suivante, vous créez un objetAWS IoT Device DefenderAudit le rôle IAM qui fournitAWS IoT Device DefenderAccès en lecture àAWS IoT.

1. Accédez à la console IAM à l'adressehttps://console.aws.amazon.com/iam/
2. Dans le volet de navigation, choisissezUsers, puis.Ajouter un utilisateur.
3. Dans User name (Nom d'utilisateur), entrez **`Administrator`**.
4. Activez la case à cocher près deAWSAccès à Management Console. Puis, sélectionnez Custom password (Mot de passe personnalisé, et entrez votre nouveau mot de passe dans la zone de texte.
5. (Facultatif) Par défaut, AWS oblige le nouvel utilisateur à créer un nouveau mot de passe lors de sa première connexion. Décochez la case en regard de User must create a new password at next sign-in (L'utilisateur doit créer un nouveau mot de passe à sa prochaine connexion) pour autoriser le nouvel utilisateur à réinitialiser son mot de passe une fois qu'il s'est connecté.
6. Choisissez Next (Suivant) Permissions (Autorisations).

7. Pour Set permissions (Définir les autorisations), sélectionnez Attach existing policies directly (Attacher directement les stratégies existantes).
8. Dans la liste des stratégies, activez la case à cocher pourAWSIOTDeviceDefenderAudit.
9. Choisissez Next (Suivant) Tags (Balises).
10. Choisissez Next (Suivant) VérificationPour afficher la liste des membres du groupe à ajouter au nouvel utilisateur. Une fois que vous êtes prêt à continuer, choisissez Create user.

Guide de détection de

Dans ce guide de mise en route, vous créez un profil de sécurité de détection ML qui utilise l'apprentissage automatique (ML) pour créer des modèles de comportement attendu basés sur les données de mesure historiques de vos périphériques. Pendant que ML Detect crée le modèle ML, vous pouvez surveiller sa progression. Une fois le modèle ML créé, vous pouvez afficher et examiner les alarmes de façon continue et atténuer les problèmes identifiés.

Pour plus d'informations sur ML Detect et ses commandes API et CLI, consultezDétection de détection (p. 932).

Ce chapitre contient les sections suivantes :

Prerequisites

- Un Compte AWS . Si vous ne disposez pas de ces éléments, consultezConfiguration de.

Comment utiliser ML Detect dans la console

Didacticiels

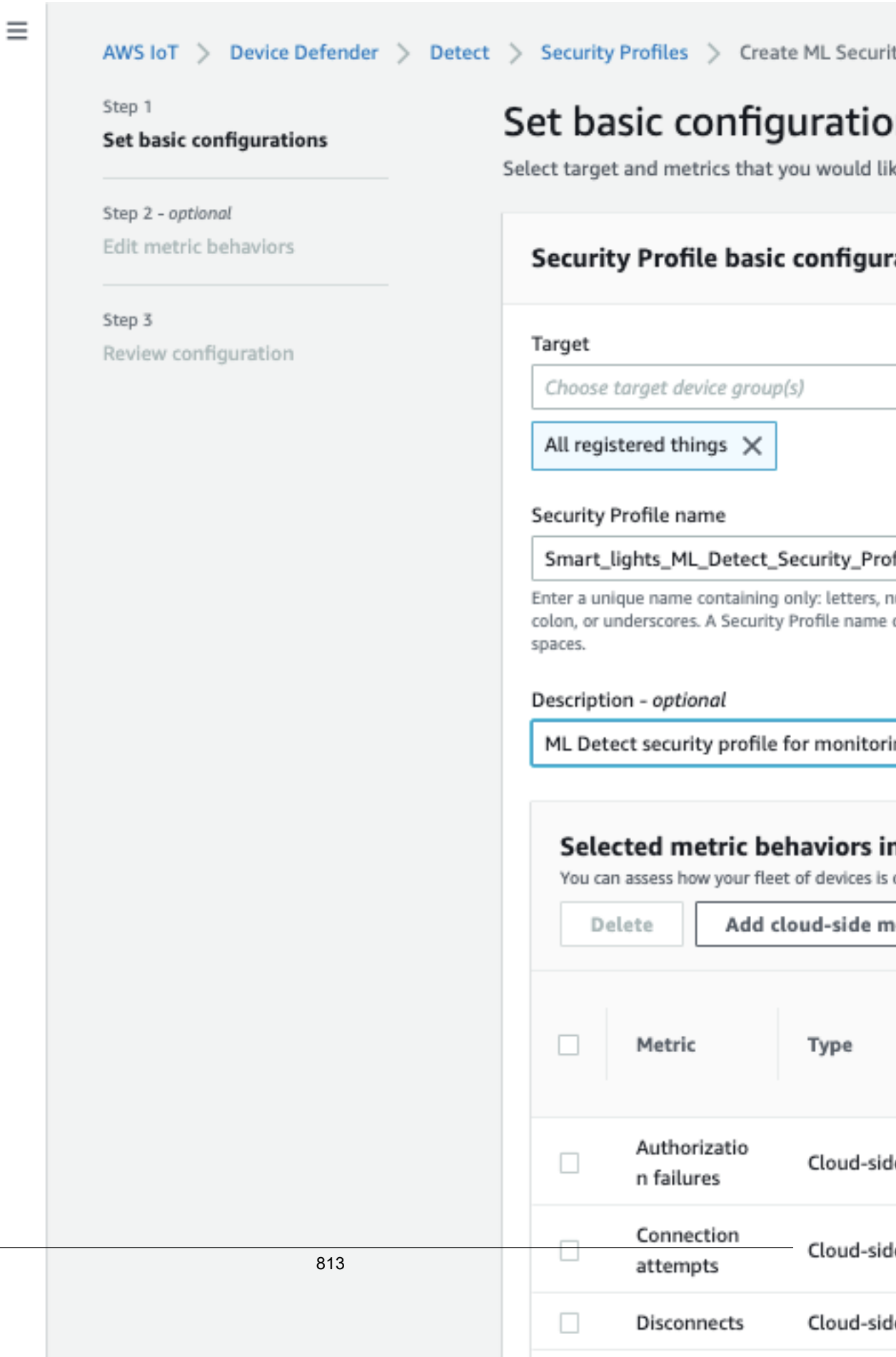
Activer la détection de

Les procédures suivantes expliquent comment configurer ML Detect dans la console.

1. Tout d'abord, assurez-vous que vos appareils vont créer les points de données minimum requis comme défini dansML Detect configuration minimale (p. 933)pour la formation continue et l'actualisation du modèle. Pour que la collecte des données progresse, assurez-vous que votre profil de sécurité est attaché à une cible, qui peut être un groupe d'objets ou de choses.
2. DansAWS IoTconsoleDans le volet de navigation, développezDéfendre. ChoisissezDétecter,Profils de sécurité,Créez un profil de sécurité, puisCréer un profil d'anomalie ML Détecter.
3. Dans la pageDéfinir les configurations de base, procédez comme suit.

    - UnderTarget, choisissez vos groupes d'équipements cibles.
    - UnderNom du profil de sécurité, saisissez un nom pour votre profil de sécurité.

- (Facultatif) SousDescriptionVous pouvez écrire dans une brève description pour le profil ML.
- UnderComportements de mesure sélectionnés dans le profil de sécurité, choisissez les mesures que vous souhaitez surveiller.

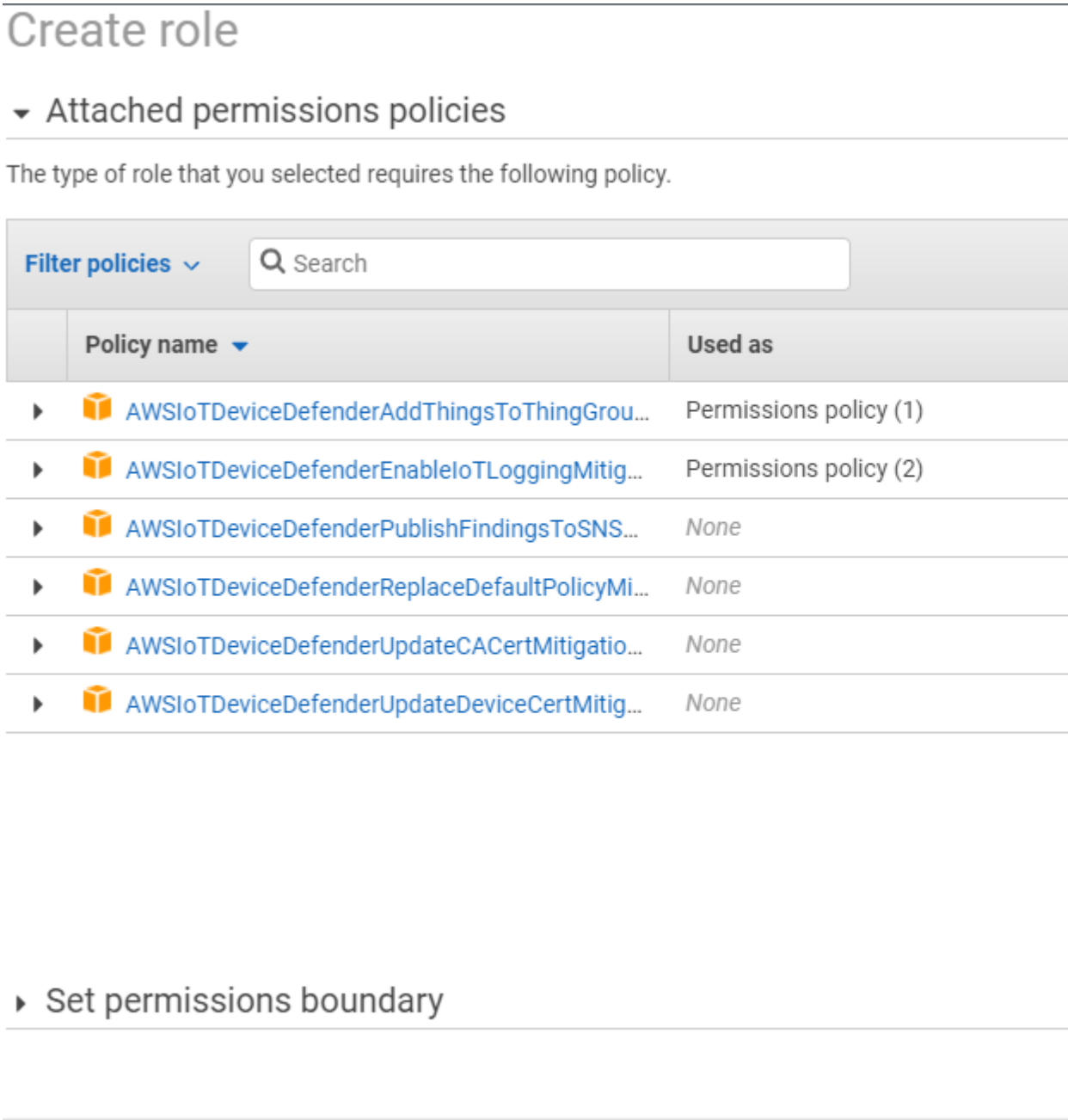

AWS IoT > Device Defender > Detect > Security Profiles > Create ML Securit
Step 1
Set basic configurations
Step 2 - optional
Edit metric behaviors
Step 3
Review configuration
Set basic configuratio
Select target and metrics that you would lik
Security Profile basic configur
Target
Choose target device group(s)
All registered things
Security Profile name
Smart_lights_ML_Detect_Security_Prof
Enter a unique name containing only: letters, n
colon, or underscores. A Security Profile name
spaces.
Description - optional
ML Detect security profile for monitori
Selected metric behaviors in
You can assess how your fleet of devices is
Delete
Add cloud-side m
Metric
Type
Authorizatio n failures
Cloud-sid
Connection attempts
Cloud-sid
Disconnects
Cloud-sid

Lorsque vous avez terminé, sélectionnez Next.

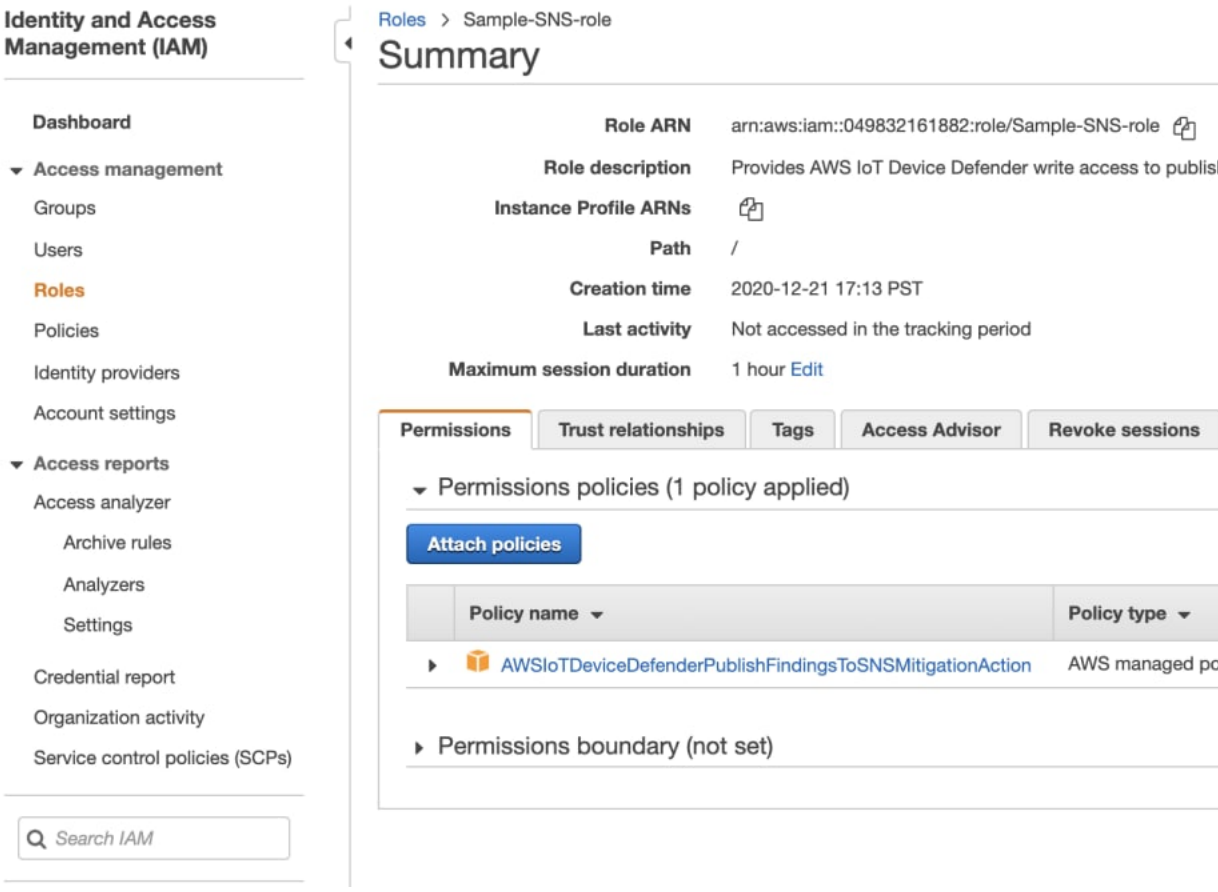
4. Dans la pageDéfinir SNS (facultatif), spécifiez une rubrique SNS pour les notifications d'alarme lorsqu'un appareil enfreint un comportement dans votre profil. Choisissez un rôle IAM que vous utiliserez pour publier dans la rubrique SNS sélectionnée.

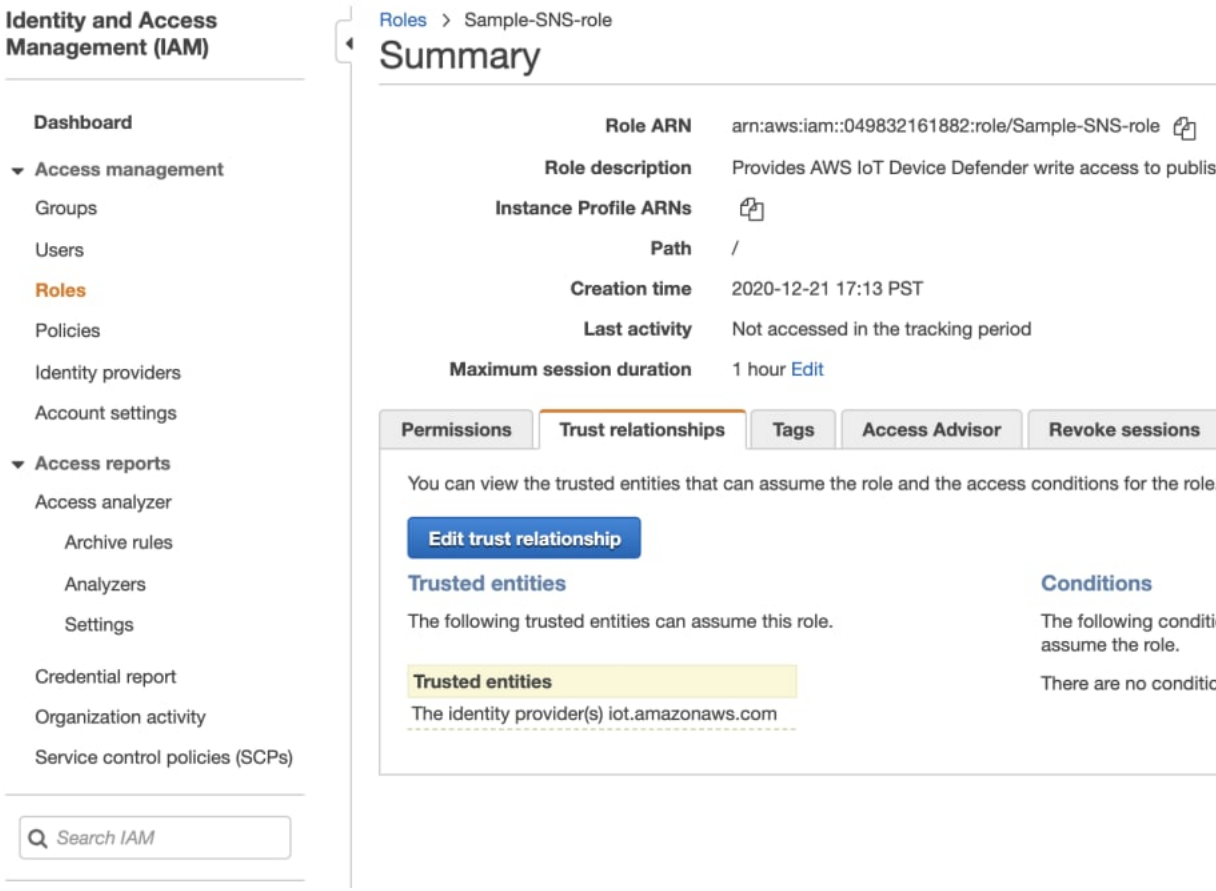
   Si vous n'avez pas encore de rôle SNS, procédez comme suit pour créer un rôle avec les autorisations appropriées et les relations d'approbation requises.

   - Accédez à .Console IAM. Dans le panneau de navigation, choisissez Rôles, puis Créer un rôle.
   - UnderSélectionnez le type d'entité de confiance, sélectionnezAWSService. Ensuite, sousChoisissez un cas d'utilisation, choisissezIoTet sousSélectionnez votre cas d'utilisation, choisissezIoT - Mesures d'atténuation de Device Defender. Lorsque vous avez terminé, sélectionnezSuivant: Permissions (Autorisations).
   - UnderStratégies d'autorisations attachées, s'assurer queAWSIOTDeviceDefenderPublishFindingStoSNMitigationActionest sélectionné, puis choisissezSuivant: Tags (Balises).

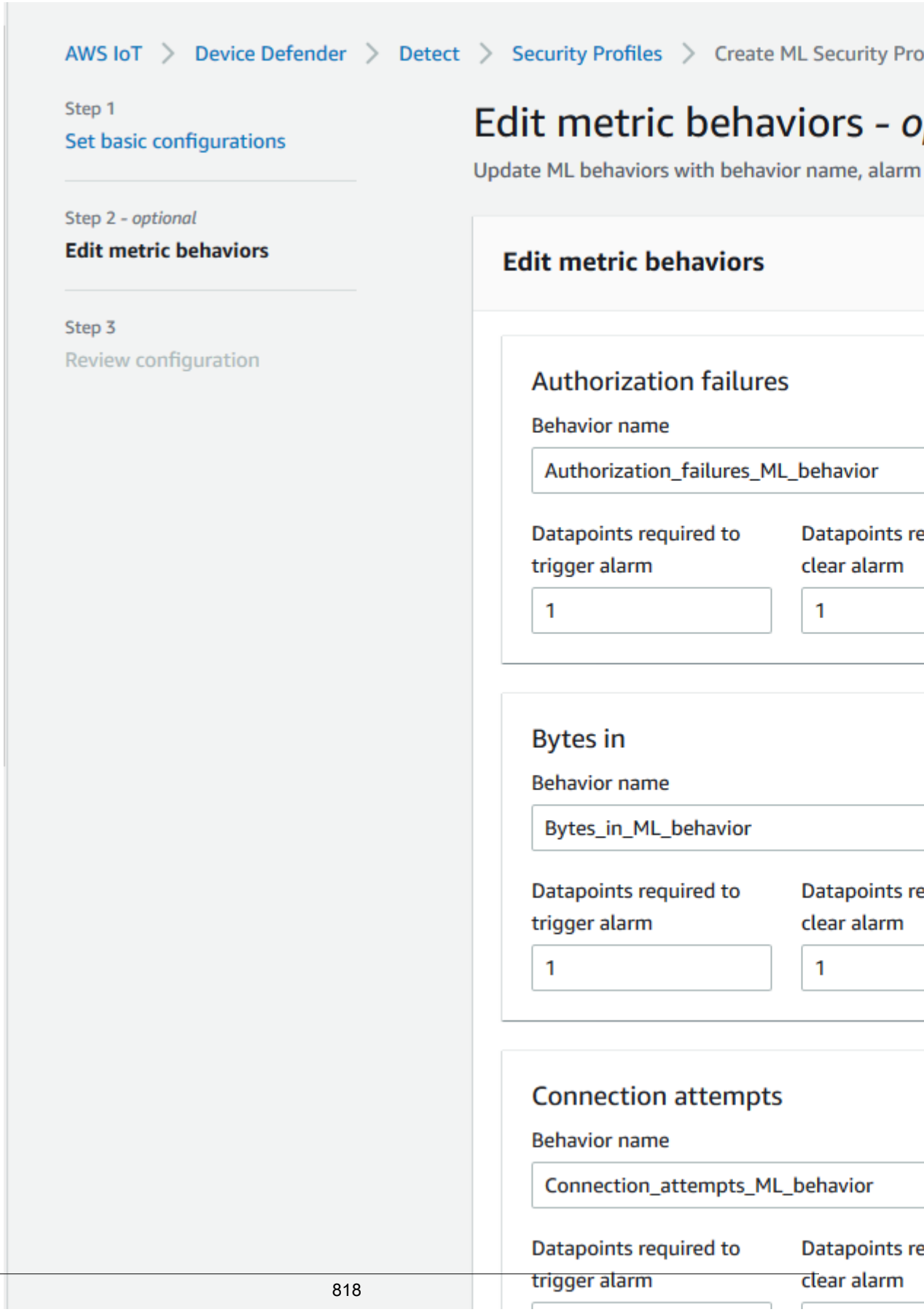


- UnderAjouter des balises (facultatif), vous pouvez ajouter toutes les balises que vous souhaitez associer à votre rôle. Lorsque vous avez terminé, sélectionnezSuivant: Review (Examiner).
- UnderVérification, donnez un nom à votre rôle et assurez-vous queAWSIOTDeviceDefenderPublishFindingStoSNMitigationActionest répertorié sousAutorisationsandAWSservice : iot.amazonaws.comest répertorié sousRelations d'approbation. Lorsque vous avez terminé, sélectionnezCréation d'un rôle.

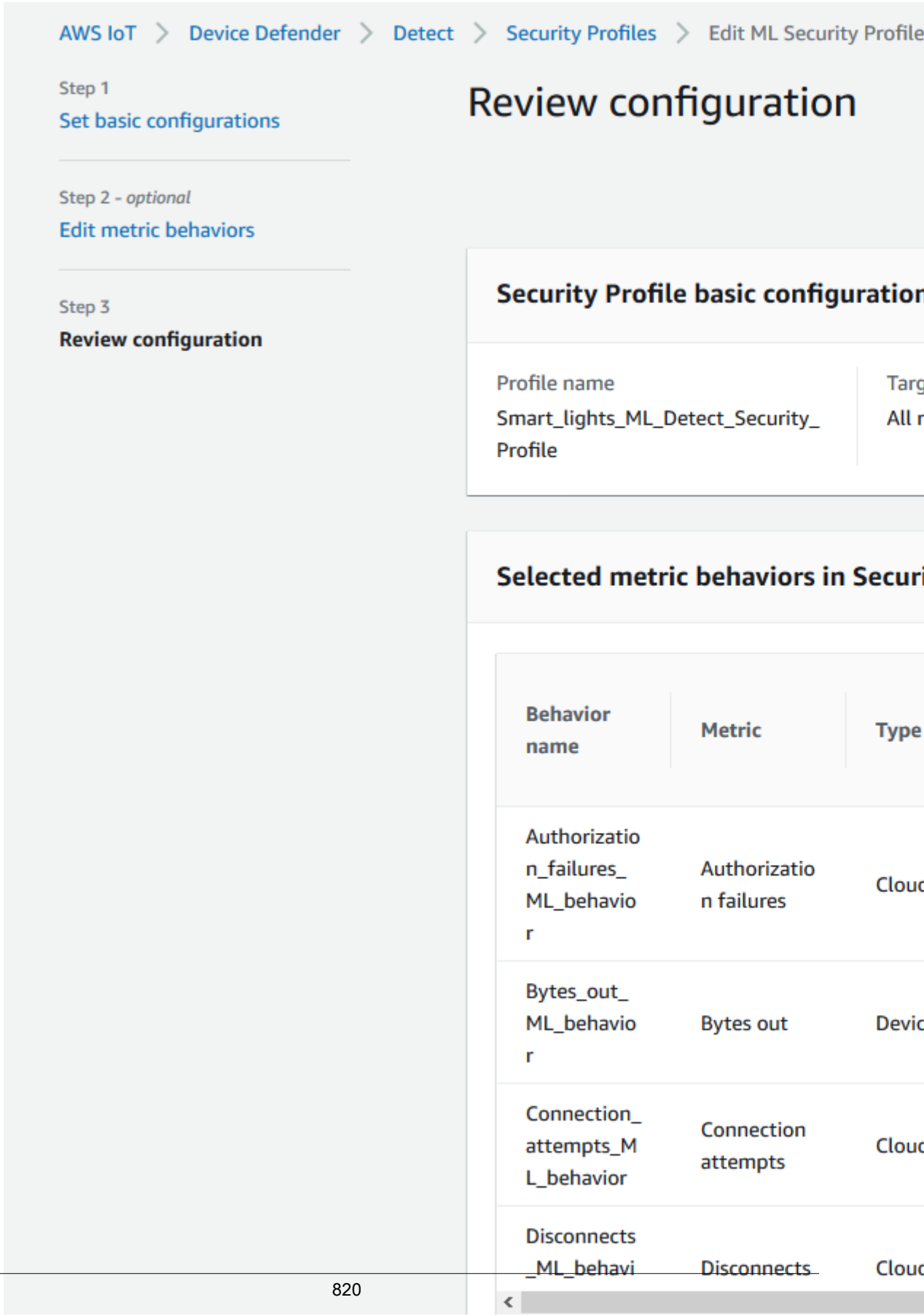

Identity and Access Management (IAM)
Dashboard
Access management
Groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
Search IAM
Roles > Sample-SNS-role
Summary
Role ARN
arn:aws:iam::049832161882:role/Sample-SNS-role
Role description
Provides AWS IoT Device Defender write access to publis
Instance Profile ARNs
Path
/
Creation time
2020-12-21 17:13 PST
Last activity
Not accessed in the tracking period
Maximum session duration
1 hour Edit
Permissions
Trust relationships
Tags
Access Advisor
Revoke sessions
Permissions policies (1 policy applied)
Attach policies
Policy name
Policy type
AWSIoTDeviceDefenderPublishFindingsToSNSMitigationAction
AWS managed po
Permissions boundary (not set)



5. Dans la pageModifier le comportement de la mesure, vous pouvez personnaliser vos paramètres de comportement ML.


AWS IoT > Device Defender > Detect > Security Profiles > Create ML Security Pro
Step 1
Set basic configurations
Step 2 - optional
Edit metric behaviors
Step 3
Review configuration
Edit metric behaviors - o
Update ML behaviors with behavior name, alarm
Edit metric behaviors
Authorization failures
Behavior name
Authorization_failures_ML_behavior
Datapoints required to trigger alarm
1
Datapoints re clear alarm
1
Bytes in
Behavior name
Bytes_in_ML_behavior
Datapoints required to trigger alarm
1
Datapoints re clear alarm
1
Connection attempts
Behavior name
Connection_attempts_ML_behavior
Datapoints required to trigger alarm
Datapoints re clear alarm

6. Lorsque vous avez terminé, sélectionnez Next.
7. Dans la pageVérifiez la configuration, vérifiez les comportements que vous souhaitez que l'apprentissage automatique surveille, puis choisissezSuivant.


AWS IoT > Device Defender > Detect > Security Profiles > Edit ML Security Profile
Step 1
Set basic configurations
Step 2 - optional
Edit metric behaviors
Step 3
Review configuration
Review configuration
Security Profile basic configuration
Profile name
Smart_lights_ML_Detect_Security_Profile
Selected metric behaviors in Securi
Behavior name
Metric
Type
Authorization_failures_ML_behavior
Authorization failures
Bytes_out_ML_behavior
Bytes out
Connection_attempts_ML_behavior
Connection attempts
Disconnects_ML_behavi
Disconnects

8. Après avoir créé votre profil de sécurité, vous êtes redirigé vers laProfils de sécurité, où apparaît le profil de sécurité nouvellement créé.

   Note

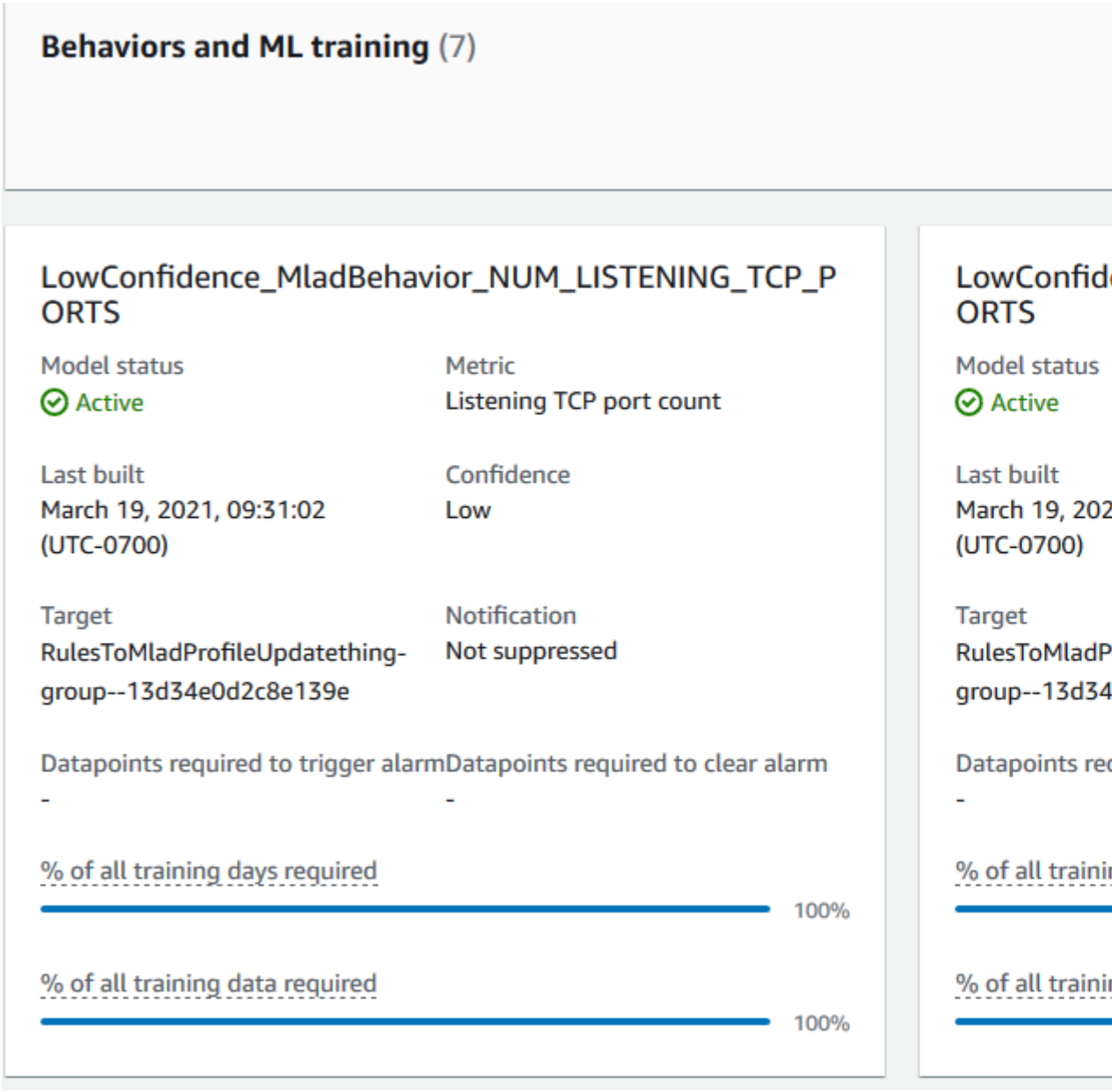
   La formation initiale et la création du modèle ML prennent 14 jours. Vous pouvez vous attendre à voir les alarmes une fois qu'elles sont terminées, s'il y a une activité anormale sur vos appareils.

Surveiller l'état de votre modèle ML

Pendant que vos modèles ML sont dans la période de formation initiale, vous pouvez suivre leur progression à tout moment en suivant les étapes suivantes.

1. DansAWS IoTconsoleDans le volet de navigation, développezDéfendre, puis.Détecter,Profils de sécurité.
2. Dans la pageProfils de sécurité, choisissez le profil de sécurité que vous souhaitez consulter. Ensuite, choisissezComportements et formation ML.
3. Dans la pageComportements et formation ML, vérifiez la progression de la formation de vos modèles ML.

   Une fois que l'état de votre modèle estActif, il commencera à prendre des décisions Détecter en fonction de votre utilisation et mettra à jour le profil tous les jours.



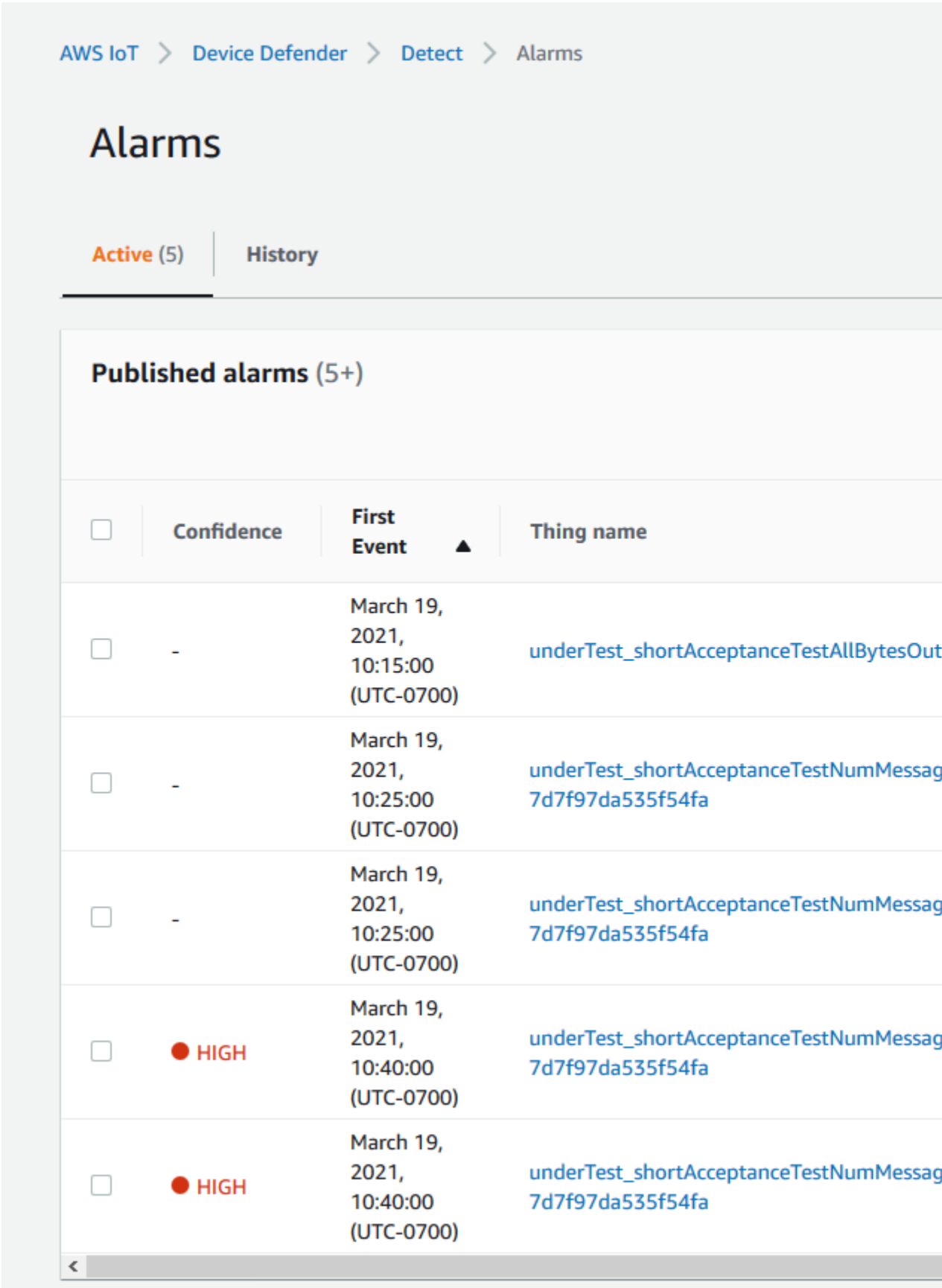
Note

Si votre modèle ne progresse pas comme prévu, assurez-vous que vos appareils respectent laConfiguration requise (p. 933).

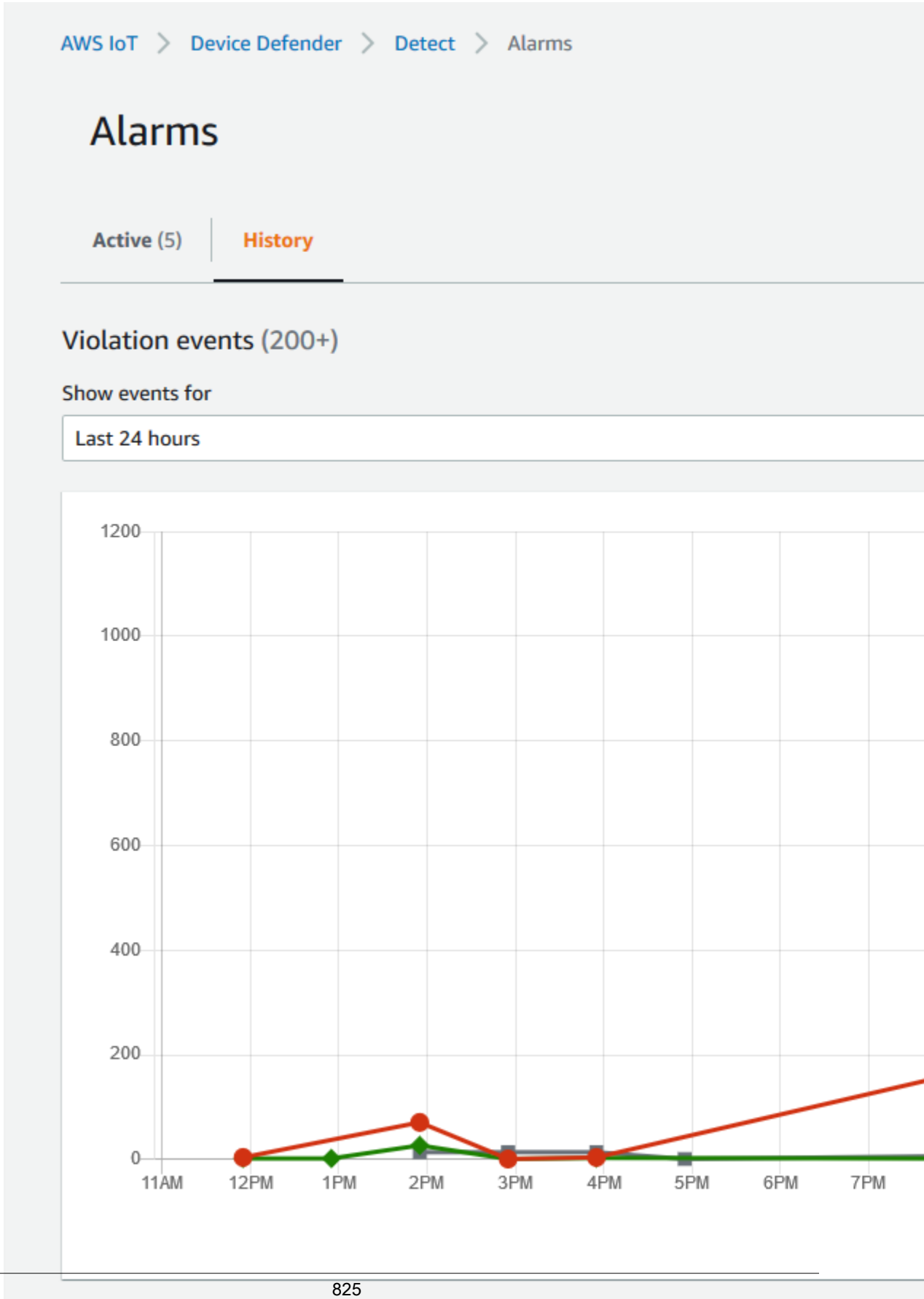
Passez en revue vos alarmes ML Detect

Une fois vos modèles ML construits et prêts pour l'inférence de données, vous pouvez consulter et examiner régulièrement les alarmes identifiées par les modèles.

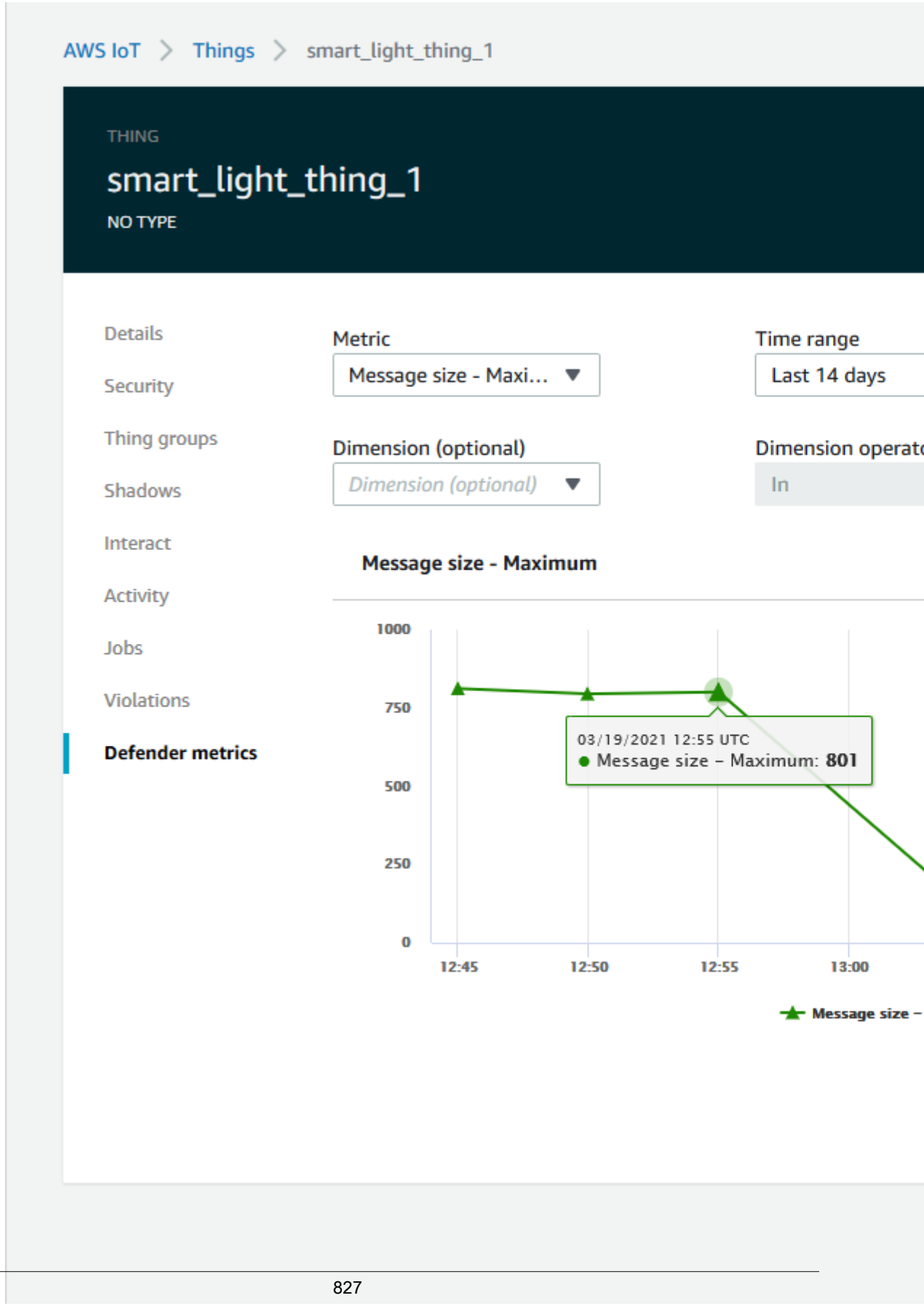
1. DansAWS IoTconsoleDans le volet de navigation, développezDéfendre, puis.Détecter,Alarmes.


AWS IoT > Device Defender > Detect > Alarms
Alarms
Active (5)
History
Published alarms (5+)
Confidence
First Event
Thing name
March 19, 2021, 10:15:00 (UTC-0700)
underTest_shortAcceptanceTestAllBytesOut
March 19, 2021, 10:25:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
March 19, 2021, 10:25:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
HIGH
March 19, 2021, 10:40:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
HIGH
March 19, 2021, 10:40:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa

2. Si vous accédez àHistorique, vous pouvez également afficher des détails sur vos appareils qui ne sont plus dans les alarmes.


AWS IoT
Device Defender
Detect
Alarms
Alarms
Active (5)
History
Violation events (200+)
Show events for
Last 24 hours
1200
1000
800
600
400
200
0
11AM
12PM
1PM
2PM
3PM
4PM
5PM
6PM
7PM

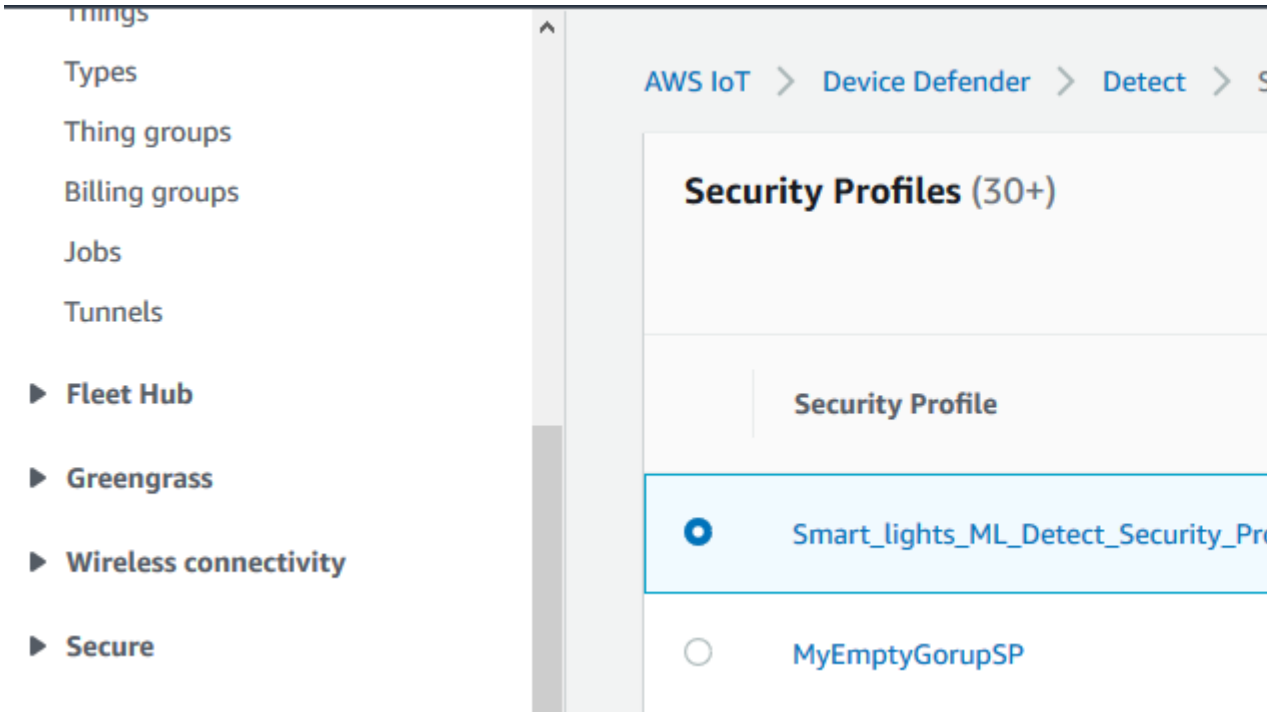
Pour obtenir plus d'informations, sousGérerchoisirObjets, choisissez la chose pour laquelle vous souhaitez voir plus de détails, puis accédez àmétriques Defender. Vous pouvez accéder à laGraphique de métriques Defenderet effectuez votre enquête sur tout ce qui est en alarme depuis leActifOnglet. Dans ce cas, le graphique montre un pic dans la taille du message, qui a déclenché l'alarme. Vous pouvez voir l'alarme effacée par la suite.


AWS IoT
Things
smart_light_thing_1
THING
smart_light_thing_1
NO TYPE
Details
Security
Thing groups
Shadows
Interact
Activity
Jobs
Violations
Defender metrics
Metric
Message size - Maxi...
Time range
Last 14 days
Dimension (optional)
Dimension (optional)
Dimension operato
In
Message size - Maximum
1000
750
500
250
0
12:45
12:50
12:55
13:00
03/19/2021 12:55 UTC
Message size – Maximum: 801
Message size –

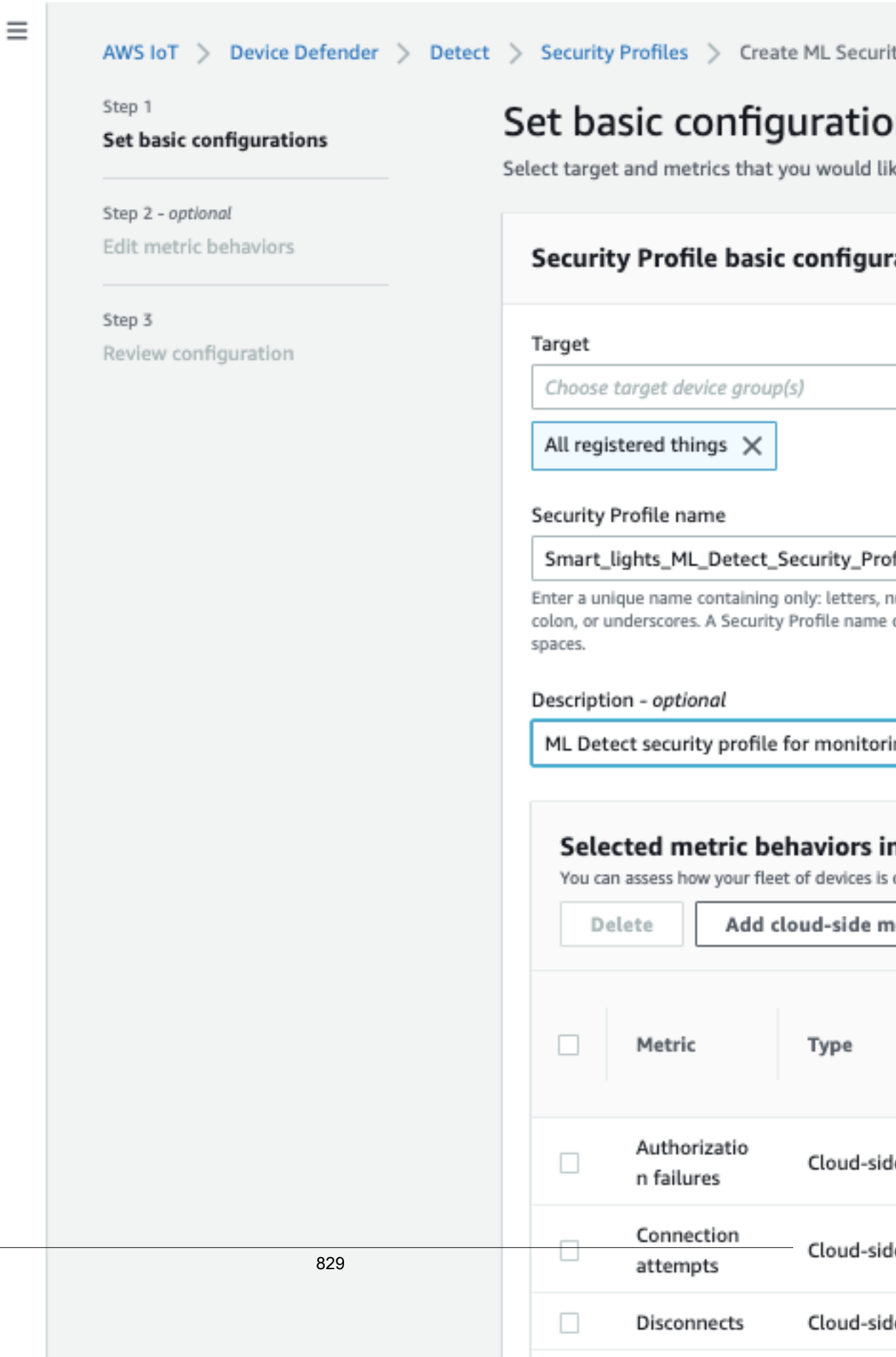
Ajustez vos alarmes ML

Une fois vos modèles ML construits et prêts pour l'évaluation des données, vous pouvez mettre à jour les paramètres de comportement ML de votre profil de sécurité pour modifier la configuration. La procédure suivante vous montre comment mettre à jour les paramètres de comportement ML de votre profil de sécurité dans laAWS CLI.

1. DansAWS IoTconsoleDans le volet de navigation, développezDéfendre, puis.Détecter,Profils de sécurité.
2. Dans la pageProfils de sécurité, cochez la case en regard du profil de sécurité que vous souhaitez consulter. Ensuite, choisissezActions,Modifier.

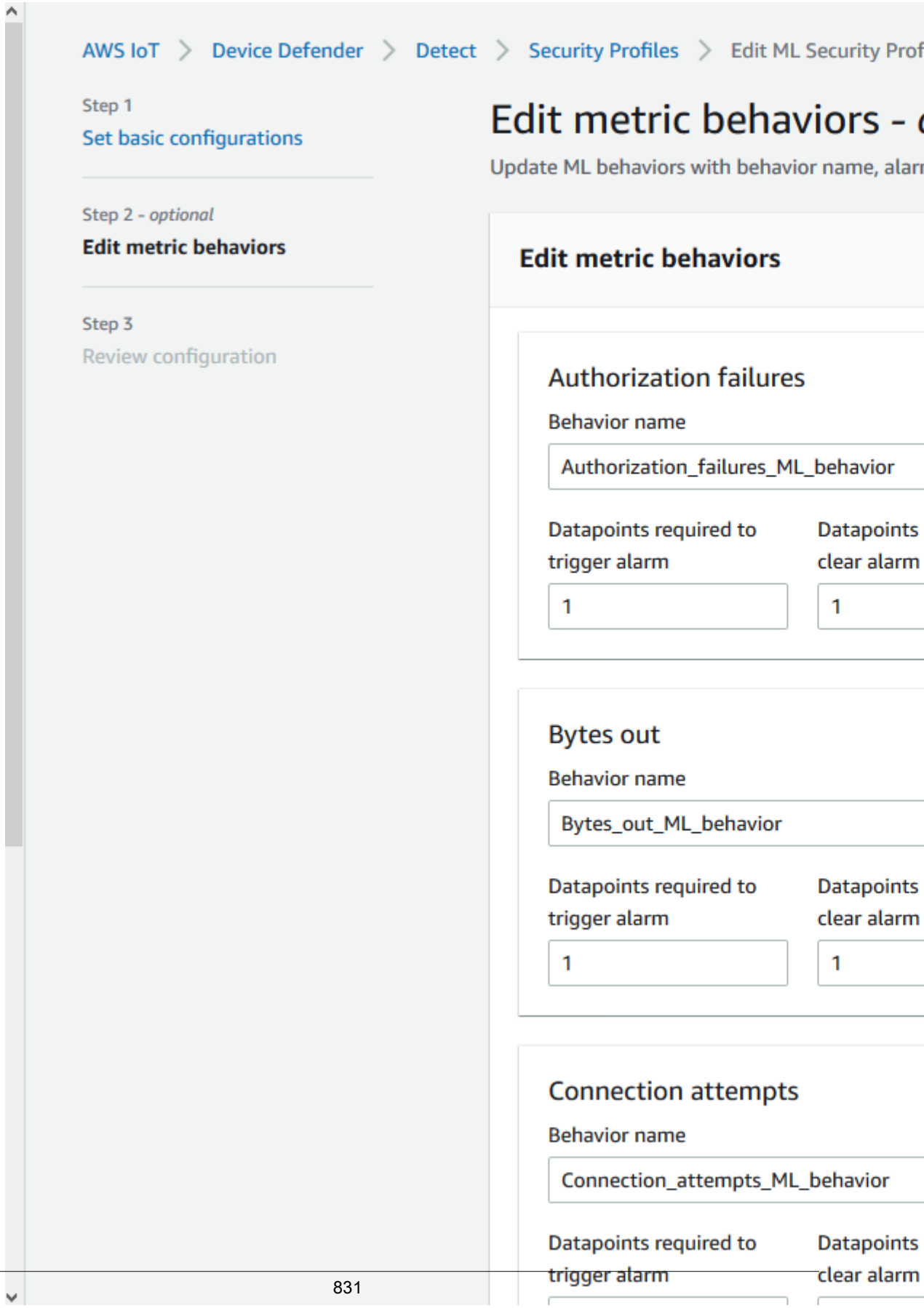


3. UnderDéfinir les configurations de base, vous pouvez ajuster les groupes cibles de profil de sécurité ou modifier les mesures que vous souhaitez surveiller.


AWS IoT > Device Defender > Detect > Security Profiles > Create ML Securit
Step 1
Set basic configurations
Step 2 - optional
Edit metric behaviors
Step 3
Review configuration
Set basic configuratio
Select target and metrics that you would lik
Security Profile basic configura
Target
Choose target device group(s)
All registered things
Security Profile name
Smart_lights_ML_Detect_Security_Prof
Enter a unique name containing only: letters, n
colon, or underscores. A Security Profile name
spaces.
Description - optional
ML Detect security profile for monitori
Selected metric behaviors in
You can assess how your fleet of devices is
Delete
Add cloud-side m
Metric
Type
Authorizatio n failures
Cloud-sid
Connection attempts
Cloud-sid
Disconnects
Cloud-sid

4. Vous pouvez mettre à jour l'une des méthodes suivantes en accédant àModifier les comportements de mesure.

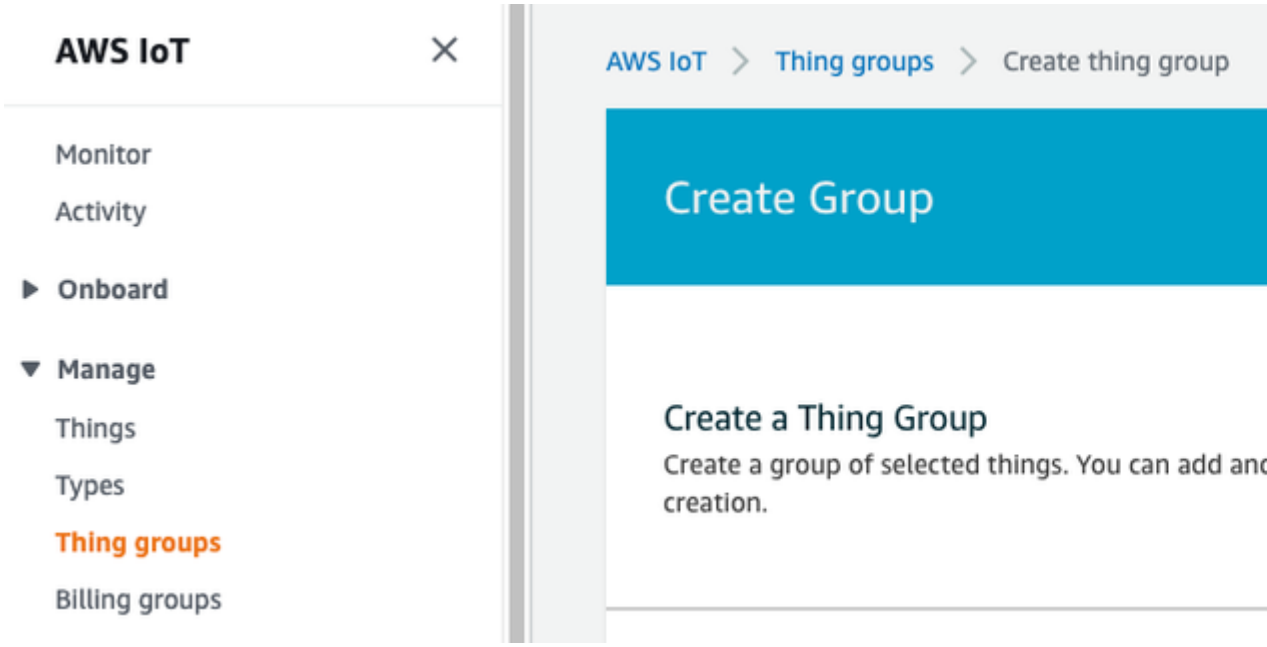
   - Vos points de données du modèle ML requis pour déclencher une alarme
   - Vos points de données de modèle ML requis pour effacer l'alarme
   - Votre niveau de confiance ML Detect
   - Vos notifications ML Detect (par exemple,Non supprimé,Suppression de)


AWS IoT
Device Defender
Detect
Security Profiles
Edit ML Security Prof
Step 1
Set basic configurations
Step 2 - optional
Edit metric behaviors
Step 3
Review configuration
Edit metric behaviors -
Update ML behaviors with behavior name, alarr
Edit metric behaviors
Authorization failures
Behavior name
Authorization_failures_ML_behavior
Datapoints required to trigger alarm
1
Datapoints clear alarm
1
Bytes out
Behavior name
Bytes_out_ML_behavior
Datapoints required to trigger alarm
1
Datapoints clear alarm
1
Connection attempts
Behavior name
Connection_attempts_ML_behavior
Datapoints required to trigger alarm
Datapoints clear alarm

Atténuer les problèmes de périphériques

1. (Facultatif)Avant de configurer des actions d'atténuation de la quarantaine, nous allons configurer un groupe de quarantaine dans lequel nous allons déplacer le périphérique en violation. Vous pouvez aussi utiliser un groupe existant.
2. Accédez àGérer,Groupes d'objets, puisCréer un groupe d'objets. Nommez votre groupe de choses. Pour ce didacticiel, nous allons nommer notre groupe d'objets`Quarantine_group`. UnderGroupe d'objets,Sécurité, appliquez la stratégie suivante au groupe d'objets.

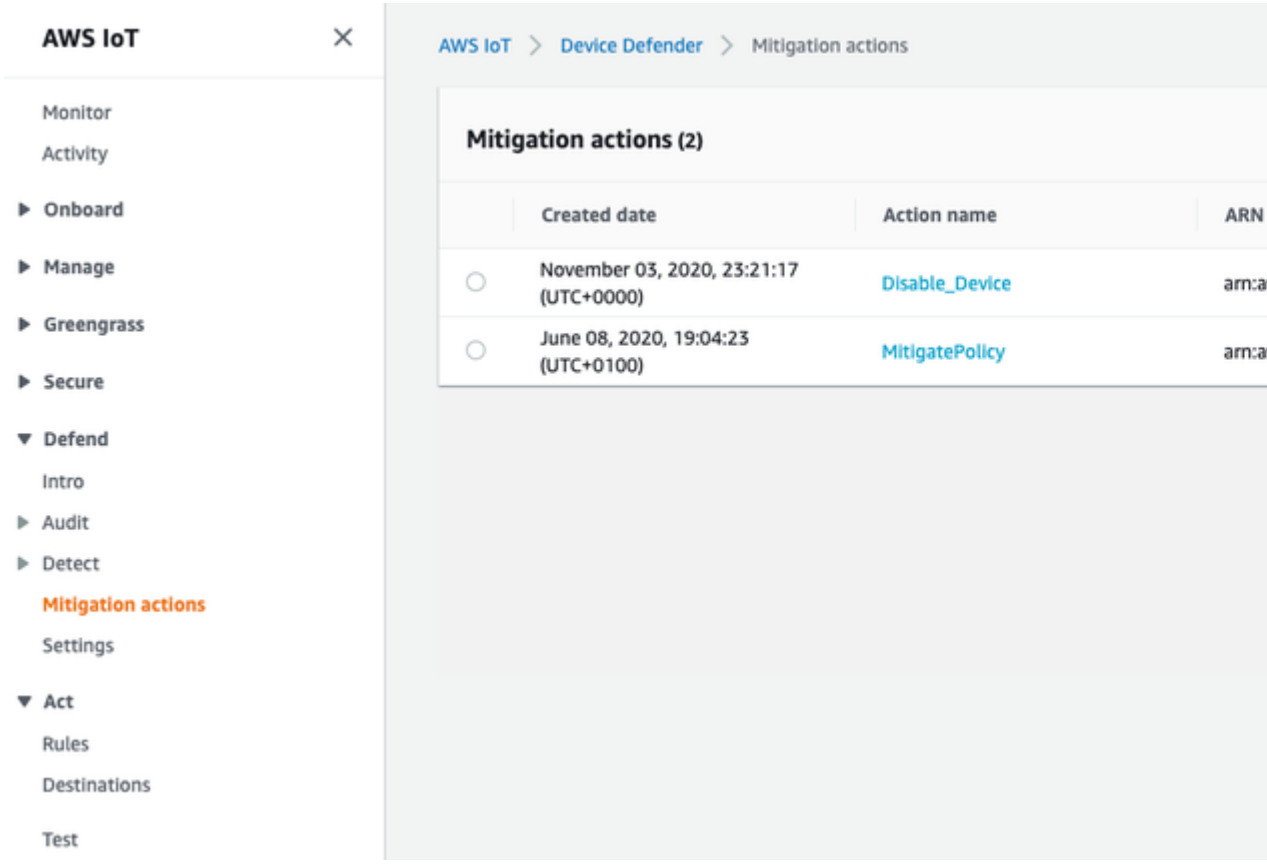
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "iot:*",
      "Resource": "*",
    }
  ]
}
```



Lorsque vous avez terminé, sélectionnezCréer un groupe d'objets.

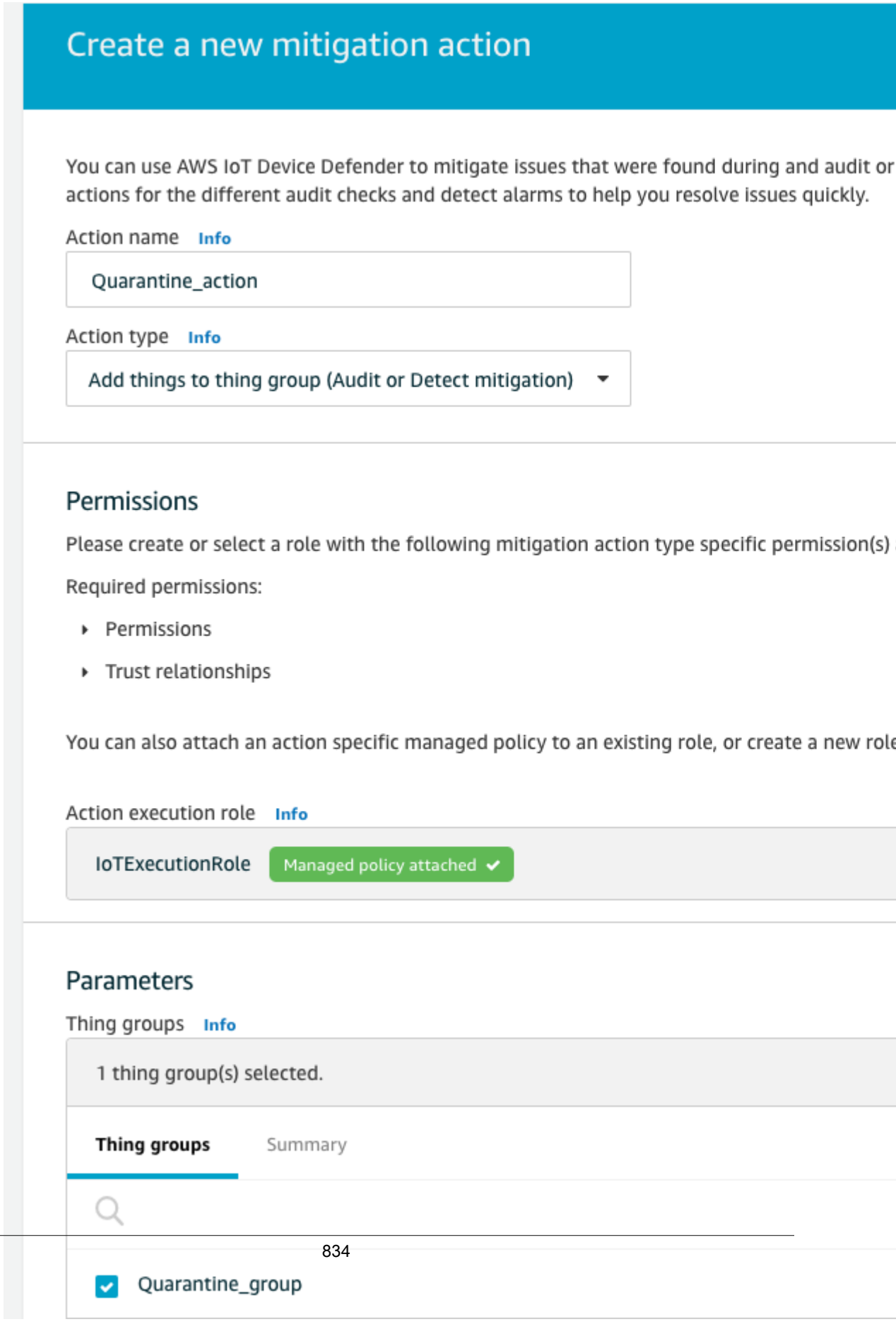
3. Maintenant que nous avons créé un groupe de choses, créons une action d'atténuation qui déplace les périphériques qui sont en alarme dans le`Quarantine_group`.

   UnderDéfendre,Actions d'atténuation, choisissezCréer.



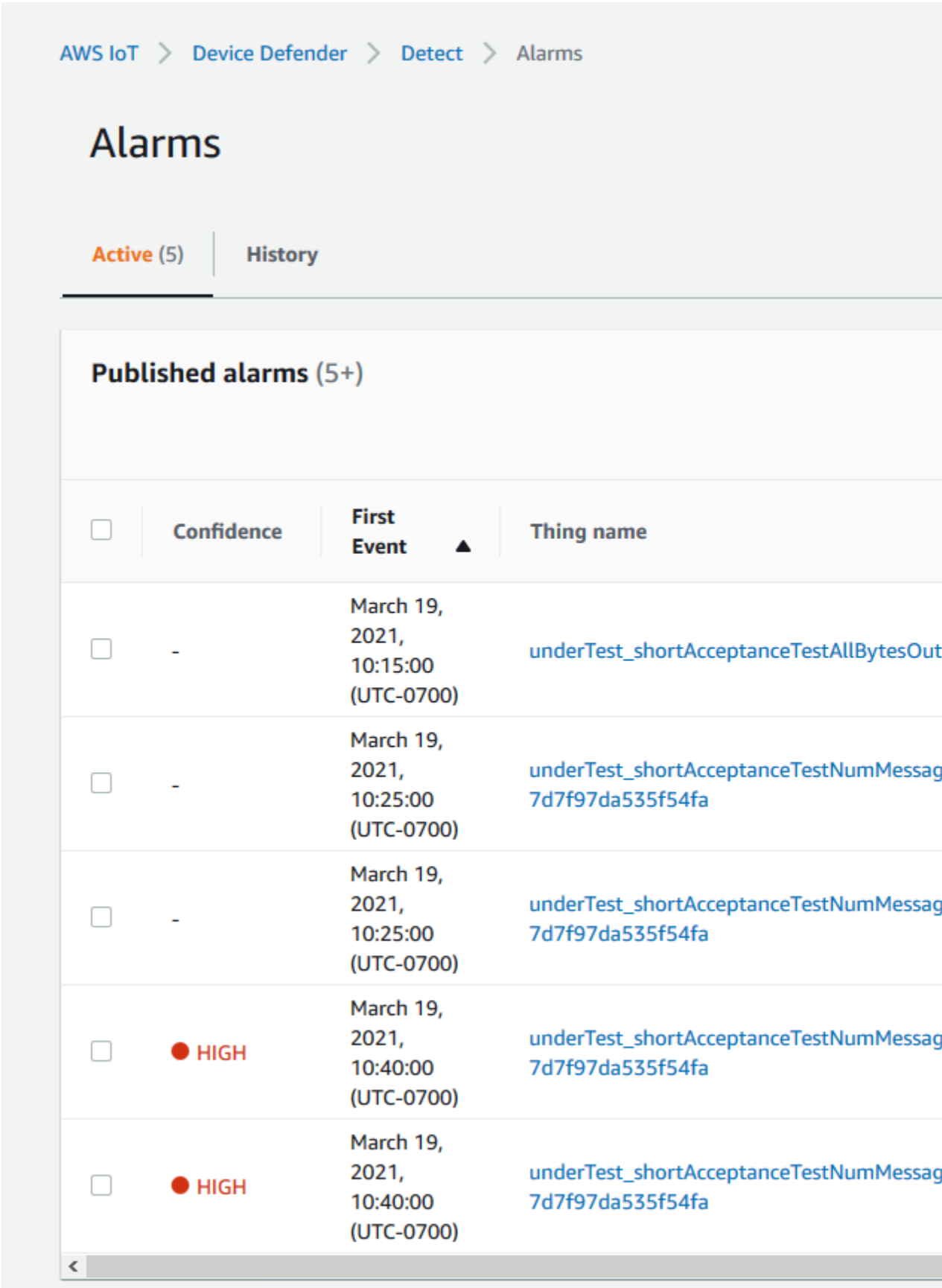
4. Dans la pageCréer une nouvelle action d'atténuation, saisissez les informations suivantes.

   - Action name (Nom de l'action) : Attribuez un nom à votre action d'atténuation, par exemple**`Quarantine_action`**.
   - Type d'action : Choisissez le type d'action. Nous choisironsAjouter des éléments à un groupe de choses (Audit ou Détecter l'atténuation).
   - Rôle d'exécution d'action : Créez un rôle ou choisissez un rôle existant si vous en avez créé un plus tôt.
   - Paramètres : Choisissez un groupe d'objets. Nous pouvons utiliser`Quarantine_group`, que nous avons créé plus tôt.


Create a new mitigation action
You can use AWS IoT Device Defender to mitigate issues that were found during and audit or
actions for the different audit checks and detect alarms to help you resolve issues quickly.
Action name Info
Quarantine_action
Action type Info
Add things to thing group (Audit or Detect mitigation)
Permissions
Please create or select a role with the following mitigation action type specific permission(s)
Required permissions:
Permissions
Trust relationships
You can also attach an action specific managed policy to an existing role, or create a new role
Action execution role Info
IoTExecutionRole
Managed policy attached
Parameters
Thing groups Info
1 thing group(s) selected.
Thing groups
Summary
Quarantine_group

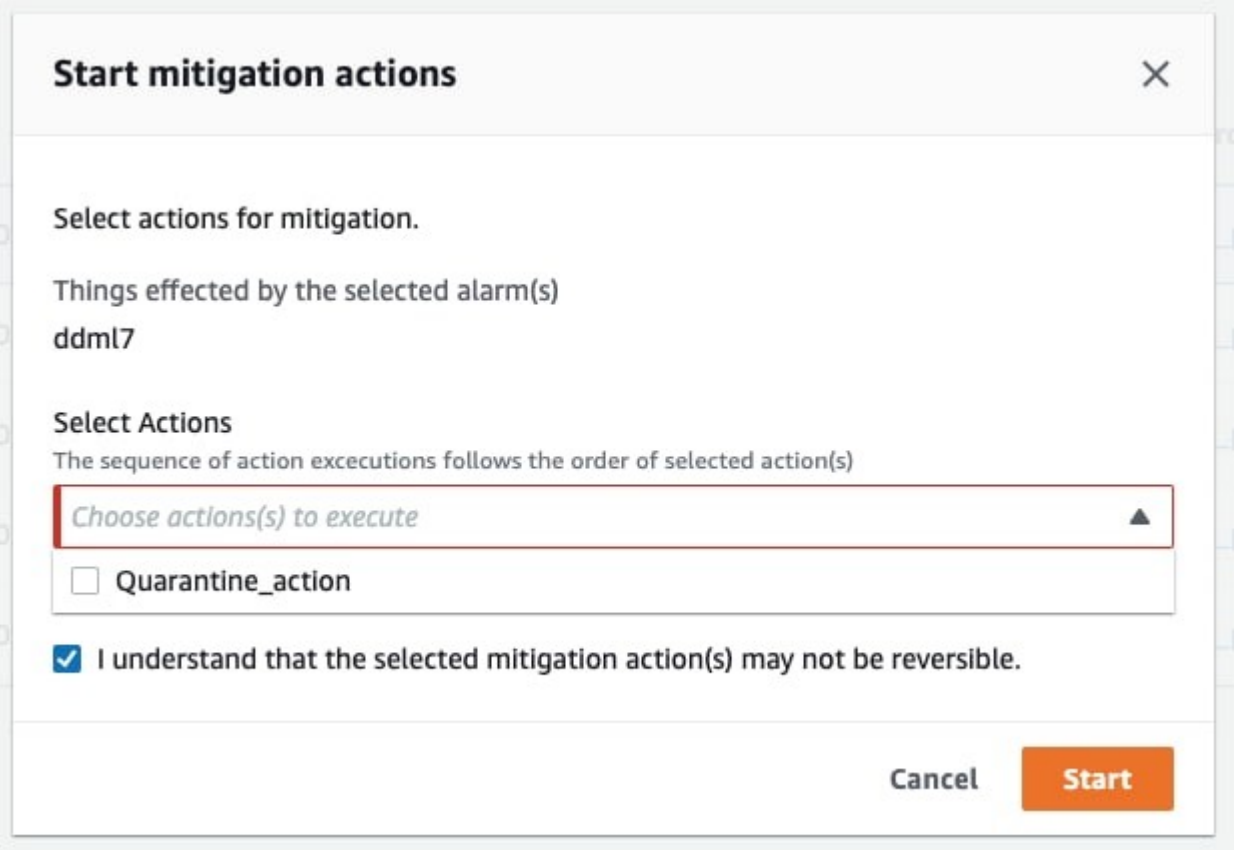
Lorsque vous avez terminé, sélectionnez Save. Vous disposez désormais d'une action d'atténuation qui déplace les périphériques en alarme vers un groupe de choses de quarantaine, et d'une action d'atténuation pour isoler le périphérique pendant que vous enquêtez.

5. Accédez àDefender,Détecter,Alarmes. Vous pouvez voir quels appareils sont en état d'alarme sousActif.

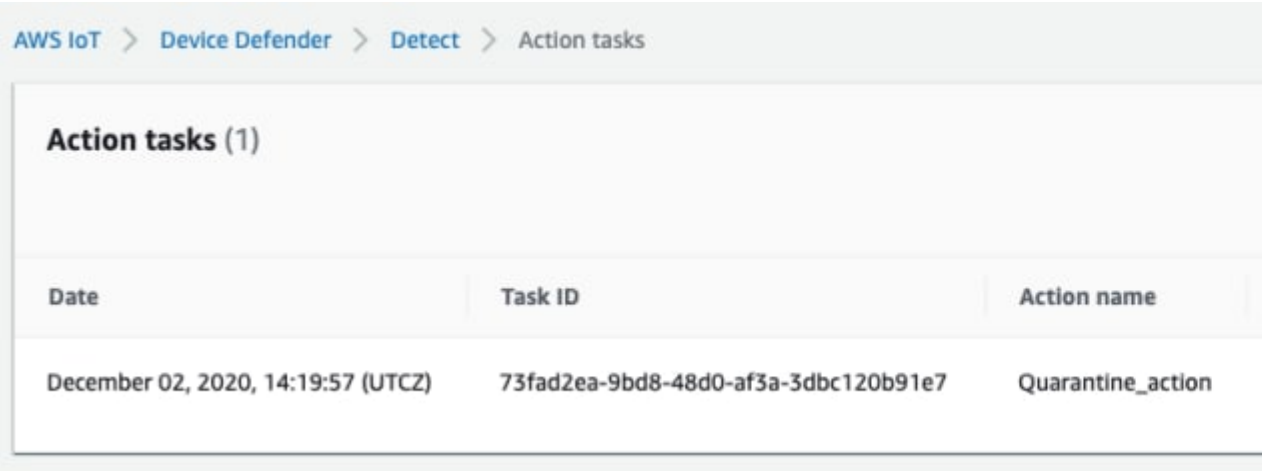

AWS IoT > Device Defender > Detect > Alarms
Alarms
Active (5)
History
Published alarms (5+)
Confidence
First Event
Thing name
March 19, 2021, 10:15:00 (UTC-0700)
underTest_shortAcceptanceTestAllBytesOut
March 19, 2021, 10:25:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
March 19, 2021, 10:25:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
HIGH
March 19, 2021, 10:40:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa
HIGH
March 19, 2021, 10:40:00 (UTC-0700)
underTest_shortAcceptanceTestNumMessag
7d7f97da535f54fa

Sélectionnez le périphérique que vous souhaitez déplacer vers le groupe de quarantaine et choisissezDémarrer des actions d'atténuation.

6. UnderDémarrer des actions d'atténuation,Actions de démarrageSélectionnez l'action d'atténuation que vous avez créée précédemment. Par exemple, nous choisirons**`Quarantine_action`**, puis.Démarrer. La page Tâches d'action s'ouvre.



7. L'appareil est maintenant isolé dans**`Quarantine_group`**et vous pouvez rechercher la cause première du problème qui a déclenché l'alarme. Une fois l'enquête terminée, vous pouvez déplacer l'appareil hors du groupe de choses ou prendre d'autres mesures.



Comment utiliser ML Detect avec l'interface de ligne de commande

Vous trouverez ci-dessous comment configurer ML Detect à l'aide de l'interface de ligne de commande.

Didacticiels

- Activer la détection de (p. 838)
- Surveiller l'état de votre modèle ML (p. 839)
- Passez en revue vos alarmes ML Detect (p. 841)
- Ajustez vos alarmes ML (p. 842)
- Atténuer les problèmes de périphériques (p. 844)

Activer la détection de

La procédure suivante vous montre comment activer ML Detect dans laAWS CLI.

1. Assurez-vous que vos appareils vont créer les points de données minimum requis comme défini dansML Detect configuration minimale (p. 933)pour la formation continue et l'actualisation du modèle. Pour que la collecte des données progresse, assurez-vous que vos objets sont dans un groupe d'objets attaché à un profil de sécurité.
2. Créez un profil de sécurité ML Detect à l'aide de l'outil`create-security-profile`Commande. L'exemple suivant crée un profil de sécurité nommé*`profil-sécurité-pour-smart-lights`*qui vérifie le nombre de messages envoyés, le nombre d'échecs d'autorisation, le nombre de tentatives de connexion et le nombre de déconnexions. L'exemple utilise`mlDetectionConfig`pour établir que la mesure utilisera le modèle ML Detect.

```
aws iot create-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
    "name": "num-messages-sent-ml-behavior",
    "metric": "aws:num-messages-sent",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-authorization-failures-ml-behavior",
    "metric": "aws:num-authorization-failures",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-connection-attempts-ml-behavior",
    "metric": "aws:num-connection-attempts",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
```

```
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  },
  {
    "name": "num-disconnects-ml-behavior",
    "metric": "aws:num-disconnects",
    "criteria": {
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "mlDetectionConfig": {
        "confidenceLevel": "HIGH"
      }
    },
    "suppressAlerts": true
  }]'
```

File d'attente:

```
{
    "securityProfileName": "security-profile-for-smart-lights",
    "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-
profile-for-smart-lights"
  }
```

3. Associez ensuite votre profil de sécurité à un ou plusieurs groupes de choses. Utilisation de l'attach-security-profilePour attacher un groupe d'objets à votre Profil de sécurité. L'exemple suivant associe un groupe de choses nomméML_DETECT_BETA_STAC_GROUPEavec leprofil-sécurité-pour-smart-lightsProfil de sécurité.

```
aws iot attach-security-profile \
--security-profile-name security-profile-for-smart-lights \
--security-profile-target-arn arn:aws:iot:eu-
west-1:123456789012:thinggroup/ML_Detect_beta_static_group
```

File d'attente:

Aucun.

4. Une fois que vous avez créé votre profil de sécurité complet, le modèle ML commence la formation. La formation initiale et la construction du modèle ML prennent 14 jours. Après 14 jours, s'il y a une activité anormale sur votre appareil, vous pouvez vous attendre à voir des alarmes.

Surveiller l'état de votre modèle ML

La procédure suivante vous montre comment surveiller vos modèles ML formation en cours.

- Utilisation de l'get-behavior-model-training-summariespour afficher la progression de votre modèle ML. L'exemple suivant obtient le résumé de la progression de la formation du modèle ML pour leprofil-sécurité-pour-smart-lightsProfil de sécurité.modelStatusvous indique si un modèle a terminé la formation ou est toujours en attente de construction pour un comportement particulier.

```
aws iot get-behavior-model-training-summaries \
   --security-profile-name security-profile-for-smart-lights
```

File d'attente:


```
{
    "summaries": [
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_sent_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.408,
            "lastModelRefreshDate": "2020-12-07T14:35:19.237000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Messages_received_ML_behavior",
            "modelStatus": "PENDING_BUILD",
            "datapointsCollectionPercentage": 0.0
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Authorization_failures_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.464,
            "lastModelRefreshDate": "2020-12-07T14:29:44.396000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Message_size_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 29.332,
            "lastModelRefreshDate": "2020-12-07T14:30:44.113000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Connection_attempts_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 32.891999999999996,
            "lastModelRefreshDate": "2020-12-07T14:29:43.121000-08:00"
        },
        {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "Disconnects_ML_behavior",
            "trainingDataCollectionStartDate": "2020-11-30T14:00:00-08:00",
            "modelStatus": "ACTIVE",
            "datapointsCollectionPercentage": 35.46,
            "lastModelRefreshDate": "2020-12-07T14:29:55.556000-08:00"
        }
    ]
}
```


Note

Si votre modèle ne progresse pas comme prévu, assurez-vous que vos appareils respectent laConfiguration requise (p. 933).

Passez en revue vos alarmes ML Detect

Une fois vos modèles ML construits et prêts pour l'évaluation des données, vous pouvez consulter régulièrement toutes les alarmes déduites par les modèles. La procédure suivante vous montre comment afficher vos alarmes dans laAWS CLI.

- Pour afficher toutes les alarmes actives, utilisez la commande`list-active-violations`Commande.

```
aws iot list-active-violations \
--max-results 2
```

File d'attente:

```
{
    "activeViolations": []
}
```

Vous pouvez également afficher toutes les violations découvertes au cours d'une période donnée à l'aide de l'outil`list-violation-events`Commande. L'exemple suivant répertorie les événements de violation du 22 septembre 2020 5:42:13 GMT au 26 octobre 2020 5:42:13 GMT.

```
aws iot list-violation-events \
    --start-time 1599500533 \
    --end-time 1600796533 \
    --max-results 2
```

File d'attente:

```
{
    "violationEvents": [
        {
            "violationId": "1448be98c09c3d4ab7cb9b6f3ece65d6",
            "thingName": "lightbulb-1",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.29
        },
        {
            "violationId": "df4537569ef23efb1c029a433ae84b52",
            "thingName": "lightbulb-2",
            "securityProfileName": "security-profile-for-smart-lights",
            "behavior": {
                "name": "LowConfidence_MladBehavior_MessagesSent",
                "metric": "aws:num-messages-sent",
                "criteria": {
                    "consecutiveDatapointsToAlarm": 1,
                    "consecutiveDatapointsToClear": 1,
```

```
                    "mlDetectionConfig": {
                        "confidenceLevel": "HIGH"
                    }
                },
                "suppressAlerts": true
            },
            "violationEventType": "alarm-invalidated",
            "violationEventTime": 1600780245.281
        }
    ],
    "nextToken":
 "Amo6XIUrsOohsojuIG6TuwSR3X9iUvH2OCksBZg6bed2j21VSnD1uP1pflxKX1+a3cvBRSosIB0xFv40kM6RYBknZ/
vxabMe/ZW31Ps/WiZHlr9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/
eoV9j7P/wveu7skNGepU/mvpV0O2Ap7hnV5U+Prx/9+iJA/341va
+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38hOQWEpz2xwFWAxAARxeIxCxt5c37RK/lRZBlhYqoB
+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD9O5v8idkTZzBxW2jrbzSUIdafPtsZHL/
yAMKr3HAKtaABz2nTsOBNre7X2d/jIjjarhon0Dh9l+8I9Y5Ey
+DIFBcqFTvhibKAafQt3gs6CUiqHdWiCenfJyb8whmDE2qxvdxGElGmRb
+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czkIFBpMppHj9cetHPvM
+qffXGAzKi8tL6eQuCdMLXmVE3jbqcJcjk9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

Ajustez vos alarmes ML

Une fois vos modèles ML construits et prêts pour l'évaluation des données, vous pouvez mettre à jour les paramètres de comportement ML de votre profil de sécurité pour modifier la configuration. La procédure suivante vous montre comment mettre à jour les paramètres de comportement ML de votre profil de sécurité dans laAWS CLI.

- Pour modifier les paramètres de comportement ML de votre profil de sécurité, utilisez l'outil`update-security-profile`Commande. L'exemple suivant met à jour l'*`profil-sécurité-pour-smart-lights`*Comportements du profil de sécurité en modifiant le`confidenceLevel`de quelques-uns des comportements et supprime les notifications pour tous les comportements.

```
aws iot update-security-profile \
    --security-profile-name security-profile-for-smart-lights \
    --behaviors \
     '[{
      "name": "num-messages-sent-ml-behavior",
      "metric": "aws:num-messages-sent",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-authorization-failures-ml-behavior",
      "metric": "aws:num-authorization-failures",
      "criteria": {
          "mlDetectionConfig": {
              "confidenceLevel" : "HIGH"
          }
      },
      "suppressAlerts": false
  },
  {
      "name": "num-connection-attempts-ml-behavior",
      "metric": "aws:num-connection-attempts",
      "criteria": {
          "mlDetectionConfig": {
```

```
            "confidenceLevel" : "HIGH"
        }
    },
    "suppressAlerts": false
},
{
    "name": "num-disconnects-ml-behavior",
    "metric": "aws:num-disconnects",
    "criteria": {
        "mlDetectionConfig": {
            "confidenceLevel" : "LOW"
        }
    },
    "suppressAlerts": false

}]'
```

File d'attente:

```
{
   "securityProfileName": "security-profile-for-smart-lights",
   "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-
profile-for-smart-lights",
   "behaviors": [
       {
           "name": "num-messages-sent-ml-behavior",
           "metric": "aws:num-messages-sent",
           "criteria": {
               "mlDetectionConfig": {
                   "confidenceLevel": "HIGH"
               }
           }
       },
       {
           "name": "num-authorization-failures-ml-behavior",
           "metric": "aws:num-authorization-failures",
           "criteria": {
               "mlDetectionConfig": {
                   "confidenceLevel": "HIGH"
               }
           }
       },
       {
           "name": "num-connection-attempts-ml-behavior",
           "metric": "aws:num-connection-attempts",
           "criteria": {
               "mlDetectionConfig": {
                   "confidenceLevel": "HIGH"
               }
           },
           "suppressAlerts": false
       },
       {
           "name": "num-disconnects-ml-behavior",
           "metric": "aws:num-disconnects",
           "criteria": {
               "mlDetectionConfig": {
                   "confidenceLevel": "LOW"
               }
           },
           "suppressAlerts": true
       }
   ],
   "version": 2,
```

```
    "creationDate": 1600799559.249,
    "lastModifiedDate": 1600800516.856
}
```

Atténuer les problèmes de périphériques

1. Utilisation de l'`create-thing-group`Commande pour créer un groupe d'objets pour l'action d'atténuation. Dans l'exemple suivant, nous créons un groupe de choses appeléThingGroupForDetectMitigationAction.

```
aws iot create-thing-group —thing-group-name ThingGroupForDetectMitigationAction
```

File d'attente:

```
{
 "thingGroupName": "ThingGroupForDetectMitigationAction",
 "thingGroupArn": "arn:aws:iot:us-
east-1:123456789012:thinggroup/ThingGroupForDetectMitigationAction",
 "thingGroupId": "4139cd61-10fa-4c40-b867-0fc6209dca4d"
}
```

2. Ensuite, utilisez la stratégie`create-mitigation-action`pour créer une action d'atténuation. Dans l'exemple suivant, nous créons une action d'atténuation appeléedetect_mitigation_actionAvec l'ARN du rôle IAM utilisé pour appliquer l'action d'atténuation. Nous définissons également le type d'action et les paramètres de cette action. Dans ce cas, notre atténuation déplacera les choses vers notre groupe de choses créé précédemment appeléThingGroupForDetectMitigationAction.

```
aws iot create-mitigation-action --action-name detect_mitigation_action \
--role-arn arn:aws:iam::123456789012:role/MitigationActionValidRole \
--action-params \
'{
     "addThingsToThingGroupParams": {
         "thingGroupNames": ["ThingGroupForDetectMitigationAction"],
         "overrideDynamicGroups": false
     }
 }'
```

File d'attente:

```
{
 "actionArn": "arn:aws:iot:us-
east-1:123456789012:mitigationaction/detect_mitigation_action",
 "actionId": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3"
}
```

3. Utilisation de l'`start-detect-mitigation-actions-task`Pour démarrer votre tâche d'actions d'atténuation.`task-id`,`target`and`actions`sont des paramètres obligatoires.

```
aws iot start-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction \
    --target '{ "violationIds" : [ "violationId-1", "violationId-2" ] }' \
    --actions "detect_mitigation_action" \
    --include-only-active-violations \
    --include-suppressed-alerts
```

File d'attente:

```
{
    "taskId": "taskIdForMitigationAction"
}
```

4. (Facultatif) Pour afficher les exécutions d'actions d'atténuation incluses dans une tâche, utilisez l'outil`list-detect-mitigation-actions-executions`Commande.

```
aws iot list-detect-mitigation-actions-executions \
    --task-id taskIdForMitigationAction \
    --max-items 5 \
    --page-size 4
```

File d'attente:

```
{
    "actionsExecutions": [
        {
            "taskId": "e56ee95e - f4e7 - 459 c - b60a - 2701784290 af",
            "violationId": "214_fe0d92d21ee8112a6cf1724049d80",
            "actionName": "underTest_MAThingGroup71232127",
            "thingName": "cancelDetectMitigationActionsTaskd143821b",
            "executionStartDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "executionEndDate": "Thu Jan 07 18: 35: 21 UTC 2021",
            "status": "SUCCESSFUL",
        }
    ]
}
```

5. (Facultatif) Utilisez l'option`describe-detect-mitigation-actions-task`Commande d'atténuation pour obtenir des informations sur une tâche d'action d'atténuation.

```
aws iot describe-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

File d'attente:

```
{
    "taskSummary": {
        "taskId": "taskIdForMitigationAction",
        "taskStatus": "SUCCESSFUL",
        "taskStartTime": 1609988361.224,
        "taskEndTime": 1609988362.281,
        "target": {
            "securityProfileName": "security-profile-for-smart-lights",
            "behaviorName": "num-messages-sent-ml-behavior"
        },
        "violationEventOccurrenceRange": {
            "startTime": 1609986633.0,
            "endTime": 1609987833.0
        },
        "onlyActiveViolationsIncluded": true,
        "suppressedAlertsIncluded": true,
        "actionsDefinition": [
            {
                "name": "detect_mitigation_action",
                "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                "roleArn": "arn:aws:iam::123456789012:role/MitigatioActionValidRole",
                "actionParams": {
                    "addThingsToThingGroupParams": {
                        "thingGroupNames": [
```

```
                            "ThingGroupForDetectMitigationAction"
                        ],
                        "overrideDynamicGroups": false
                    }
                }
            }
        ],
        "taskStatistics": {
            "actionsExecuted": 0,
            "actionsSkipped": 0,
            "actionsFailed": 0
        }
    }
}
```

6. (Facultatif) Pour obtenir une liste de vos tâches d'actions d'atténuation, utilisez l'outil`list-detect-mitigation-actions-tasks`Commande.

```
aws iot list-detect-mitigation-actions-tasks \
    --start-time 1609985315 \
    --end-time 1609988915 \
    --max-items 5 \
    --page-size 4
```

File d'attente:

```
{
    "tasks": [
        {
            "taskId": "taskIdForMitigationAction",
            "taskStatus": "SUCCESSFUL",
            "taskStartTime": 1609988361.224,
            "taskEndTime": 1609988362.281,
            "target": {
                "securityProfileName": "security-profile-for-smart-lights",
                "behaviorName": "num-messages-sent-ml-behavior"
            },
            "violationEventOccurrenceRange": {
                "startTime": 1609986633.0,
                "endTime": 1609987833.0
            },
            "onlyActiveViolationsIncluded": true,
            "suppressedAlertsIncluded": true,
            "actionsDefinition": [
                {
                    "name": "detect_mitigation_action",
                    "id": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3",
                    "roleArn": "arn:aws:iam::123456789012:role/
MitigatioActionValidRole",
                    "actionParams": {
                        "addThingsToThingGroupParams": {
                            "thingGroupNames": [
                                "ThingGroupForDetectMitigationAction"
                            ],
                            "overrideDynamicGroups": false
                        }
                    }
                }
            ],
            "taskStatistics": {
                "actionsExecuted": 0,
                "actionsSkipped": 0,
                "actionsFailed": 0
```

```
            }
        }
    ]
}
```

7. (Facultatif) Pour annuler une tâche d'actions d'atténuation, utilisez l'outil`cancel-detect-mitigation-actions-task`Commande.

```
aws iot cancel-detect-mitigation-actions-task \
    --task-id taskIdForMitigationAction
```

File d'attente:

Aucun.

Personnaliser quand et comment vous affichezAWS IoT Device DefenderRésultats de l'audit

AWS IoT Device Defenderl'audit fournit des contrôles de sécurité périodiques pour confirmerAWS IoTles périphériques et les ressources suivent les meilleures pratiques. Pour chaque vérification, les résultats de l'audit sont classés comme conformes ou non conformes, lorsque la non-conformité entraîne des icônes d'avertissement de la console. Pour réduire le bruit lié à la répétition de problèmes connus, la fonction de suppression de recherche d'audit vous permet de réduire temporairement ces notifications de non-conformité.

Vous pouvez supprimer certaines vérifications d'audit pour une ressource ou un compte spécifique pour une période prédéterminée. Un résultat de vérification qui a été supprimé est classé comme une constatation supprimée, distincte des catégories conformes et non conformes. Cette nouvelle catégorie ne déclenche pas une alarme comme un résultat non conforme. Cela vous permet de réduire les perturbations de notification de non-conformité pendant les périodes de maintenance connues ou jusqu'à ce qu'une mise à jour soit planifiée.

Mise en route

Les sections suivantes détaillent comment utiliser les suppressions de recherche d'audit pour supprimer un`Device certificate expiring`Vérifiez dans la console et l'interface de ligne de commande. Si vous souhaitez suivre l'une ou l'autre des démonstrations, vous devez d'abord créer deux certificats expirant pour que Device Defender détecte.

Utilisez ce qui suit pour créer vos certificats.

- Création et enregistrement d'un certificat d'autorité de certification (p. 240)
- Création d'un certificat client à l'aide de votre certificat d'autorité de certification. À l'étape 3, définissez votre`days`Paramètre à`1`.

Si vous utilisez l'interface de ligne de commande pour créer vos certificats, entrez la commande suivante.

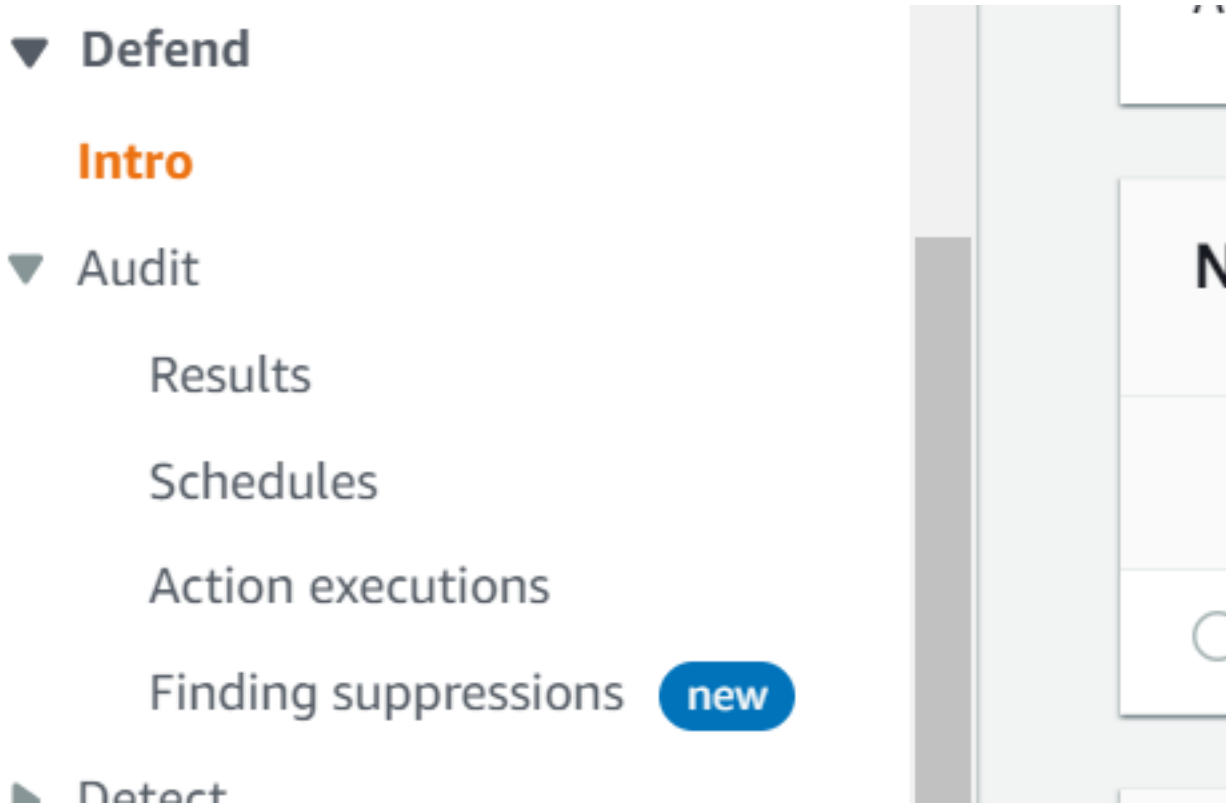
```
openssl x509 -req \
    -in device_cert_csr_filename \
    -CA root_ca_pem_filename \
    -CAkey root_ca_key_filename \
    -CAcreateserial \
    -out device_cert_pem_filename \
    -days 1 -sha256
```

Personnalisez vos résultats d'audit dans la console

La procédure pas à pas suivante utilise un compte avec deux certificats d'appareil expirés qui déclenchent une vérification d'audit non conforme. Dans ce scénario, nous voulons désactiver l'avertissement car nos développeurs testent une nouvelle fonctionnalité qui permettra de résoudre le problème. Nous créons une suppression de constatation d'audit pour chaque certificat afin d'empêcher que le résultat de l'audit ne soit pas conforme pour la semaine suivante.

1. Nous allons d'abord exécuter un audit à la demande pour montrer que la vérification du certificat de périphérique expirée n'est pas conforme.

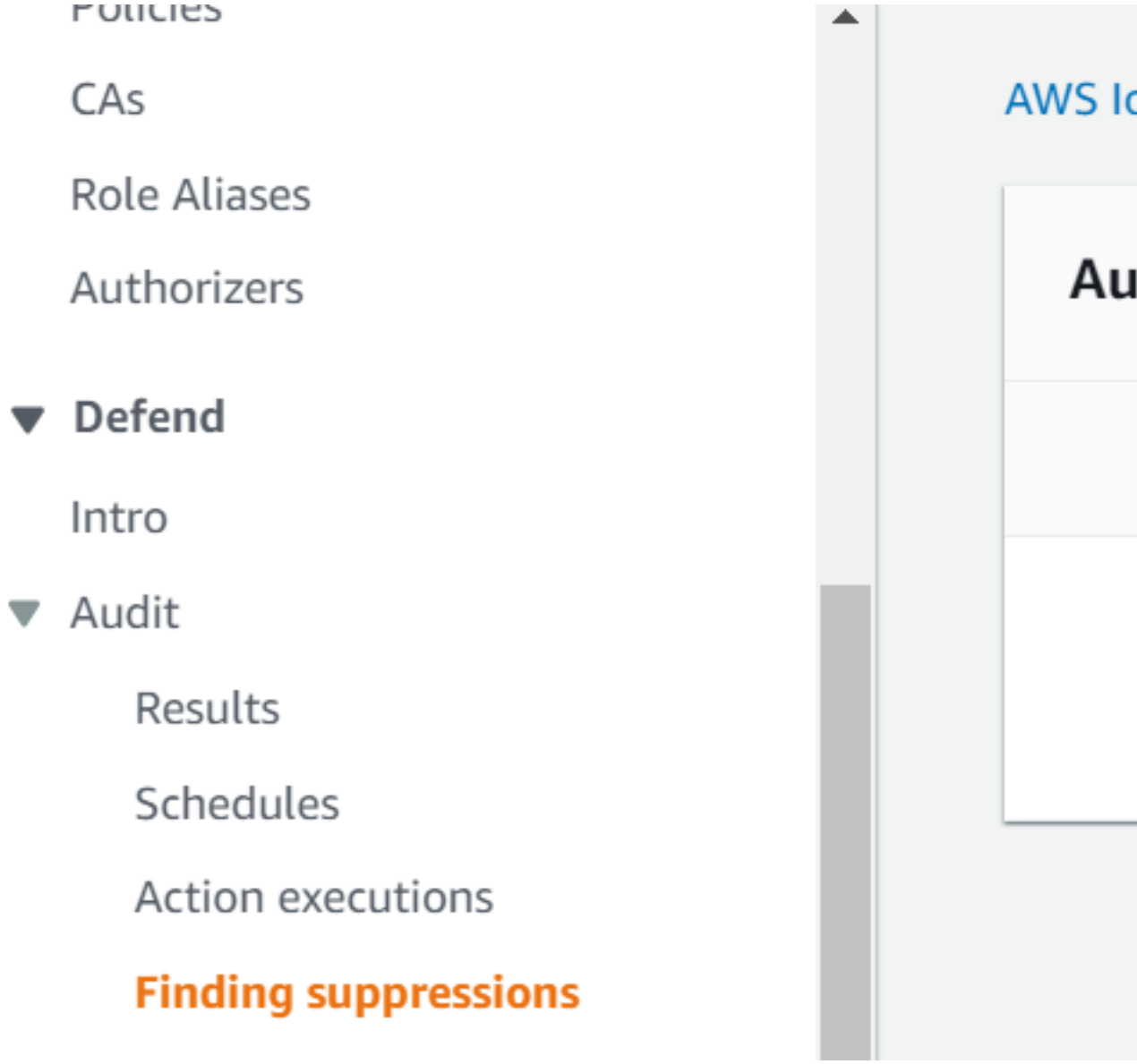
   À partir de laAWS IoTconsole, choisissezDéfendreà partir de la barre latérale gauche, puisAudit, puisRésultats. Dans la pageRésultats de l'audit, choisissezCréer. La .Créer un nouvel audits'ouvre. Sélectionnez Créer.



   À partir des résultats de l'audit à la demande, nous pouvons voir que « Certificat de périphérique expirant » n'est pas conforme pour deux ressources.

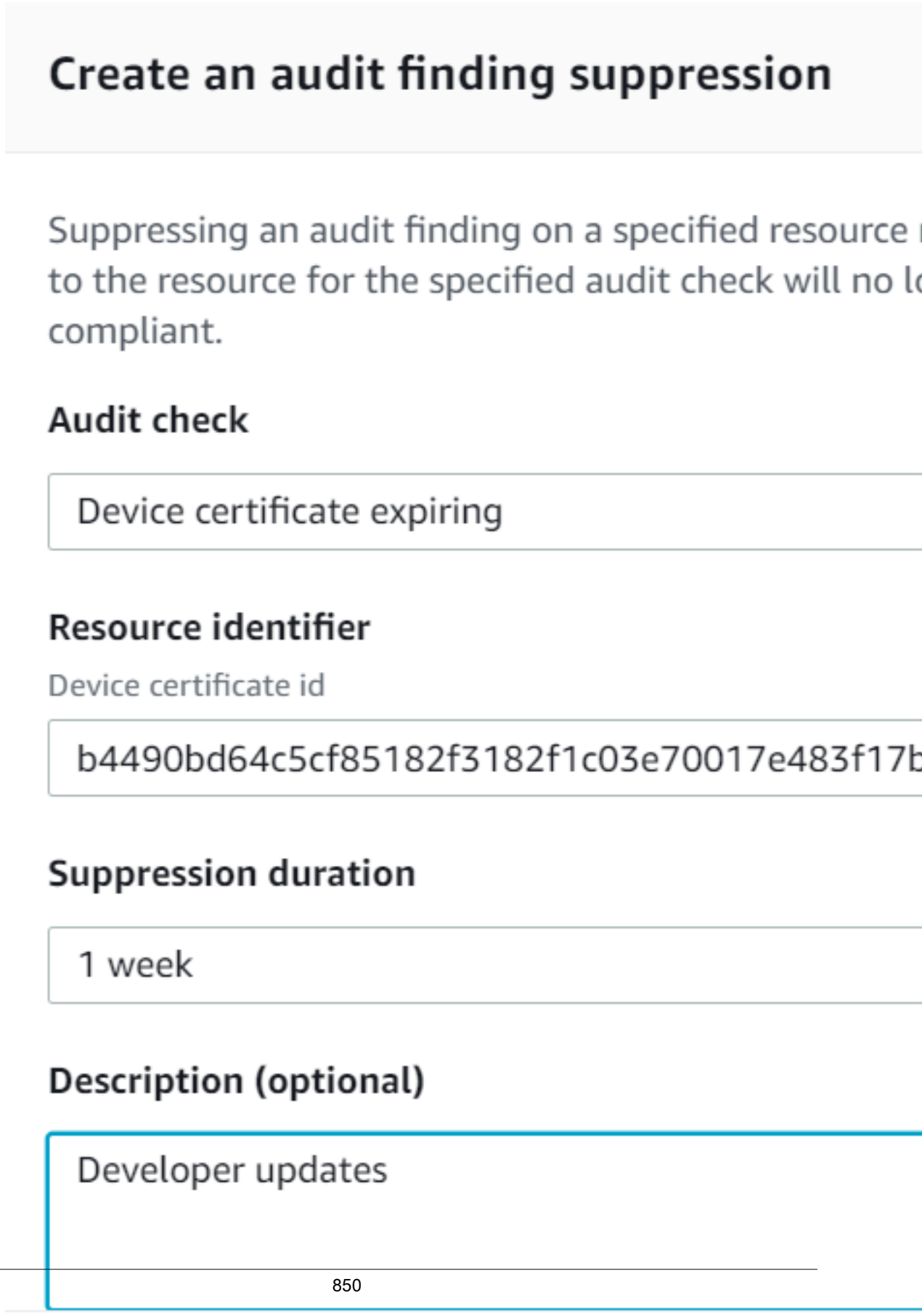
2. Maintenant, nous aimerions désactiver l'avertissement de vérification non conforme « Certificat de périphérique expirant » car nos développeurs testent de nouvelles fonctionnalités qui corrigeront l'avertissement.

   À partir de la barre latérale de gauche sousDéfendre, choisissezAudit, puis.Recherche de suppressions. Dans la pageSuppressions des résultats d'audit, choisissezCréer.



3. Dans la pageCréer une suppression de recherche d'audit, nous devons remplir ce qui suit.

   - Contrôle d'audit : Il est sélectionné :`Device certificate expiring`, car c'est la vérification d'audit que nous aimerions supprimer.
   - Identificateur de ressource : Nous entrons l'ID de certificat de périphérique de l'un des certificats pour lesquels nous aimerions supprimer les résultats d'audit.
   - Durée de suppression : Il est sélectionné :`1 week`, parce que c'est combien de temps nous aimerions supprimer la`Device certificate expiring`vérification d'audit pour.
   - Description (facultative) : Nous ajoutons une note décrivant les raisons pour lesquelles nous supprimons cette constatation d'audit.

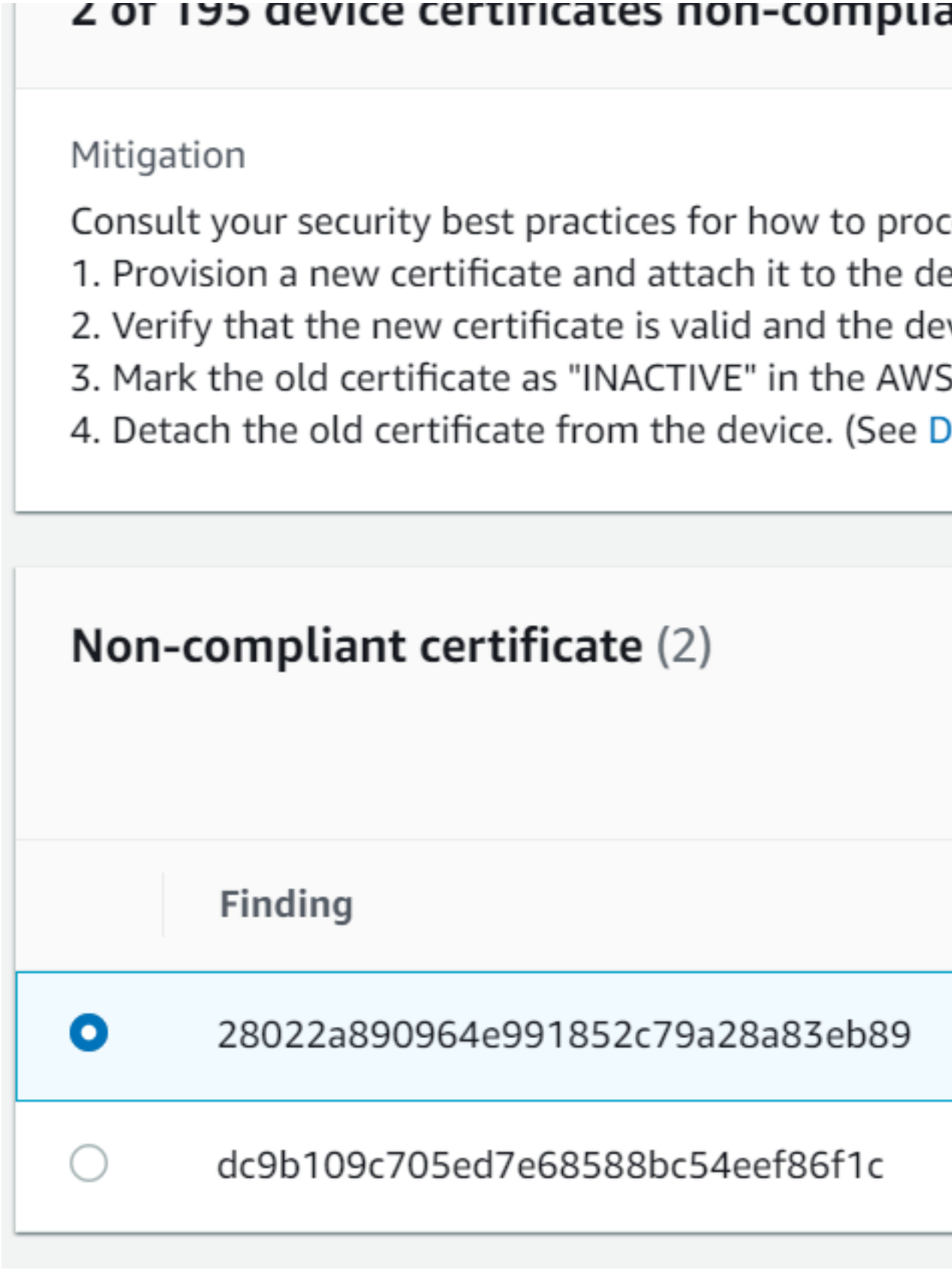

Create an audit finding suppression
Suppressing an audit finding on a specified resource
to the resource for the specified audit check will no l
compliant.
Audit check
Device certificate expiring
Resource identifier
Device certificate id
b4490bd64c5cf85182f3182f1c03e70017e483f17b
Suppression duration
1 week
Description (optional)
Developer updates

Une fois que nous remplissons les champs, choisissezCréer. Une bannière de réussite apparaît après la création de la suppression des constatations d'audit.

4. Nous avons supprimé une constatation d'audit pour l'un des certificats et maintenant nous devons supprimer la constatation d'audit pour le second certificat. Nous pourrions utiliser la même méthode de suppression que nous avons utilisée à l'étape 3, mais nous utiliserons une méthode différente à des fins de démonstration.

   À partir de la barre latérale de gauche sousDéfendre, choisissezAudit, puis.Résultats. Dans la pageRésultats de l'audit, choisissez l'audit avec la ressource non conforme. Ensuite, sélectionnez la ressource sousContrôles non conformes. Dans notre cas, nous sélectionnons « Certificat d'appareil expirant ».

5. Dans la pageCertificat de l'appareil expirant, sousNon conforme à la politiqueChoisissez le bouton d'option en regard de la constatation qui doit être supprimée. Ensuite, choisissez laActions, puis choisissez la durée pour laquelle vous souhaitez supprimer la recherche. Dans notre cas, nous choisissons`1 week`comme nous l'avons fait pour l'autre certificat. Dans la pageConfirm la suppression, choisissezActiver la suppression.


2 of 195 device certificates non-compli
Mitigation
Consult your security best practices for how to proc
1. Provision a new certificate and attach it to the de
2. Verify that the new certificate is valid and the dev
3. Mark the old certificate as "INACTIVE" in the AWS
4. Detach the old certificate from the device. (See D
Non-compliant certificate (2)
Finding
28022a890964e991852c79a28a83eb89
dc9b109c705ed7e68588bc54eef86f1c

Une bannière de réussite apparaît après la création de la suppression des constatations d'audit. Maintenant, les deux constatations d'audit ont été supprimées pendant une semaine pendant que nos développeurs travaillent sur une solution pour répondre à l'avertissement.

Personnalisez vos conclusions d'audit dans l'interface de ligne de commande

La procédure pas à pas suivante utilise un compte avec un certificat de périphérique expiré qui déclenche une vérification d'audit non conforme. Dans ce scénario, nous voulons désactiver l'avertissement car nos développeurs testent une nouvelle fonctionnalité qui permettra de résoudre le problème. Nous créons une suppression de constatation d'audit pour le certificat afin d'empêcher que le résultat de l'audit ne soit pas conforme pour la semaine suivante.

Nous utilisons les commandes CLI suivantes.

- création-audit-suppression
- describe-audit-suppression
- update-audit-suppression
- supprimer-audit-suppression
- list-audit-suppressions

1. Utilisez la commande suivante pour activer l'audit.

```
aws iot update-account-audit-configuration \
   --audit-check-configurations "{\"DEVICE_CERTIFICATE_EXPIRING_CHECK\":{\"enabled
\":true}}"
```

File d'attente:

Aucun.

2. Utilisez la commande suivante pour exécuter un audit à la demande qui cible la`DEVICE_CERTIFICATE_EXPIRING_CHECK`Contrôle d'audit.

```
aws iot start-on-demand-audit-task \
    --target-check-names DEVICE_CERTIFICATE_EXPIRING_CHECK
```

File d'attente:

```
{
    "taskId": "787ed873b69cb4d6cdbae6ddd06996c5"
}
```

3. Utilisation de l'describe-account-audit-configurationpour décrire la configuration de l'audit. Nous voulons confirmer que nous avons activé la vérification d'audit pour`DEVICE_CERTIFICATE_EXPIRING_CHECK`.

```
aws iot describe-account-audit-configuration
```

File d'attente:

```
{
```

```
    "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
    "auditNotificationTargetConfigurations": {
        "SNS": {
            "targetArn": "arn:aws:sns:us-east-1:<accountid>:project_sns",
            "roleArn": "arn:aws:iam::<accountid>:role/service-role/project",
            "enabled": true
        }
    },
    "auditCheckConfigurations": {
        "AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": false
        },
        "CA_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "CONFLICTING_CLIENT_IDS_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "enabled": true
        },
        "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK": {
            "enabled": false
        },
        "DEVICE_CERTIFICATE_SHARED_CHECK": {
            "enabled": false
        },
        "IOT_POLICY_OVERLY_PERMISSIVE_CHECK": {
            "enabled": true
        },
        "IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK": {
            "enabled": false
        },
        "IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        },
        "LOGGING_DISABLED_CHECK": {
            "enabled": false
        },
        "REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK": {
            "enabled": false
        },
        "UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
            "enabled": false
        }
    }
}
```

`DEVICE_CERTIFICATE_EXPIRING_CHECK`doit avoir une valeur de`true`.

4. Utilisation de l'liste-audit-tâchepour identifier les tâches d'audit terminées.

```
aws iot list-audit-tasks \
    --task-status "COMPLETED" \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

File d'attente:

```
{
    "tasks": [
        {
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "taskStatus": "COMPLETED",
            "taskType": "SCHEDULED_AUDIT_TASK"
        }
    ]
}
```

La .`taskId`de l'audit que vous avez exécuté à l'étape 1 devrait avoir un`taskStatus`de`COMPLETED`.

5. Utilisation de l'describe-audit-taskPour obtenir des détails sur l'audit terminé à l'aide de la`taskId`Sortie de l'étape précédente. Cette commande répertorie les détails de votre audit.

```
aws iot describe-audit-task \
    --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

File d'attente:

```
{
    "taskStatus": "COMPLETED",
    "taskType": "SCHEDULED_AUDIT_TASK",
    "taskStartTime": 1596168096.157,
    "taskStatistics": {
        "totalChecks": 1,
        "inProgressChecks": 0,
        "waitingForDataCollectionChecks": 0,
        "compliantChecks": 0,
        "nonCompliantChecks": 1,
        "failedChecks": 0,
        "canceledChecks": 0
    },
    "scheduledAuditName": "AWSIoTDeviceDefenderDailyAudit",
    "auditDetails": {
        "DEVICE_CERTIFICATE_EXPIRING_CHECK": {
            "checkRunStatus": "COMPLETED_NON_COMPLIANT",
            "checkCompliant": false,
            "totalResourcesCount": 195,
            "nonCompliantResourcesCount": 2
        }
    }
}
```

6. Utilisation de l'liste-vérification-constatspour trouver l'ID de certificat non conforme afin que nous puissions suspendre les alertes d'audit pour cette ressource.

```
aws iot list-audit-findings \
    --start-time 2020-07-31 \
    --end-time 2020-08-01
```

File d'attente:

```
{
    "findings": [
        {
            "findingId": "296ccd39f806bf9d8f8de20d0ceb33a1",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
```

```
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "b4490<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1582862626000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        },
        {
            "findingId": "37ecb79b7afb53deb328ec78e647631c",
            "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "taskStartTime": 1596168096.157,
            "findingTime": 1596168096.651,
            "severity": "MEDIUM",
            "nonCompliantResource": {
                "resourceType": "DEVICE_CERTIFICATE",
                "resourceIdentifier": {
                    "deviceCertificateId": "c7691<shortened>"
                },
                "additionalInfo": {
                "EXPIRATION_TIME": "1583424717000"
                }
            },
            "reasonForNonCompliance": "Certificate is past its expiration.",
            "reasonForNonComplianceCode": "CERTIFICATE_PAST_EXPIRATION",
            "isSuppressed": false
        }
    ]
}
```

7. Utilisation de l'création-audit-suppressionpour supprimer les notifications pour le`DEVICE_CERTIFICATE_EXPIRING_CHECK`vérification d'audit pour un certificat de périphérique avec l'identifiant*c7691e<shortened>*jusqu'à*20-08-20*.

```
aws iot create-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>" \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-20
```

8. Utilisation de l'suppression de liste-audit-pour confirmer le paramètre de suppression d'audit et obtenir des détails sur la suppression.

```
aws iot list-audit-suppressions
```

File d'attente:

```
{
    "suppressions": [
        {
        "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",
            "resourceIdentifier": {
                "deviceCertificateId": "c7691e<shortened>"
            },
        "expirationDate": 1597881600.0,
```

```
        "suppressIndefinitely": false
        }
    ]
}
```

9. La .update-audit-suppressionpeut être utilisée pour mettre à jour la suppression de recherche d'audit. L'exemple ci-dessous met à jour le`expiration-date`sur`08/21/20`.

```
aws iot update-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId=c7691e<shortened> \
    --no-suppress-indefinitely \
    --expiration-date 2020-08-21
```

10. La .supprimer-audit-suppressionpeut être utilisée pour supprimer une suppression de recherche d'audit.

```
aws iot delete-audit-suppression \
    --check-name DEVICE_CERTIFICATE_EXPIRING_CHECK \
    --resource-identifier deviceCertificateId="c7691e<shortened>"
```

Pour confirmer la suppression, utilisez l'interface de ligne de commandelist-audit-suppressionsCommande.

```
aws iot list-audit-suppressions
```

File d'attente:

```
{
 "suppressions": []
}
```

Dans ce didacticiel, nous vous avons montré comment supprimer une`Device certificate expiring`Vérifiez dans la console et l'interface de ligne de commande. Pour plus d'informations sur les suppressions de résultats d'audit, consultezSuppressions des résultats d'audit (p. 911)

Audit

Un audit AWS IoT Device Defender examine les paramètres et les stratégies liés aux comptes et aux appareils afin de vérifier que les mesures de sécurité sont en place. Un audit peut vous aider à détecter les écarts par rapport aux bonnes pratiques de sécurité ou aux stratégies d'accès, comme l'utilisation de la même identité ou des stratégies trop permissives qui autorisent un appareil à lire et mettre à jour des données pour beaucoup d'autres appareils. Vous pouvez exécuter les audits en fonction des besoins (audits à la demande) ou les planifier pour les exécuter régulièrement (audits planifiés).

Un audit AWS IoT Device Defender exécute un ensemble de contrôles prédéfinis pour vérifier les bonnes pratiques de sécurité IoT courantes et les vulnérabilités des appareils. Les contrôles prédéfinis portent, par exemple, sur les stratégies qui accordent les autorisations de lecture et de mise à jour des données sur plusieurs appareils, sur les appareils qui partagent une identité (certificat X.509) ou sur les certificats qui ont expiré ou ont été révoqués, mais qui sont toujours actifs.

Gravité du problème

La gravité du problème indique le niveau de préoccupation associé à chaque cas de non-conformité identifié ainsi que le délai recommandé pour la correction.

Critique

Les contrôles d'audit non conformes ayant ce niveau de gravité identifient les problèmes nécessitant une attention urgente. Les problèmes critiques permettent souvent aux personnes malveillantes avec peu de sophistication et sans connaissances d'initiés ou informations d'identifications spéciales d'accéder facilement à vos ressources ou de les contrôler.

Élevé

Les contrôles d'audit non conformes ayant ce niveau de gravité doivent être examinés en urgence et nécessitent que les mesures correctives nécessaires soient planifiées après résolution des problèmes critiques. Comme les problèmes de gravité critique, les problèmes de gravité élevée permettent souvent aux personnes malveillantes d'accéder à vos ressources ou de les contrôler. Cependant, les problèmes de gravité élevée sont souvent plus difficiles à exploiter. Ils peuvent nécessiter des outils spéciaux, des connaissances d'initiés ou des configurations spécifiques.

Medium

Les contrôles d'audit non conformes ayant ce niveau de gravité présentent des problèmes qui nécessitent votre attention dans le cadre de la gestion continue de votre posture de sécurité. Les problèmes de gravité moyenne peuvent avoir un impact opérationnel négatif, comme des pannes imprévues dues à un dysfonctionnement des contrôles de sécurité. Ces problèmes peuvent également fournir aux personnes malveillantes un accès limité à vos ressources ou un contrôle limité de vos ressources, ou faciliter certaines de leurs actions malveillantes.

Faible

Les contrôles d'audit non conformes ayant ce niveau de gravité indiquent souvent que les bonnes pratiques de sécurité ont été négligées ou contournées. Bien que ces erreurs ne puissent pas avoir d'impact immédiat sur la sécurité, elles peuvent être exploitées par des personnes malveillantes. Tout comme les problèmes de gravité moyenne, les problèmes de gravité faible nécessitent votre attention dans le cadre de la gestion continue de votre posture de sécurité.

Étapes suivantes

Pour connaître les types de vérification d'audit qui peuvent être effectuées, veuillez consulter Contrôles d'audit (p. 858). Pour obtenir des informations sur les quotas de service qui s'appliquent aux audits, veuillez consulter Quotas de service.

Contrôles d'audit

Note

Lorsque vous activez une vérification, la collecte des données démarre immédiatement. Si un grand nombre de données doit être collecté dans votre compte, les résultats du contrôle risquent de ne pas être disponibles immédiatement après l'activation.

Les contrôles d'audit suivants sont pris en charge :

Certificat d'autorité de certification révoqué mais certificats de périphérique toujours actifs

Un certificat CA a été révoqué, mais demeure actif dans AWS IoT.

Cette vérification s'affiche sous la forme`REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Un certificat CA est marqué comme étant révoqué dans la liste de révocation des certificats gérée par l'autorité d'émission mais est encore marqué comme étant ACTIVE (ACTIF) ou PENDING TRANSFER (EN ATTENTE DE TRANSFERT) dans AWS IoT.

Voici les codes de motif renvoyés lorsque ce contrôle trouve un certificat CA non conforme :

- CERTIFICATE_REVOKED_BY_ISSUER

Pourquoi est-ce important ?

Un certificat CA révoqué ne doit plus être utilisé pour signer des certificats d'appareil. Il peut avoir été révoqué car compromis. Les appareils nouvellement ajoutés avec des certificats signés à l'aide de ce certificat CA peuvent constituer une menace à la sécurité.

Comment réparer

1. Utilisez UpdateCACertificate pour marquer le certificat CA comme INACTIVE (INACTIF) dans AWS IoT . Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :
   - Appliquer l'action d'atténuation `UPDATE_CA_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation pour mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
2. Vérifiez l'activité d'enregistrement de certificat d'appareil pendant la période après laquelle le certificat de CA a été révoqué et envisagez de révoquer les certificats d'appareil qui ont pu être émis pendant cette période. Utilisez ListCertificatesByCA pour répertorier les certificats d'appareil signés par le certificat CA et UpdateCertificate pour révoquer un certificat d'appareil.

Certificat de périphérique partagé

Plusieurs connexions simultanées utilisent le même certificat X.509 pour s'authentifier auprès d'AWS IoT.

Cette vérification s'affiche sous la formeDEVICE_CERTIFICATE_SHARED_CHECKdans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Lorsqu'il est effectué dans le cadre d'un audit à la demande, ce contrôle examine les certificats et les ID client qui ont été utilisés par les appareils pour se connecter au cours des 31 jours avant le début de l'audit jusqu'à 2 heures avant l'exécution de la vérification. Pour les audits planifiés, ce contrôle examine les données entre 2 heures avant la dernière fois où l'audit a été exécuté et 2 heures avant le début de cette instance de l'audit. Si vous avez pris des mesures pour atténuer cette condition pendant la période contrôlée, notez à quel moment les connexions simultanées ont été effectuées pour déterminer si le problème persiste.

Les codes de motif sont renvoyés lorsque ce contrôle trouve un certificat non conforme :

- CERTIFICATE_SHARED_BY_MULTIPLE_DEVICES

En outre, les résultats renvoyés par ce contrôle incluent l'ID du certificat partagé, les ID des clients utilisant le certificat pour se connecter et les heures de connexion/déconnexion. Les résultats les plus récents sont répertoriés en premier.

Pourquoi est-ce important ?

Chaque appareil doit avoir un certificat unique pour s'authentifier auprès d'AWS IoT. Lorsque plusieurs appareils utilisent le même certificat, cela peut indiquer qu'un appareil est compromis. Son identité peut avoir été clonée pour compromettre davantage le système.

Comment réparer

Vérifiez que le certificat d'appareil n'a pas été compromis. S'il l'a été, suivez les bonnes pratiques en matière de sécurité pour traiter cette situation.

Si vous utilisez le même certificat sur plusieurs appareils, vous pouvez :

1. Allouer de nouveaux certificats uniques et les attacher à chaque appareil.
2. Vérifier que les nouveaux certificats sont valides et que les appareils peuvent les utiliser pour se connecter.
3. Utiliser UpdateCertificate pour marquer l'ancien certificat comme REVOKED (RÉVOQUÉ) dans AWS IoT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les opérations suivantes :
   - Appliquer l'action d'atténuation UPDATE_DEVICE_CERTIFICATE sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer l'action d'atténuation ADD_THINGS_TO_THING_GROUP pour ajouter le dispositif à un groupe où vous pouvez prendre des mesures à son égard.
   - Appliquer lePUBLISH_FINDINGS_TO_SNSAction d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
4. Détacher l'ancien certificat de chacun des appareils.

Qualité de la clé de l'appareil

Les clients AWS IoT s'appuient souvent sur l'authentification mutuelle TLS à l'aide de certificats X.509 pour s'authentifier auprès de l'agent de messages AWS IoT. Ces certificats et leurs certificats d'autorité de certification doivent être enregistrés dans leur compte AWS IoT avant d'être utilisés. AWS IoT effectue les

vérifications d'intégrité élémentaires sur ces certificats lorsqu'ils sont enregistrés. Ces vérifications portent notamment sur les éléments suivants :

- Le format des certificats doit être valide
- Les certificats doivent être signés par une autorité de certification enregistrée
- La période de validité des certificats ne doit pas avoir expiré.
- La taille des clé de chiffrement des certificats doit correspondre à une taille minimale requise (pour les clés RSA, elles doivent être de 2 048 bits ou plus).

Cette vérification d'audit fournit les tests supplémentaires suivants concernant la qualité de votre clé de chiffrement :

- CVE-2008-0166 — Vérifie si la clé a été générée à l'aide d'OpenSSL versions 0.9.8c-1 à 0.9.8g-9 (non incluse) sur un système d'exploitation basé sur Debian. Ces versions d'OpenSSL utilisent un générateur de nombres aléatoires qui génère des nombres prévisibles, ce qui facilite les attaques par force brute des clés de chiffrement menées par des personnes malveillantes.
- CVE-2017-15361 — Vérifie si la clé a été générée par la bibliothèque RSA Infineon 1.02.013 dans le microprogramme du module TPM (Trusted Platform Module) Infineon, par exemple les versions antérieures à 00000000000422 — 4.34, antérieures à 0000000000062b — 6.43 et antérieures à 00000000000000000000000000000422 — 133.33. Cette bibliothèque gère de manière incorrecte la génération de clés RSA, ce qui permet aux personnes malveillantes de vaincre plus facilement certains mécanismes de protection de chiffrement grâce à des attaques ciblées. BitLocker avec TPM 1.2, la génération de clés PGP YubiKey 4 (avant 4.3.5) et la fonction de chiffrement des données utilisateur mises en cache dans Chrome OS sont des exemples de technologies ayant été affectées.

AWS IoT Device Defender déclare les certificats non conformes s'ils échouent à ces tests.

Cette vérification s'affiche sous la forme`DEVICE_CERTIFICATE_KEY_QUALITY_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Ce contrôle s'applique aux certificats d'appareil qui sont ACTIVE ou PENDING_TRANSFER.

Les codes de motif sont renvoyés lorsque ce contrôle trouve un certificat non conforme :

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361
- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

Pourquoi est-ce important ?

Lorsqu'un appareil utilise un certificat vulnérable, les personnes malveillantes peuvent plus facilement le compromettre.

Comment réparer

Mettez à jour les certificats de vos appareils afin de remplacer ceux qui présentent des vulnérabilités connues.

Si vous utilisez le même certificat sur plusieurs appareils, vous pouvez :

1. Allouer de nouveaux certificats uniques et les attacher à chaque appareil.
2. Vérifier que les nouveaux certificats sont valides et que les appareils peuvent les utiliser pour se connecter.

3. Utiliser UpdateCertificate pour marquer l'ancien certificat comme REVOKED (RÉVOQUÉ) dans AWS IoT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :
   - Appliquer l'action d'atténuation `UPDATE_DEVICE_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer l'action d'atténuation `ADD_THINGS_TO_THING_GROUP` pour ajouter le dispositif à un groupe où vous pouvez prendre des mesures à son égard.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
4. Détacher l'ancien certificat de chacun des appareils.

Qualité de la clé de certificat CA

Les clients AWS IoT s'appuient souvent sur l'authentification mutuelle TLS à l'aide de certificats X.509 pour s'authentifier auprès de l'agent de messages AWS IoT. Ces certificats et leurs certificats d'autorité de certification doivent être enregistrés dans leur compte AWS IoT avant d'être utilisés. AWS IoT effectue des vérifications d'intégrité élémentaires sur ces certificats lorsqu'ils sont enregistrés, notamment :

- Le format des certificats doit être valide.
- La période de validité des certificats ne doit pas avoir expiré.
- La taille des clé de chiffrement des certificats doit correspondre à une taille minimale requise (pour les clés RSA, elles doivent être de 2048 bits ou plus).

Cette vérification d'audit fournit les tests supplémentaires suivants concernant la qualité de votre clé de chiffrement :

- CVE-2008-0166 — Vérifie si la clé a été générée à l'aide d'OpenSSL versions 0.9.8c-1 à 0.9.8g-9 (non incluse) sur un système d'exploitation basé sur Debian. Ces versions d'OpenSSL utilisent un générateur de nombres aléatoires qui génère des nombres prévisibles, ce qui facilite les attaques par force brute des clés de chiffrement menées par des personnes malveillantes.
- CVE-2017-15361 — Vérifie si la clé a été générée par la bibliothèque RSA Infineon 1.02.013 dans le microprogramme du module TPM (Trusted Platform Module) Infineon, par exemple les versions antérieures à 00000000000422 — 4.34, antérieures à 0000000000062b — 6.43 et antérieures à 000000000000000000000000000422 — 133.33. Cette bibliothèque gère de manière incorrecte la génération de clés RSA, ce qui permet aux personnes malveillantes de vaincre plus facilement certains mécanismes de protection de chiffrement grâce à des attaques ciblées. BitLocker avec TPM 1.2, la génération de clés PGP YubiKey 4 (avant 4.3.5) et la fonction de chiffrement des données utilisateur mises en cache dans Chrome OS sont des exemples de technologies ayant été affectées.

AWS IoT Device Defender déclare les certificats non conformes s'ils échouent à ces tests.

Cette vérification s'affiche sous la forme`CA_CERTIFICATE_KEY_QUALITY_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Ce contrôle s'applique aux certificats CA ACTIVE ou PENDING_TRANSFER.

Les codes de motif sont renvoyés lorsque ce contrôle trouve un certificat non conforme :

- CERTIFICATE_KEY_VULNERABILITY_CVE-2017-15361

- CERTIFICATE_KEY_VULNERABILITY_CVE-2008-0166

Pourquoi est-ce important ?

Les appareils nouvellement ajoutés signés à l'aide de ce certificat CA peuvent constituer une menace pour la sécurité.

Comment réparer

1. Utilisez UpdateCACertificate pour marquer le certificat CA comme INACTIVE (INACTIF) dans AWS IoT . Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :
   - Appliquer l'action d'atténuation `UPDATE_CA_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
2. Vérifiez l'activité d'enregistrement de certificat d'appareil pendant la période après laquelle le certificat de CA a été révoqué et envisagez de révoquer les certificats d'appareil qui ont pu être émis pendant cette période. (Utilisez ListCertificatesByCA pour répertorier les certificats d'appareil signés par le certificat CA et UpdateCertificate pour révoquer un certificat d'appareil.)

Rôle Cognito non authentifié trop permissif

Une stratégie attachée à un rôle de groupe d'identités Amazon Cognito non authentifiées est considérée comme trop permissive, car elle accorde l'autorisation d'exécuter les tâches suivantes :AWS IoTActions :

- Gérer ou modifier des objets.
- Lire les données administratives d'objet.
- Gérer les données ou ressources liées à d'autres éléments que les objets.

Ou, car elle accorde l'autorisation d'effectuer les actions AWS IoT suivantes sur une large gamme d'appareils :

- Utiliser MQTT pour la connexion, la publication, l'abonnement aux rubriques réservées (y compris les données de shadow ou d'exécution des tâches).
- Utiliser les commandes d'API pour lire ou modifier les données shadow ou d'exécution des tâches.

En général, les appareils qui se connectent à l'aide d'un rôle de groupe d'identités Amazon Cognito non authentifiées ne doit disposer que d'une autorisation limitée pour publier les rubriques MQTT spécifiques à l'objet (et s'y abonner) ou utiliser les commandes d'API pour lire et modifier les données spécifiques à l'objet et liées aux données shadow ou d'exécution des tâches.

Cette vérification s'affiche sous la forme`UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Pour ce contrôle,AWS IoT Device Defendervérifie tous les groupes d'identités Amazon Cognito qui ont été utilisés pour se connecter à l'instanceAWS IoTcourtier de messages au cours des 31 jours précédant

l'exécution de l'audit. Tous les groupes d'identités Amazon Cognito à partir desquels une identité Amazon Cognito authentifiée ou non s'est connectée sont inclus dans la vérification.

Voici les codes de motif renvoyés lorsque ce contrôle trouve un groupe d'identités Amazon Cognito non authentifiées non conforme :

- ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Pourquoi est-ce important ?

Comme les identités non authentifiées ne sont jamais authentifiées par l'utilisateur, elles présentent un risque bien plus élevé que les identités Amazon Cognito authentifiées. Si une identité non authentifiée est compromise, elle peut utiliser les actions administratives pour modifier des paramètres du compte, supprimer des ressources ou accéder à des données sensibles. Ou, avec un large accès aux paramètres de l'appareil, elle peut accéder ou modifier des shadows et des tâches pour tous les appareils de votre compte. Un utilisateur invité peut utiliser les autorisations nécessaires pour compromettre l'ensemble de votre parc ou lancer une attaque DDOS à l'aide de messages.

Comment réparer

Une stratégie attachée à un rôle de groupe d'identités Amazon Cognito non authentifié doit accorder uniquement les autorisations requises pour qu'un appareil fasse son travail. Nous vous recommandons la procédure suivante :

1. Créez un nouveau rôle conforme.
2. Créez un groupe d'identités Amazon Cognito et attachez-lui le rôle conforme.
3. Vérifiez que vos identités peuvent accéder à AWS IoT à l'aide du nouveau groupe.
4. Une fois le contrôle terminé, attachez le rôle conforme au groupe d'identités Amazon Cognito qui a été signalé comme non conforme.

Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation pour mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Gérer ou modifier des objets

Les actions d'API AWS IoT suivantes sont utilisées pour gérer ou modifier des objets. L'autorisation d'exécuter ces actions ne doit pas être accordée aux appareils se connectant via un groupe d'identités Amazon Cognito non authentifiées.

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`

- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

Tout rôle qui accorde l'autorisation d'effectuer ces actions sur une seule ressource est considéré comme non conforme.

Lire les données administratives d'objet.

Les actions d'API AWS IoT suivantes sont utilisées pour lire ou modifier les données d'objet. Les appareils qui se connectent par le biais d'un groupe d'identités Amazon Cognito non authentifiées ne doivent pas être autorisés à effectuer ces actions.

- `DescribeThing`
- `ListJobExecutionsForThing`
- `ListThingGroupsForThing`
- `ListThingPrincipals`

Example

- non conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

Cela permet à l'appareil d'effectuer les actions spécifiées, même si l'action est accordée pour un seul objet.

Gérer les non-objets

Les appareils qui se connectent par le biais d'un groupe d'identités Amazon Cognito non authentifiées ne doivent pas être autorisés à effectuer desAWS IoTActions API autres que celles abordées dans ces sections. Vous pouvez gérer votre compte avec une application qui se connecte via un groupe d'identités Amazon Cognito non authentifiées en créant un autre groupe d'identités non utilisé par les appareils.

S'abonner/publier sur les rubriques MQTT

Les messages MQTT sont envoyés via l'agent de messages AWS IoT et sont utilisés par les appareils afin d'effectuer diverses actions, dont l'accès à l'état du shadow et sa modification, ainsi que l'état de l'exécution des tâches. Une stratégie qui accorde l'autorisation à un appareil de se connecter à des messages MQTT, de les publier ou de s'y abonner, doit limiter ces actions à des ressources spécifiques comme suit :

Connexion

- non conforme :

```
arn:aws:iot:region:account-id:client/*
```

Le caractère générique « * » permet à n'importe quel appareil de se connecter à AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

Sauf si `iot:Connection.Thing.IsAttached` est défini sur true dans les clés de condition, c'est l'équivalent du caractère générique « * » dans l'exemple précédent.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

La spécification de ressource contient une variable qui correspond au nom de l'appareil utilisé pour la connexion. L'instruction de condition limite encore l'autorisation en vérifiant que le certificat utilisé par le client MQTT correspondent à celui attaché à l'objet avec le nom utilisé.

Publier

- non conforme :

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Cela permet à l'appareil de mettre à jour le shadow de n'importe quel appareil (* = tous les appareils).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Cela permet à l'appareil de lire, mettre à jour ou supprimer le shadow de n'importe quel appareil.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
```

```
  ]
}
```

La spécification de ressource contient un caractère générique, mais il correspond uniquement à une rubrique liée au shadow pour l'appareil dont le nom d'objet est utilisé pour la connexion.

S'abonner

- non conforme :

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Cela permet à l'appareil de s'abonner aux rubriques de shadow ou de tâche réservées pour tous les appareils.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Identique à l'exemple précédent, mais à l'aide du caractère générique #.

```
arn:aws:iot:region:account-id:topic/$aws/things/+/shadow/update
```

Cela permet à l'appareil d'afficher les mises à jour du shadow sur n'importe quel appareil (+ = tous les appareils).

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

La spécification de ressource contient des caractères génériques, mais ils correspondent uniquement à une rubrique liée au shadow ou à une tâche pour l'appareil dont le nom d'objet est utilisé pour la connexion.

Réception

- conforme :

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Cela est acceptable, car le dispositif peut recevoir uniquement des messages à partir de rubriques auxquelles il a l'autorisation de s'abonner.

Lecture/modification des données shadow ou de tâche

Une stratégie qui accorde l'autorisation à un appareil d'exécuter une action d'API pour accéder aux données des shadows d'appareil ou d'exécution des tâches, ou les modifier, doit limiter ces actions à des ressources spécifiques. Voici les actions d'API :

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Example

- non conforme :

```
arn:aws:iot:region:account-id:thing/*
```

  Cela permet à l'appareil d'effectuer l'action spécifiée sur n'importe quel objet.
- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

  Cela permet à l'appareil d'effectuer les actions spécifiées sur deux objets uniquement.

Rôle Cognito authentifié trop permissif

Une stratégie attachée à un rôle de groupe d'identités Amazon Cognito authentifiées est considérée comme trop permissive, car elle accorde l'autorisation d'exécuter les tâches suivantes :AWS IoTActions :

- Gérer ou modifier des objets.
- Gérer les données ou ressources liées à d'autres éléments que les objets.

Ou, car elle accorde l'autorisation d'effectuer les actions AWS IoT suivantes sur une large gamme d'appareils :

- Lire les données administratives d'objet.
- Utiliser MQTT pour la connexion/la publication/l'abonnement aux rubriques réservées (y compris les données shadow ou d'exécution des tâches).
- Utiliser les commandes d'API pour lire ou modifier les données shadow ou d'exécution des tâches.

En général, les appareils qui se connectent à l'aide d'un rôle de groupe d'identités Amazon Cognito authentifiées ne doit disposer que d'une autorisation limitée pour lire les données administratives spécifiques à l'objet, publier les rubriques MQTT spécifiques à l'objet (et s'y abonner) ou utiliser les commandes d'API pour lire et modifier les données spécifiques à l'objet et liées aux shadow ou tâche Exécution de données.

Cette vérification s'affiche sous la forme`AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Pour ce contrôle,AWS IoT Device Defendervérifie tous les groupes d'identités Amazon Cognito qui ont été utilisés pour se connecter à l'instanceAWS IoTcourtier de messages au cours des 31 jours précédant l'exécution de l'audit. Tous les groupes d'identités Amazon Cognito à partir desquels une identité Amazon Cognito authentifiée ou non s'est connectée sont inclus dans la vérification.

Voici les codes de motif renvoyés lorsque ce contrôle trouve un groupe d'identités Amazon Cognito authentifiées non conforme :

- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS

Pourquoi est-ce important ?

Si une identité authentifiée est compromise, elle peut utiliser les actions administratives pour modifier des paramètres du compte, supprimer des ressources ou accéder à des données sensibles.

Comment réparer

Une stratégie attachée à un rôle de groupe d'identités Amazon Cognito authentifiées doit accorder uniquement les autorisations requises pour qu'un appareil fasse son travail. Nous vous recommandons la procédure suivante :

1. Créez un nouveau rôle conforme.
2. Créez un groupe d'identités Amazon Cognito et attachez-lui le rôle conforme.
3. Vérifiez que vos identités peuvent accéder à AWS IoT à l'aide du nouveau groupe.
4. Une fois le contrôle terminé, attachez le rôle au groupe d'identités Amazon Cognito qui a été signalé comme non conforme.

Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation pour mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Gérer ou modifier des objets

Procédez comme suit :AWS IoTLes actions d'API sont utilisées pour gérer ou modifier les objets afin que l'autorisation de les exécuter ne soit pas accordée aux appareils se connectant via un groupe d'identités Amazon Cognito authentifiées :

- `AddThingToThingGroup`
- `AttachThingPrincipal`
- `CreateThing`
- `DeleteThing`
- `DetachThingPrincipal`
- `ListThings`
- `ListThingsInThingGroup`
- `RegisterThing`
- `RemoveThingFromThingGroup`
- `UpdateThing`
- `UpdateThingGroupsForThing`

Tout rôle qui accorde l'autorisation d'effectuer ces actions sur une seule ressource est considéré comme non conforme.

Gérer les non-objets

Les appareils qui se connectent par le biais d'un groupe d'identités Amazon Cognito authentifiées ne doivent pas être autorisés à effectuer desAWS IoTActions API autres que celles abordées dans ces sections. Afin de gérer votre compte avec une application qui se connecte via un groupe d'identités Amazon Cognito authentifiées, créez un autre groupe d'identités non utilisé par les appareils.

Lire les données administratives d'objet.

Procédez comme suit :AWS IoTLes actions d'API sont utilisées pour lire les données des objets afin que les appareils se connectant via un groupe d'identités Amazon Cognito authentifiées reçoive l'autorisation de ne les exécuter que sur un ensemble limité d'objets :

- `DescribeThing`
- `ListJobExecutionsForThing`
- `ListThingGroupsForThing`
- `ListThingPrincipals`

- non conforme :

```
arn:aws:iot:region:account-id:thing/*
```

  Cela permet à l'appareil d'effectuer l'action spécifiée sur n'importe quel objet.
- conforme :

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

```
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
      ]
    }
  ]
}
```

Cela permet à l'appareil d'effectuer les actions spécifiées sur un seul objet.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing*"
      ]
    }
  ]
}
```

Cela est conforme, car même si la ressource est spécifiée à l'aide d'un caractère générique (« * »), elle est précédée d'une chaîne spécifique qui limite l'ensemble des éléments accessibles à ceux dont les noms ont le préfixe donné.

- non conforme :

```
arn:aws:iot:region:account-id:thing/*
```

Cela permet à l'appareil d'effectuer l'action spécifiée sur n'importe quel objet.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing"
```

```
      ]
    }
  ]
}
```

Cela permet à l'appareil d'effectuer les actions spécifiées sur un seul objet.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DescribeThing",
          "iot:ListJobExecutionsForThing",
          "iot:ListThingGroupsForThing",
          "iot:ListThingPrincipals"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing*"
      ]
    }
  ]
}
```

Cela est conforme, car même si la ressource est spécifiée à l'aide d'un caractère générique (« * »), elle est précédée d'une chaîne spécifique qui limite l'ensemble des éléments accessibles à ceux dont les noms ont le préfixe donné.

S'abonner/publier sur les rubriques MQTT

Les messages MQTT sont envoyés via l'agent de messages AWS IoT et sont utilisés par les appareils pour effectuer diverses actions, dont l'accès à l'état du shadow et d'exécution des tâches, ainsi que sa modification. Une stratégie qui accorde l'autorisation à un appareil de se connecter à des messages MQTT, de les publier ou de s'y abonner, doit limiter ces actions à des ressources spécifiques comme suit :

Connexion

- non conforme :

```
arn:aws:iot:region:account-id:client/*
```

Le caractère générique « * » permet à n'importe quel appareil de se connecter à AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

Sauf si `iot:Connection.Thing.IsAttached` est défini sur true dans les clés de condition, c'est l'équivalent du caractère générique « * » dans l'exemple précédent.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
```

```
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
}
```

La spécification de ressource contient une variable qui correspond au nom de l'appareil utilisé pour se connecter et la déclaration de la condition limite plus avant l'autorisation en vérifiant que le certificat utilisé par le client MQTT correspond à celui attaché à l'objet avec le nom utilisé.

Publier

- non conforme :

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Cela permet à l'appareil de mettre à jour le shadow de n'importe quel appareil (* = tous les appareils).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Cela permet à l'appareil de lire/mettre à jour/supprimer le shadow de n'importe quel appareil.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

La spécification de ressource contient un caractère générique, mais il correspond uniquement à une rubrique liée au shadow pour l'appareil dont le nom d'objet est utilisé pour la connexion.

S'abonner

- non conforme :

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Cela permet à l'appareil de s'abonner aux rubriques de shadow ou de tâche réservées pour tous les appareils.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/#
```

Identique à l'exemple précédent, mais à l'aide du caractère générique #.

```
arn:aws:iot:region:account-id:topic/$aws/things/+/shadow/update
```

Cela permet à l'appareil d'afficher les mises à jour du shadow sur n'importe quel appareil (+ = tous les appareils).

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

La spécification de ressource contient des caractères génériques, mais ils correspondent uniquement à une rubrique liée au shadow ou à une tâche pour l'appareil dont le nom d'objet est utilisé pour la connexion.

Réception

- conforme :

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Conforme, car l'appareil peut uniquement recevoir des messages à partir de rubriques auxquelles il est autorisé à s'abonner.

Lire ou modifier les données shadow ou de tâche

Une stratégie qui accorde l'autorisation à un appareil d'exécuter une action d'API pour accéder aux données des shadows d'appareil ou d'exécution des tâches, ou les modifier, doit limiter ces actions à des ressources spécifiques. Voici les actions d'API :

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Exemples

- non conforme :

```
arn:aws:iot:region:account-id:thing/*
```

Cela permet à l'appareil d'effectuer l'action spécifiée sur n'importe quel objet.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

Cela permet à l'appareil d'effectuer les actions spécifiées sur deux objets uniquement.

AWS IoTPolitiques excessivement permissives

Une stratégie AWS IoT accorde des autorisations qui sont trop larges ou illimitées. Elle accorde l'autorisation d'envoyer ou de recevoir des messages MQTT pour une large gamme d'appareils, ou accorde l'autorisation d'accéder aux données shadow et d'exécution des tâches (ou de les modifier) pour un vaste éventail d'appareils.

En général, une stratégie pour un appareil doit accorder l'accès à des ressources associées pratiquement à ce seul appareil. Avec certaines exceptions, l'utilisation d'un caractère générique (par exemple, « * ») pour spécifier des ressources dans une telle stratégie est considérée comme trop large ou illimitée.

Cette vérification s'affiche sous la forme`IOT_POLICY_OVERLY_PERMISSIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Le code de motif suivant est renvoyé lorsque ce contrôle trouve une stratégie AWS IoT non conforme :

- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Pourquoi est-ce important ?

Un certificat, une identité Amazon Cognito ou un groupe d'objets avec une stratégie trop permissive peuvent influer, s'ils sont compromis, sur la sécurité de l'ensemble de votre compte. Un pirate informatique pourrait utiliser un tel accès étendu pour lire ou modifier les shadows, les tâches ou les exécutions de tâche de tous vos appareils. Ou un pirate peut utiliser un certificat mis en danger pour connecter des appareils malveillantes ou lancer une attaque DDOS sur votre réseau.

Comment réparer

Suivez ces étapes pour corriger les stratégies non conformes attachées à des objets, des groupes d'objets ou d'autres entités :

1. Utilisez CreatePolicyVersion pour créer une nouvelle version conforme de la stratégie. Définissez l'indicateur `setAsDefault` sur true. (Cela rend cette nouvelle version opérationnelle pour toutes les entités qui utilisent la stratégie.)
2. Utilisez ListTargetsForPolicy pour obtenir la liste des cibles (certificats, groupes d'objets) auxquelles la stratégie est attachée et déterminer les appareils qui sont inclus dans les groupes ou qui utilisent les certificats pour se connecter.
3. Vérifiez que tous les appareils associés sont en mesure de se connecter à AWS IoT. Si un appareil n'est pas en mesure de se connecter, utilisez SetPolicyVersion pour restaurer la stratégie par défaut à la version précédente, révisez la stratégie et faites une nouvelle tentative.

Vous pouvez utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer l'action d'atténuation `REPLACE_DEFAULT_POLICY_VERSION` sur vos résultats d'audit pour effectuer ce changement.
- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Utilisez les variables de stratégie AWS IoT (p. 273) pour faire référence de manière dynamique aux ressources AWS IoT de vos stratégies.

Autorisations MQTT

Les messages MQTT sont envoyés via l'agent de messages AWS IoT et sont utilisés par les appareils afin d'effectuer diverses actions, dont l'accès à l'état du shadow et sa modification, ainsi que l'état de l'exécution des tâches. Une stratégie qui accorde l'autorisation à un appareil de se connecter à des messages MQTT, de les publier ou de s'y abonner, doit limiter ces actions à des ressources spécifiques comme suit :

Connexion

- non conforme :

```
arn:aws:iot:region:account-id:client/*
```

Le caractère générique « * » permet à n'importe quel appareil de se connecter à AWS IoT.

```
arn:aws:iot:region:account-id:client/${iot:ClientId}
```

Sauf si `iot:Connection.Thing.IsAttached` est défini sur true dans les clés de condition, c'est l'équivalent du caractère générique « * » comme dans l'exemple précédent.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Connect" ],
      "Resource": [
        "arn:aws:iot:region:account-id:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": { "iot:Connection.Thing.IsAttached": "true" }
      }
    }
  ]
```

```
}
```

La spécification de ressource contient une variable qui correspond au nom de l'appareil utilisé pour la connexion. L'instruction de condition limite encore l'autorisation en vérifiant que le certificat utilisé par le client MQTT correspondent à celui attaché à l'objet avec le nom utilisé.

Publier

- non conforme :

```
arn:aws:iot:region:account-id:topic/$aws/things/*/shadow/update
```

Cela permet à l'appareil de mettre à jour le shadow de n'importe quel appareil (* = tous les appareils).

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Cela permet à l'appareil de lire, mettre à jour ou supprimer le shadow de n'importe quel appareil.

- conforme :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Publish" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topic/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
      ],
    }
  ]
}
```

La spécification de ressource contient un caractère générique, mais il correspond uniquement à une rubrique liée au shadow pour l'appareil dont le nom d'objet est utilisé pour la connexion.

S'abonner

- non conforme :

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Cela permet à l'appareil de s'abonner aux rubriques de shadow ou de tâche réservées pour tous les appareils.

```
arn:aws:iot:region:account-id:topicfilter/$aws/things/*
```

Identique à l'exemple précédent, mais à l'aide du caractère générique #.

```
arn:aws:iot:region:account-id:topic/$aws/things/+/shadow/update
```

Cela permet à l'appareil d'afficher les mises à jour du shadow sur n'importe quel appareil (+ = tous les appareils).

- conforme :

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [ "iot:Subscribe" ],
      "Resource": [
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/shadow/*"
        "arn:aws:iot:region:account-id:topicfilter/$aws/things/
${iot:Connection.Thing.ThingName}/jobs/*"
      ],
    }
  ]
}
```

La spécification de ressource contient des caractères génériques, mais ils correspondent uniquement à une rubrique liée au shadow ou à une tâche pour l'appareil dont le nom d'objet est utilisé pour la connexion.

Réception

- conforme :

```
arn:aws:iot:region:account-id:topic/$aws/things/*
```

Conforme, car l'appareil peut uniquement recevoir des messages à partir de rubriques auxquelles il est autorisé à s'abonner.

Autorisations de tâche et de shadow

Une stratégie qui accorde l'autorisation à un appareil d'exécuter une action d'API pour accéder aux données des shadows d'appareil ou d'exécution des tâches, ou les modifier, doit limiter ces actions à des ressources spécifiques. Voici les actions d'API :

- `DeleteThingShadow`
- `GetThingShadow`
- `UpdateThingShadow`
- `DescribeJobExecution`
- `GetPendingJobExecutions`
- `StartNextPendingJobExecution`
- `UpdateJobExecution`

Exemples

- non conforme :

```
arn:aws:iot:region:account-id:thing/*
```

Cela permet à l'appareil d'effectuer l'action spécifiée sur n'importe quel objet.

- conforme :

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

```
      "Action": [
          "iot:DeleteThingShadow",
          "iot:GetThingShadow",
          "iot:UpdateThingShadow",
          "iot:DescribeJobExecution",
          "iot:GetPendingJobExecutions",
          "iot:StartNextPendingJobExecution",
          "iot:UpdateJobExecution"
      ],
      "Resource": [
        "arn:aws:iot:region:account-id:/thing/MyThing1",
        "arn:aws:iot:region:account-id:/thing/MyThing2"
      ]
    }
  ]
}
```

Cela permet à l'appareil d'effectuer les actions spécifiées sur deux objets uniquement.

Alias de rôle trop permissif

AWS IoTfournit un mécanisme permettant aux périphériques connectés de s'authentifier auprès deAWS IoTà l'aide de certificats X.509, puis obtenir de courte duréeAWSInformations d'identification depuis un rôle IAM associé à uneAWS IoTAlias du rôle. Les autorisations pour ces informations d'identification doivent être limitées à l'aide de stratégies d'accès avec des variables de contexte d'authentification. Si vos stratégies ne sont pas configurées correctement, vous risquez de vous exposer à une attaque par escalade de privilèges. Ce contrôle d'audit garantit que les informations d'identification temporaires fournies par les alias de rôle AWS IoT ne sont pas trop permissives.

Ce contrôle est déclenché si l'une des conditions suivantes est identifiée :

- La stratégie fournit des autorisations administratives à tous les services utilisés au cours de l'année écoulée par cet alias de rôle (par exemple, « iot:* », « dynamodb:* », « iam:* », etc.).
- La stratégie fournit un accès étendu aux actions de métadonnées d'objets, un accès aux actions AWS IoT restreintes ou un accès étendu aux actions de plan de données AWS IoT.
- La stratégie donne accès à des services d'audit de sécurité tels que « iam », « cloudtrail », « guardduty », « inspecteur » ou « trustedadvisor ».

Cette vérification s'affiche sous la forme`IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Critique

Details

Les codes de motif suivants sont renvoyés lorsque ce contrôle trouve une stratégie IoT non conforme :

- ALLOWS_BROAD_ACCESS_TO_USED_SERVICES
- ALLOWS_ACCESS_TO_SECURITY_AUDITING_SERVICES
- ALLOWS_BROAD_ACCESS_TO_IOT_THING_ADMIN_READ_ACTIONS
- ALLOWS_ACCESS_TO_IOT_NON_THING_ADMIN_ACTIONS
- ALLOWS_ACCESS_TO_IOT_THING_ADMIN_WRITE_ACTIONS
- ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS

Pourquoi est-ce important ?

En limitant les autorisations à celles qui sont nécessaires pour qu'un appareil puisse fonctionner normalement, vous réduisez les risques qui pèsent sur votre compte si un appareil est compromis.

Comment réparer

Suivez ces étapes pour corriger les stratégies non conformes attachées à des objets, des groupes d'objets ou d'autres entités :

1. Suivez les étapes de la section Autorisation des appels directs àAWSservices (p. 307) pour appliquer une stratégie plus restrictive à votre alias de rôle.

Vous pouvez utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une action personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

L'alias de rôle permet d'accéder aux services inutilisés

AWS IoTfournit un mécanisme permettant aux périphériques connectés de s'authentifier auprès deAWS IoTà l'aide de certificats X.509, puis obtenir de courte duréeAWSInformations d'identification depuis un rôle IAM associé à uneAWS IoTAlias du rôle. Les autorisations pour ces informations d'identification doivent être limitées à l'aide de stratégies d'accès avec des variables de contexte d'authentification. Si vos stratégies ne sont pas configurées correctement, vous risquez de vous exposer à une attaque par escalade de privilèges. Ce contrôle d'audit garantit que les informations d'identification temporaires fournies par les alias de rôle AWS IoT ne sont pas trop permissives.

Ce contrôle est déclenché si l'alias de rôle a accès à des services qui n'ont pas été utilisés pour l'appareil AWS IoT au cours de l'année écoulée. Par exemple, les rapports d'audit indiquent si un rôle IAM est lié à l'alias de rôle a uniquement utiliséAWS IoTau cours de la dernière année, mais la politique attachée au rôle accorde également la permission de`"iam:getRole"`and`"dynamodb:PutItem"`.

Cette vérification s'affiche sous la forme`IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Medium

Details

Les codes de motif suivants sont renvoyés lorsque ce contrôle trouve une stratégie AWS IoT non conforme :

- ALLOWS_ACCESS_TO_UNUSED_SERVICES

Pourquoi est-ce important ?

En limitant les autorisations aux services qui sont nécessaires pour qu'un appareil puisse fonctionner normalement, vous réduisez les risques qui pèsent sur votre compte si un appareil est compromis.

Comment réparer

Suivez ces étapes pour corriger les stratégies non conformes attachées à des objets, des groupes d'objets ou d'autres entités :

1. Suivez les étapes de la section Autorisation des appels directs àAWSservices (p. 307) pour appliquer une stratégie plus restrictive à votre alias de rôle.

Vous pouvez utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une action personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Certificat CA expirant

Un certificat CA expire sous 30 jours ou a expiré.

Cette vérification s'affiche sous la forme`CA_CERTIFICATE_EXPIRING_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Medium

Details

Ce contrôle s'applique aux certificats CA ACTIVE ou PENDING_TRANSFER.

Voici les codes de motif renvoyés lorsque ce contrôle trouve un certificat CA non conforme :

- CERTIFICATE_APPROACHING_EXPIRATION
- CERTIFICATE_PAST_EXPIRATION

Pourquoi est-ce important ?

Un certificat CA expiré ne doit plus être utilisé pour signer de nouveaux certificats d'appareil.

Comment réparer

Consultez vos bonnes pratiques de sécurité pour savoir comment procéder. Il se peut que vous souhaitiez :

1. Enregistrer un nouveau certificat de CA auprès d'AWS IoT.
2. Vérifier que vous pouvez signer les certificats d'appareil à l'aide du nouveau certificat de CA.
3. Utilisez UpdateCertificate pour marquer l'ancien certificat comme INACTIVE (INACTIF) dans AWS IoT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les opérations suivantes :
   - Appliquer l'action d'atténuation `UPDATE_CA_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).

ID de client MQTT en conflit

Plusieurs appareils se connectent en utilisant le même ID client.

Cette vérification s'affiche sous la forme`CONFLICTING_CLIENT_IDS_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Élevé

Details

Plusieurs connexions ont été établies avec le même ID client, ce qui a entraîné la déconnexion d'un appareil déjà connecté. La spécification MQTT autorise une seule connexion active par ID client. Par conséquent, si un autre appareil se connecte avec le même ID client, l'appareil précédent est déconnecté.

Lorsqu'il est effectué dans le cadre d'une demande d'audit, ce contrôle examine la façon dont les ID client ont été utilisés pour se connecter au cours des 31 jours avant le début de l'audit. Pour les audits planifiés, ce contrôle examine les données entre la dernière fois où le contrôle a été exécuté et le moment où cette instance de l'audit a démarré. Si vous avez pris des mesures pour atténuer cette condition pendant la période contrôlée, notez à quel moment les connexions/déconnexions ont été effectuées pour déterminer si le problème persiste.

Les codes de motif sont renvoyés lorsque ce contrôle trouve une non-conformité :

- DUPLICATE_CLIENT_ID_ACROSS_CONNECTIONS

Les résultats renvoyés par ce contrôle incluent également l'ID client utilisé pour se connecter, les ID principaux et les heures de déconnexion. Les résultats les plus récents sont répertoriés en premier.

Pourquoi est-ce important ?

Les appareils dont les ID sont en conflit sont contraints de se reconnecter en permanence, ce qui peut entraîner la perte de messages ou faire qu'un appareil ne peut pas se connecter.

Cela peut indiquer qu'un appareil ou les informations d'identification d'un appareil ont été divulgués, et peut faire partie d'une attaque DDoS. Il est également possible que les appareils soient mal configurés dans le compte ou qu'un appareil ait une mauvaise connexion et soit forcé de se reconnecter plusieurs fois par minute.

Comment réparer

Enregistrez chaque appareil en tant qu'objet unique dans AWS IoT et utilisez le nom d'objet comme ID client pour la connexion. Ou utilisez un UUID comme ID client lors de la connexion de l'appareil via MQTT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Certificat de l'appareil expirant

Un certificat d'appareil expire sous 30 jours ou a expiré.

Cette vérification s'affiche sous la forme`DEVICE_CERTIFICATE_EXPIRING_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Medium

Details

Ce contrôle s'applique aux certificats d'appareil qui sont ACTIVE ou PENDING_TRANSFER.

Les codes de motif suivants sont renvoyés lorsque ce contrôle trouve un certificat d'appareil non conforme :

- CERTIFICATE_APPROACHING_EXPIRATION

- CERTIFICATE_PAST_EXPIRATION

Pourquoi est-ce important ?

Un certificat d'appareil ne doit pas être utilisé après son expiration.

Comment réparer

Consultez vos bonnes pratiques de sécurité pour savoir comment procéder. Il se peut que vous souhaitiez :

1. Allouer un nouveau certificat et l'attacher à l'appareil.
2. Vérifier que le nouveau certificat est valide et que l'appareil peut l'utiliser pour se connecter.
3. Utilisez UpdateCertificate pour marquer l'ancien certificat comme étant INACTIVE dans AWS IoT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :
   - Appliquer l'action d'atténuation `UPDATE_DEVICE_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer l'action d'atténuation `ADD_THINGS_TO_THING_GROUP` pour ajouter le dispositif à un groupe où vous pouvez prendre des mesures à son égard.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
4. Détacher l'ancien certificat de l'appareil. (Voir DetachThingPrincipal.)

Certificat de périphérique révoqué toujours actif

Un certificat d'appareil révoqué est toujours actif.

Cette vérification s'affiche sous la forme`REVOKED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Medium

Details

Un certificat d'appareil figure dans la liste de révocation des certificats de sa CA, mais est encore actif dans AWS IoT.

Ce contrôle s'applique aux certificats d'appareil qui sont ACTIVE ou PENDING_TRANSFER.

Les codes de motif sont renvoyés lorsque ce contrôle trouve une non-conformité :

- CERTIFICATE_REVOKED_BY_ISSUER

Pourquoi est-ce important ?

Un certificat d'appareil est généralement révoqué s'il a été compromis. Il est possible qu'il n'ait pas encore été révoqué dans AWS IoT en raison d'une erreur ou d'une omission.

Comment réparer

Vérifiez que le certificat d'appareil n'a pas été compromis. S'il l'a été, suivez les bonnes pratiques en matière de sécurité pour traiter cette situation. Il se peut que vous souhaitiez :

1. Allouer un nouveau certificat pour l'appareil.

2. Vérifier que le nouveau certificat est valide et que l'appareil peut l'utiliser pour se connecter.
3. Utiliser UpdateCertificate pour marquer l'ancien certificat comme REVOKED (RÉVOQUÉ) dans AWS IoT. Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :
   - Appliquer l'action d'atténuation `UPDATE_DEVICE_CERTIFICATE` sur vos résultats d'audit pour effectuer ce changement.
   - Appliquer l'action d'atténuation `ADD_THINGS_TO_THING_GROUP` pour ajouter le dispositif à un groupe où vous pouvez prendre des mesures à son égard.
   - Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

   Pour plus d'informations, consultez Actions d'atténuation (p. 978).
4. Détacher l'ancien certificat de l'appareil. (Voir DetachThingPrincipal.)

Connexion désactivée

AWS IoTLes journaux ne sont pas activés dans Amazon CloudWatch. Vérifie la journalisation V1 et V2.

Cette vérification s'affiche sous la forme`LOGGING_DISABLED_CHECK`dans l'interface de ligne de commande et l'API.

Gravité : Faible

Details

Les codes de motif sont renvoyés lorsque ce contrôle trouve une non-conformité :

- LOGGING_DISABLED

Pourquoi est-ce important ?

AWS IoTdans CloudWatch fournissent une visibilité sur les comportements dansAWS IoT, y compris les échecs d'authentification, et les connexions et déconnexions inattendues qui peuvent indiquer qu'un appareil a été compromis.

Comment réparer

Activer lesAWS IoTdans CloudWatch. Consultez Outils de surveillance (p. 343). Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquer l'action d'atténuation `ENABLE_IOT_LOGGING` sur vos résultats d'audit pour effectuer ce changement.
- Appliquer le`PUBLISH_FINDINGS_TO_SNS`Action d'atténuation si vous souhaitez mettre en œuvre une réponse personnalisée pour répondre au message Amazon SNS.

Pour plus d'informations, consultez Actions d'atténuation (p. 978).

Commandes d'audit

Gestion des paramètres d'audit

Utilisez `UpdateAccountAuditConfiguration` pour configurer les paramètres d'audit de votre compte.. Cette commande vous permet d'activer les contrôles que vous souhaitez disponibles pour les audits, de configurer les notifications facultatives et de configurer les autorisations.

Vérifiez ces paramètres avec `DescribeAccountAuditConfiguration`.

Utilisez `DeleteAccountAuditConfiguration` pour supprimer vos paramètres d'audit. Rétablit toutes les valeurs par défaut et désactive efficacement les audits, car tous les contrôles sont désactivés par défaut.

UpdateAccountAuditConfiguration

Configure ou reconfigure les paramètres d'audit Device Defender pour ce compte. Les paramètres incluent le mode d'envoi des notifications d'audit et les contrôles activés ou désactivés.

Résumé

```
aws iot  update-account-audit-configuration \
    [--role-arn <value>] \
    [--audit-notification-target-configurations <value>] \
    [--audit-check-configurations <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

Champs **`cli-input-json`**

Nom	Type	Description
roleArn	string longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde à AWS IoT l'autorisation d'accéder aux informations de vos appareils, stratégies, certificats et autres éléments lors d'un audit.
auditNotificationTargetConfigurations	map	Informations sur les cibles auxquelles les notifications d'audit sont envoyées.
targetArn	string	L'ARN de la cible (rubrique SNS) à laquelle des notifications d'audit sont envoyées.
roleArn	string longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde l'autorisation d'envoyer des notifications à la cible.

Nom	Type	Description
enabled	boolean	La valeur est true si les notifications vers la cible sont activées.
auditCheckConfigurations	map	Spécifie les vérifications d'audit activées et désactivées pour ce compte. Utilisez `DescribeAccountAuditConfiguration` pour afficher la liste de tous les contrôles, y compris ceux qui sont actuellement activés. Certaines collectes de données peuvent démarrer immédiatement lorsque certains contrôles sont activés. Si un contrôle est désactivé, toutes les données collectées jusqu'à présent en relation avec lui sont supprimées. Vous ne pouvez pas désactiver un contrôle s'il est utilisé par un audit planifié. Vous devez d'abord supprimer le contrôle de l'audit planifié ou supprimer l'audit planifié lui-même. Dans le premier appel à `UpdateAccountAuditConfiguration`, ce paramètre est obligatoire et doit spécifier au moins un contrôle activé.
enabled	boolean	La valeur est true si cette vérification d'audit est activée pour ce compte.

Sortie

Aucune

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

DescribeAccountAuditConfiguration

Récupère les informations sur les paramètres d'audit Device Defender pour ce compte. Les paramètres incluent le mode d'envoi des notifications d'audit et les contrôles activés ou désactivés.

Résumé

```
aws iot  describe-account-audit-configuration  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
}
```

Sortie

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
roleArn	string longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde à AWS IoT l'autorisation d'accéder aux informations de vos appareils, stratégies, certificats et autres éléments lors d'un audit. Lors du premier appel à `UpdateAccountAuditConfiguration`, ce paramètre est requis.
auditNotificationTargetConfigurations	map	Des informations sur les cibles auxquelles les notifications d'audit sont envoyées pour ce compte.
targetArn	string	L'ARN de la cible (rubrique SNS) à laquelle des notifications d'audit sont envoyées.

Nom	Type	Description
roleArn	string longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde l'autorisation d'envoyer des notifications à la cible.
enabled	boolean	La valeur est true si les notifications vers la cible sont activées.
auditCheckConfigurations	map	Les contrôles d'audit activés et désactivés pour ce compte.
enabled	boolean	La valeur est true si cette vérification d'audit est activée pour ce compte.

Erreurs

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

DeleteAccountAuditConfiguration

Restaure les paramètres par défaut des audits Device Defender pour ce compte. Toutes les données de configuration saisies sont supprimées et tous les contrôles d'audit sont réinitialisés pour être désactivés.

Résumé

```
aws iot  delete-account-audit-configuration \
    [--delete-scheduled-audits | --no-delete-scheduled-audits]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "deleteScheduledAudits": "boolean"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
deleteScheduledAudits	boolean	Si la valeur est true, tous les audits planifiés sont supprimés.

Sortie

Aucune

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

Audits planifiés

Utilisez `CreateScheduledAudit` pour créer un ou plusieurs audits planifiés. Cette commande vous permet de spécifier les contrôles que vous souhaitez exécuter lors d'un audit, ainsi que la fréquence à laquelle ces audits doivent être exécutés.

Assurez le suivi de vos audits planifiés avec `ListScheduledAudits` et `DescribeScheduledAudit`.

Modifiez un audit planifié existant avec `UpdateScheduledAudit` ou supprimez-le avec `DeleteScheduledAudit`.

CreateScheduledAudit

Crée un audit planifié exécuté à un intervalle de temps spécifié.

Résumé

```
aws iot  create-scheduled-audit \
    --frequency <value> \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    --target-check-names <value> \
    [--tags <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "scheduledAuditName": "string"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
frequency	string	À quelle fréquence se déroule l'audit planifiée. Peut être « DAILY », « WEEKLY », « BIWEEKLY » ou « MONTHLY ». L'heure de début réelle de chaque audit est déterminée par le système. enum : TOUS LES JOURS | HEBDOMADAIRE | BIMENSUEL | MENSUEL
dayOfMonth	string Modèle : ^ (01]) |3 [01]) |01]) |01]) |01]) $\|^DERNIER$	Le jour du mois auquel l'audit planifié se déroule. Peut être « 1 » à « 31 » ou « LAST ». Ce champ est obligatoire uniquement si le paramètre `frequency` est défini sur « MONTHLY ». Si les jours « 29 » à « 31 » sont spécifiés et que le mois ne compte pas autant de jours, l'audit se déroule le « LAST » (dernier) jour du mois.
dayOfWeek	string	Le jour de la semaine pendant lequel l'audit planifié se déroule. Peut être « SUN », « MON », « TUE », « WED », « THU », « FRI » ou « SAT ». Ce champ est obligatoire si le paramètre `frequency` est défini sur « WEEKLY » ou « BIWEEKLY ». enum : DIM | LUN | MAR | MER | JEU | VEN | SAM
targetCheckNames	liste membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit planifié. Les contrôles doivent être activés sur votre compte. (Utilisez `DescribeAccountAuditConfiguration` pour afficher la liste de tous les contrôles, y compris ceux activés, ou `UpdateAccountAuditConfiguration` pour sélectionner les contrôles activés.)
tags	liste membre : Tag classe Java : java.util.List	Métadonnées qui peuvent être utilisées pour gérer l'audit planifié.

Nom	Type	Description
Key	string	Clé de la balise.
Valeur	string	Valeur de la balise.
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom que vous souhaitez donner à l'audit planifié. (128 caractères maximum)

Sortie

```
{
  "scheduledAuditArn": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
scheduledAuditArn	string	L'ARN de l'audit planifié.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

`LimitExceededException`

Une limite a été dépassée.

ListScheduledAudits

Répertorie tous vos audits planifiés.

Résumé

```
aws iot  list-scheduled-audits \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "nextToken": "string",
  "maxResults": "integer"
```

```
}
```

Champs **`cli-input-json`**

Nom	Type	Description
nextToken	string	Jeton de l'ensemble de résultats suivant.
maxResults	entier plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer simultanément. La valeur par défaut est 25.

Sortie

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "string",
      "scheduledAuditArn": "string",
      "frequency": "string",
      "dayOfMonth": "string",
      "dayOfWeek": "string"
    }
  ],
  "nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
scheduledAudits	liste membre : ScheduledAuditMetadata classe Java : java.util.List	Liste des audits planifiés.
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié.
scheduledAuditArn	string	L'ARN de l'audit planifié.
frequency	string	À quelle fréquence se déroule l'audit planifiée. enum : TOUS LES JOURS | HEBDOMADAIRE | BIMENSUEL | MENSUEL
dayOfMonth	string Modèle : ^ (01]) |3 [01]) |01]) |01]) |01]) $\|^DERNIER$	Jour du mois où l'audit planifié est exécuté (si l'élément `frequency` est « MONTHLY »). Si les jours « 29 » à « 31 » sont spécifiés et que le mois

Nom	Type	Description
		ne compte pas autant de jours, l'audit se déroule le « LAST » (dernier) jour du mois.
dayOfWeek	string	Jour de la semaine où l'audit planifié est exécuté (si `frequency` est « WEEKLY » OU « BIWEEKLY »). enum : DIM | LUN | MAR | MER | JEU | VEN | SAM
nextToken	string	Jeton qui peut être utilisé pour obtenir l'ensemble de résultats suivant, ou `null` s'il n'y a pas d'autres résultats.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

DescribeScheduledAudit

Obtient des informations sur un audit planifié.

Résumé

```
aws iot  describe-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "scheduledAuditName": "string"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié dont vous souhaitez obtenir les informations.

Sortie

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
frequency	string	À quelle fréquence se déroule l'audit planifiée. « DAILY », « WEEKLY », « BIWEEKLY » ou « MONTHLY ». L'heure de début réelle de chaque audit est déterminée par le système. enum : TOUS LES JOURS | HEBDOMADAIRE | BIMENSUEL | MENSUEL
dayOfMonth	string Modèle : ^ (01]) |3 [01]) |01]) |01]) |01]) $\|^DERNIER$	Le jour du mois auquel l'audit planifié se déroule. Peut être « 1 » à « 31 » ou « LAST ». Si les jours « 29 » à « 31 » sont spécifiés et que le mois ne compte pas autant de jours, l'audit se déroule le « LAST » (dernier) jour du mois.
dayOfWeek	string	Le jour de la semaine pendant lequel l'audit planifié se déroule. « SUN », « MON », « TUE », « WED », « THU », « FRI » ou « SAT ». enum : DIM | LUN | MAR | MER | JEU | VEN | SAM
targetCheckNames	liste membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit planifié. Les contrôles doivent être activés sur votre compte. (Utilisez `DescribeAccountAuditConfiguration` pour afficher la liste de tous les contrôles, y compris ceux activés, ou `UpdateAccountAuditConfiguration` pour sélectionner les contrôles activés.)
scheduledAuditName	string	Le nom de l'audit planifié.

Nom	Type	Description
	longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	
scheduledAuditArn	string	L'ARN de l'audit planifié.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

UpdateScheduledAudit

Met à jour un audit régulier, y compris les contrôles exécutés et la fréquence à laquelle l'audit a lieu.

Résumé

```
aws iot  update-scheduled-audit \
    [--frequency <value>] \
    [--day-of-month <value>] \
    [--day-of-week <value>] \
    [--target-check-names <value>] \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
frequency	string	À quelle fréquence se déroule l'audit planifiée. Peut être « DAILY »,

Nom	Type	Description
		« WEEKLY », « BIWEEKLY » ou « MONTHLY ». L'heure de début réelle de chaque audit est déterminée par le système. enum : TOUS LES JOURS | HEBDOMADAIRE | BIMENSUEL | MENSUEL
dayOfMonth	string Modèle : ^ (01]) |3 [01]) |01]) |01]) |01]) $\|^DERNIER$	Le jour du mois auquel l'audit planifié se déroule. Peut être « 1 » à « 31 » ou « LAST ». Ce champ est obligatoire uniquement si le paramètre `frequency` est défini sur « MONTHLY ». Si les jours « 29 » à « 31 » sont spécifiés et que le mois ne compte pas autant de jours, l'audit se déroule le « LAST » (dernier) jour du mois.
dayOfWeek	string	Le jour de la semaine pendant lequel l'audit planifié se déroule. Peut être « SUN », « MON », « TUE », « WED », « THU », « FRI » ou « SAT ». Ce champ est obligatoire si le paramètre `frequency` est défini sur « WEEKLY » ou « BIWEEKLY ». enum : DIM | LUN | MAR | MER | JEU | VEN | SAM
targetCheckNames	liste membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit planifié. Les contrôles doivent être activés sur votre compte. (Utilisez `DescribeAccountAuditConfiguration` pour afficher la liste de tous les contrôles, y compris ceux activés, ou `UpdateAccountAuditConfiguration` pour sélectionner les contrôles activés.)
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié. (128 caractères maximum)

Sortie

```
{
  "scheduledAuditArn": "string"
```

```
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
scheduledAuditArn	string	L'ARN de l'audit planifié.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

DeleteScheduledAudit

Supprime un audit planifié.

Résumé

```
aws iot  delete-scheduled-audit \
    --scheduled-audit-name <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "scheduledAuditName": "string"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié que vous souhaitez supprimer.

Sortie

Aucune

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

Exécution d’un audit à la demande

Utilisez `StartOnDemandAuditTask` pour spécifier les contrôles que vous souhaitez exécuter et démarrer une exécution d'audit immédiatement.

StartOnDemandAuditTask

Démarre un audit Device Defender à la demande.

Résumé

```
aws iot  start-on-demand-audit-task \
    --target-check-names <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

Champs **`cli-input-json`**

Nom	Type	Description
targetCheckNames	liste membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit. Les contrôles que vous spécifiez doivent être activés pour votre compte ou une exception se produit. Utilisez `DescribeAccountAuditConfiguration` pour afficher la liste de tous les contrôles, y compris ceux activés, ou `UpdateAccountAuditConfiguration` pour sélectionner les contrôles activés.

Sortie

```
{
  "taskId": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	ID de l'audit à la demande que vous avez démarré.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

`LimitExceededException`

Une limite a été dépassée.

Gérez les instances d'audit

Utilisez `DescribeAuditTask` pour obtenir des informations sur une instance d'audit spécifique. Si la fonction est déjà exécutée, les résultats incluent les contrôles en échec ou réussis, ceux n'ayant pas pu être achevés par le système et, si l'audit est toujours en cours, ceux sur lesquels ce dernier travaille toujours.

Utilisez `ListAuditTasks` pour trouver les audits exécutés lors d'un intervalle de temps spécifié.

Utilisez `CancelAuditTask` pour arrêter un audit en cours.

DescribeAuditTask

Obtient des informations sur un audit Device Defender.

Résumé

```
aws iot  describe-audit-task \
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "taskId": "string"
```

```
}
```

Champs **`cli-input-json`**

Nom	Type	Description
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	L'ID de l'audit dont vous souhaitez obtenir les informations.

Sortie

```
{
  "taskStatus": "string",
  "taskType": "string",
  "taskStartTime": "timestamp",
  "taskStatistics": {
    "totalChecks": "integer",
    "inProgressChecks": "integer",
    "waitingForDataCollectionChecks": "integer",
    "compliantChecks": "integer",
    "nonCompliantChecks": "integer",
    "failedChecks": "integer",
    "canceledChecks": "integer"
  },
  "scheduledAuditName": "string",
  "auditDetails": {
    "string": {
      "checkRunStatus": "string",
      "checkCompliant": "boolean",
      "totalResourcesCount": "long",
      "nonCompliantResourcesCount": "long",
      "errorCode": "string",
      "message": "string"
    }
  }
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
taskStatus	string	Le statut de l'audit : « IN_PROGRESS », « COMPLETED », « FAILED » ou « CANCELED ». enum : IN_PROGRESS | TERMINÉ | ÉCHOUÉ | ANNULÉ
taskType	string	Type de vérification : ON_DEMAND_DEMAND_TASK ou SCHEDULED_AUDIT_TASK. enum : ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK

Nom	Type	Description
taskStartTime	timestamp	L'heure de début de l'audit.
taskStatistics	TaskStatistics	Les statistiques de l'audit.
totalChecks	entier	Le nombre de contrôles dans cet audit.
inProgressChecks	entier	Le nombre de contrôles en cours.
waitingForDataCollectionChecks	entier	Le nombre de contrôles en attente de collecte des données.
compliantChecks	entier	Le nombre de contrôles conformes aux ressources.
nonCompliantChecks	entier	Le nombre de contrôles non conformes aux ressources.
failedChecks	entier	Le nombre de contrôles.
canceledChecks	entier	Le nombre de contrôles non exécutés à cause de l'annulation de l'audit.
scheduledAuditName	string longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié (uniquement si ce dernier était planifié).
auditDetails	map	Les informations détaillées sur chaque contrôle effectué au cours de cet audit.
checkRunStatus	string	Le statut de finalisation de ce contrôle : « IN_PROGRESS », « WAITING_FOR_DATA_COLLECTION », « CANCELED », « COMPLETED_COMPLIANT », « COMPLETED_NON_COMPLIANT » ou « FAILED ». enum : IN_PROGRESS | WAITING_FOR_DATA_COLLECTION | ANNULÉ | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT | ÉCHEC
checkCompliant	boolean	La valeur est true si le contrôle est terminé et a trouvé toutes les ressources conformes.
totalResourcesCount	long	Le nombre de ressources sur lesquelles le contrôle a été effectué.

Nom	Type	Description
nonCompliantResourcesCount	long	Le nombre de ressources non conformes trouvées par le contrôle.
errorCode	string	Le code des erreurs rencontrées lors de l'exécution de ce contrôle pendant l'audit. « INSUFFICIENT_PERMISSIONS » ou « AUDIT_CHECK_DISABLED ».
message	string longueur - max. : 2048	Le message associé aux erreurs rencontrées lors de l'exécution de ce contrôle pendant l'audit.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

ListAuditTasks

Répertorie les audits Device Defender qui ont été exécutés au cours d'une période donnée.

Résumé

```
aws iot  list-audit-tasks \
    --start-time <value> \
    --end-time <value> \
    [--task-type <value>] \
    [--task-status <value>] \
    [--next-token <value>] \
    [--max-results <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
```

```
  "nextToken": "string",
  "maxResults": "integer"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
startTime	timestamp	Début de la période. Les informations d'audit sont conservées pendant une durée limitée (180 jours). Une demande d'heure de début antérieure à ce qui est conservé génère une exception `InvalidRequestException`.
endTime	timestamp	Fin de la période.
taskType	string	Filtre pour limiter la sortie du type d'audit spécifié : peut être « ON_DEMAND_AUDIT_TASK » ou « SCHEDULED__AUDIT_TASK ». enum : ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
taskStatus	string	Filtre pour limiter la sortie des audits spécifiée avec le statut d'achèvement : « IN_PROGRESS », « COMPLETED », « FAILED » ou « CANCELED ». enum : IN_PROGRESS | TERMINÉ | ÉCHOUÉ | ANNULÉ
nextToken	string	Jeton de l'ensemble de résultats suivant.
maxResults	entier plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer simultanément. La valeur par défaut est 25.

Sortie

```
{
  "tasks": [
    {
      "taskId": "string",
      "taskStatus": "string",
      "taskType": "string"
    }
  ],
  "nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
tasks	liste membre : AuditTaskMetadata classe Java : java.util.List	Audits exécutés au cours de la période spécifiée.
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	ID de cet audit.
taskStatus	string	Statut de l'audit : « IN_PROGRESS », « COMPLETED », « FAILED » ou « CANCELED ». enum : IN_PROGRESS | TERMINÉ | ÉCHOUÉ | ANNULÉ
taskType	string	Type de l'audit : « ON_DEMAND_AUDIT_TASK » ou « SCHEDULED_AUDIT_TASK ». enum : ON_DEMAND_AUDIT_TASK | SCHEDULED_AUDIT_TASK
nextToken	string	Jeton qui peut être utilisé pour obtenir l'ensemble de résultats suivant, ou `null`, s'il n'y a pas de résultats supplémentaires.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

CancelAuditTask

Annule un audit en cours. L'audit peut être planifié ou à la demande. Si le contrôle n'est pas en cours, une exception `InvalidRequestException`.

Résumé

```
aws iot  cancel-audit-task \
```

```
    --task-id <value>  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "taskId": "string"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	L'ID de l'audit que vous souhaitez annuler. Vous pouvez uniquement annuler un audit « IN_PROGRESS ».

Sortie

Aucune

Erreurs

`ResourceNotFoundException`

La ressource spécifiée n'existe pas.

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

Vérifiez les résultats de l'audit

Utilisez `ListAuditFindings` pour consulter les résultats d'un audit. Vous pouvez filtrer les résultats par contrôle, ressource spécifique ou heure d'audit. Vous pouvez utiliser ces informations pour atténuer les problèmes détectés.

Vous pouvez définir des mesures d'atténuation et les appliquer aux résultats de votre audit. Pour plus d'informations, consultez Actions d'atténuation (p. 978).

ListAuditFindings

Répertorie les conclusions (résultats) d'un audit Device Defender ou des audits effectués pendant une période déterminée. (Les conclusions sont conservées pendant 180 jours.)

Résumé

```
aws iot  list-audit-findings \
```

```
    [--task-id <value>] \
    [--check-name <value>] \
    [--resource-identifier <value>] \
    [--max-results <value>] \
    [--next-token <value>] \
    [--start-time <value>] \
    [--end-time <value>]  \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

Format de `cli-input-json`

```
{
  "taskId": "string",
  "checkName": "string",
  "resourceIdentifier": {
    "deviceCertificateId": "string",
    "caCertificateId": "string",
    "cognitoIdentityPoolId": "string",
    "clientId": "string",
    "policyVersionIdentifier": {
      "policyName": "string",
      "policyVersionId": "string"
    },

    "roleAliasArn": "string",
    "account": "string"
  },
  "maxResults": "integer",
  "nextToken": "string",
  "startTime": "timestamp",
  "endTime": "timestamp"
}
```

Champs **`cli-input-json`**

Nom	Type	Description
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	Filtre pour limiter les résultats à l'audit avec l'ID spécifié. Vous devez spécifier le taskId ou les startTime et endTime, mais pas les deux.
checkName	string	Filtre pour limiter les conclusions au contrôle d'audit spécifié.
resourceIdentifier	ResourceIdentifier	Informations qui identifient les ressources non conformes.
deviceCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.

Nom	Type	Description
cognitoIdentityPoolId	string	ID du groupe d'identités Amazon Cognito.
clientId	string	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.
policyName	string longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	string Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.
roleAliasArn	string	ARN de l'alias de rôle ayant des actions trop permissives. longueur - max. : 2 048. Min. : 1
account	string longueur - max. : 12. Min. : 12 Modèle : [0-9] +	Compte auquel la ressource est associée.
maxResults	entier plage - max. : 250 min. : 1	Nombre maximal de résultats à renvoyer simultanément. La valeur par défaut est 25.
nextToken	string	Jeton de l'ensemble de résultats suivant.
startTime	timestamp	Filtre pour limiter les résultats à ceux obtenus après l'heure spécifiée. Vous devez spécifier le taskId ou les startTime et endTime, mais pas les deux.
endTime	timestamp	Filtre pour limiter les résultats à ceux obtenus avant l'heure spécifiée. Vous devez spécifier le taskId ou les startTime et endTime, mais pas les deux.

Sortie

```
{
  "findings": [
    {
      "taskId": "string",
      "checkName": "string",
      "taskStartTime": "timestamp",
      "findingTime": "timestamp",
      "severity": "string",
```

```
      "nonCompliantResource": {
        "resourceType": "string",
        "resourceIdentifier": {
          "deviceCertificateId": "string",
          "caCertificateId": "string",
          "cognitoIdentityPoolId": "string",
          "clientId": "string",
          "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
          },
          "account": "string"
        },
        "additionalInfo": {
          "string": "string"
        }
      },
      "relatedResources": [
        {
          "resourceType": "string",
          "resourceIdentifier": {
            "deviceCertificateId": "string",
            "caCertificateId": "string",
            "cognitoIdentityPoolId": "string",
            "clientId": "string",

            "iamRoleArn": "string",

            "policyVersionIdentifier": {
              "policyName": "string",
              "policyVersionId": "string"
            },
            "account": "string"
          },

          "roleAliasArn": "string",

          "additionalInfo": {
            "string": "string"
          }
        }
      ],
      "reasonForNonCompliance": "string",
      "reasonForNonComplianceCode": "string"
    }
  ],
  "nextToken": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
findings	liste membre : AuditFinding	Conclusions (résultats) de l'audit.
taskId	string longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	ID de l'audit qui a généré ce résultat.

Nom	Type	Description
checkName	string	Contrôle d'audit qui a généré le résultat.
taskStartTime	timestamp	L'heure de début de l'audit.
findingTime	timestamp	Heure à laquelle le résultat (finding) a été découvert.
severity	string	Gravité du résultat (finding). enum : CRITIQUE | ÉLEVÉ | MOYEN | FAIBLE
nonCompliantResource	NonCompliantResource	Ressource qui a été détectée comme non conforme avec le contrôle d'audit.
type de ressource	string	Type de la ressource non conforme. enum : DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Informations qui identifient les ressources non conformes.
deviceCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.
cognitoIdentityPoolId	string	ID du groupe d'identités Amazon Cognito.
clientId	string	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.
policyName	string longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	string Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.

Nom	Type	Description
compte	string longueur - max. : 12. Min. : 12 Modèle : [0-9] +	Compte auquel la ressource est associée.
additionalInfo	map	Autres informations relatives à la ressource non conforme.
relatedResources	liste membre : RelatedResource	Liste des ressources associées.
type de ressource	string	Le type de ressource. enum : DEVICE_CERTIFICATE | CA_CERTIFICATE | IOT_POLICY | COGNITO_IDENTITY_POOL | CLIENT_ID | ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Informations qui identifient la ressource.
deviceCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	string longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.
cognitoIdentityPoolId	string	ID du groupe d'identités Amazon Cognito.
clientId	string	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.
iamRoleArn	string longueur - max. : 2048. Min. : 20	ARN du rôle IAM ayant des actions trop permissives.
policyName	string longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	string Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.

Nom	Type	Description
roleAliasArn	string longueur - max. : 2 048. Min. : 1	ARN de l'alias de rôle ayant des actions trop permissives.
compte	string longueur - max. : 12. Min. : 12 Modèle : [0-9] +	Compte auquel la ressource est associée.
additionalInfo	map	Autres informations relatives à la ressource.
reasonForNonCompliance	string	Raison pour laquelle la ressource était non conforme.
reasonForNonComplianceCode	string	Code qui indique la raison pour laquelle la ressource était non conforme.
nextToken	string	Jeton qui peut être utilisé pour obtenir l'ensemble de résultats suivant, ou `null`, s'il n'y a pas de résultats supplémentaires.

Erreurs

`InvalidRequestException`

Le contenu de la demande n'était pas valide.

`ThrottlingException`

Le tarif dépasse la limite.

`InternalFailureException`

Une erreur inattendue est survenue.

Suppressions des résultats d'audit

Lorsque vous exécutez un audit, il signale les résultats de toutes les ressources non conformes. Cela signifie que vos rapports d'audit contiennent des conclusions pour les ressources dans lesquelles vous travaillez à l'atténuation des problèmes ainsi que pour les ressources connues pour être non conformes, telles que les périphériques de test ou les périphériques défectueux. L'audit continue de signaler les constatations relatives aux ressources qui restent non conformes lors d'exécutions d'audit successives, ce qui peut ajouter des informations indésirables à vos rapports. Les suppressions de recherche d'audit vous permettent de supprimer ou de filtrer les résultats pendant une période définie jusqu'à ce que la ressource soit fixe, ou indéfiniment pour une ressource associée à un périphérique test ou cassé.

Note

Les mesures d'atténuation ne seront pas disponibles pour les constatations d'audit supprimées. Pour plus d'informations sur les actions d'atténuation, consultezActions d'atténuation (p. 978).

Pour plus d'informations sur les quotas de suppression, consultezAWS IoTPoints de terminaison et quotas Device Defender.

Fonctionnement des suppressions de recherche d'audit

Lorsque vous créez une suppression de recherche d'audit pour une ressource non conforme, vos rapports d'audit et notifications se comportent différemment.

Vos rapports d'audit comprendront une nouvelle section qui répertorie toutes les constatations supprimées associées au rapport. Les constatations supprimées ne seront pas prises en compte lorsque nous évaluons si une vérification d'audit est conforme ou non. Un nombre de ressources supprimé est également renvoyé pour chaque vérification d'audit lorsque vous utilisez la commandedescribe-audit-taskDans l'interface de ligne de commande (CLI).

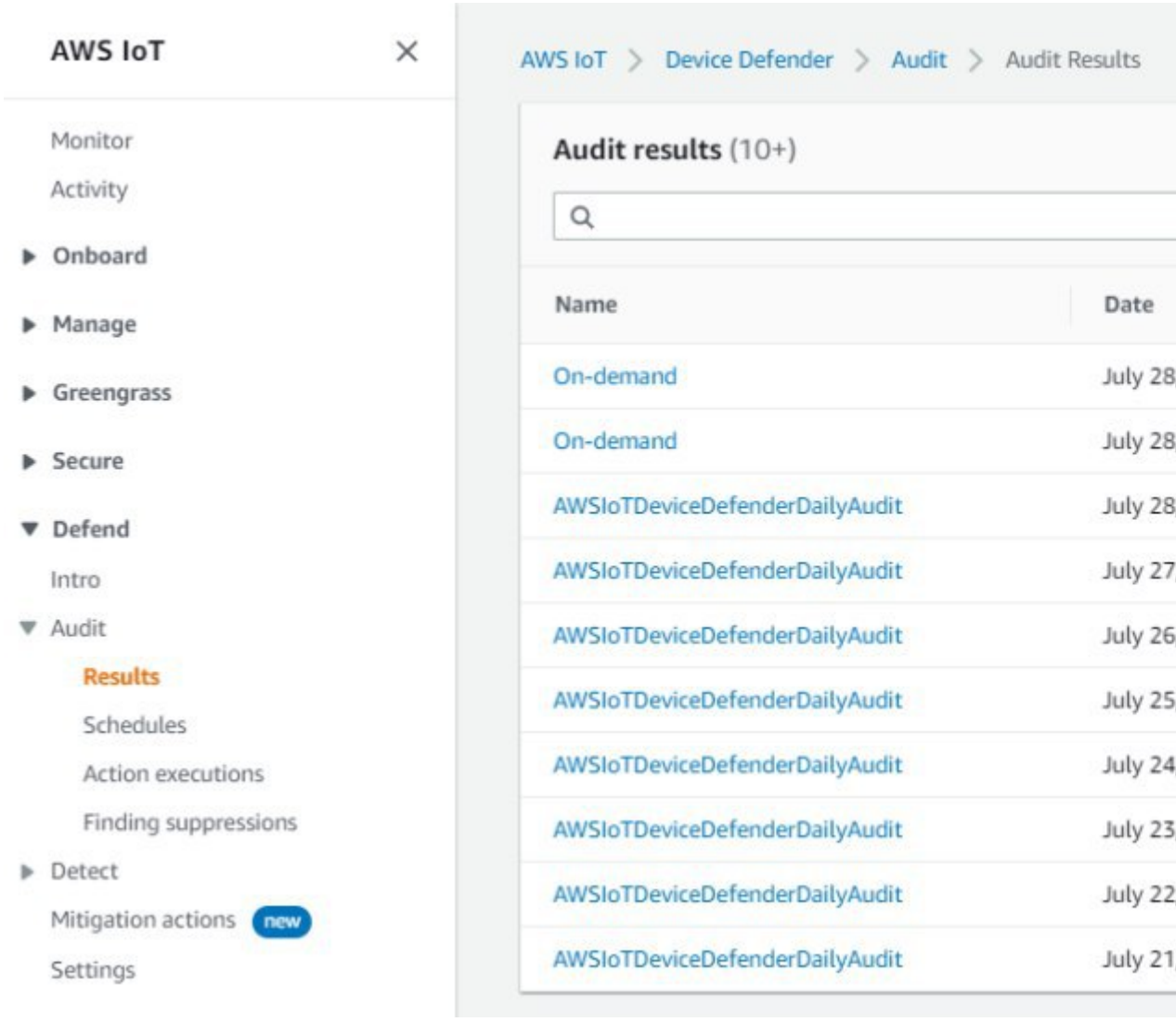
Pour les notifications d'audit, les constatations supprimées ne sont pas prises en compte lorsque nous évaluons si une vérification d'audit est conforme ou non. Un nombre de ressources supprimé est également inclus dans chaque notification de vérificationAWS IoT Device Defenderpublie sur Amazon CloudWatch et Amazon Simple Notification Service (Amazon SNS).

Comment utiliser les suppressions de recherche d'audit dans la console

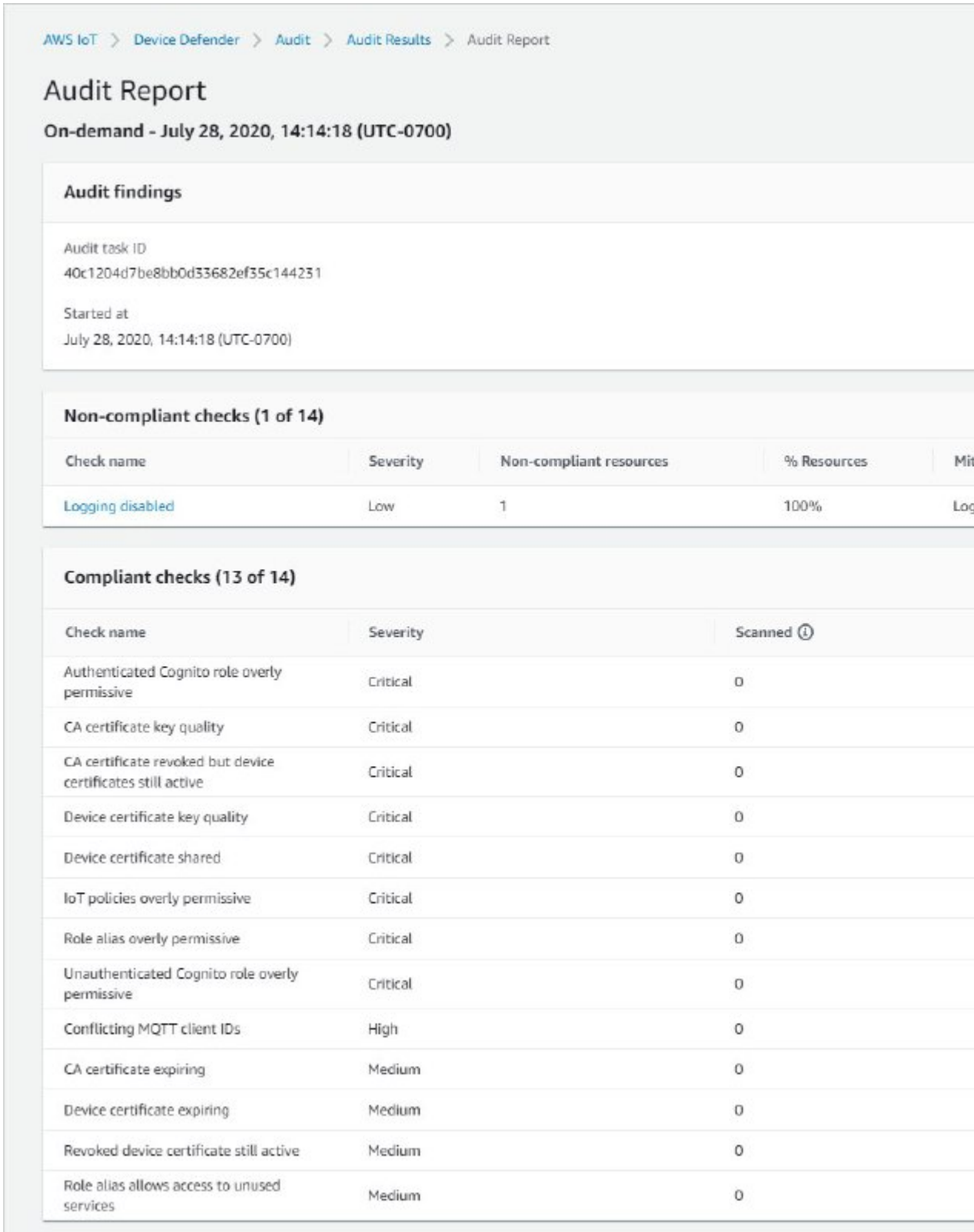
Pour supprimer une constatation d'un rapport d'audit

La procédure suivante vous montre comment créer une suppression de recherche d'audit dans la boîte de dialogueAWS IoTconsole

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Audit,Résultats.
2. Sélectionnez un rapport d'audit que vous souhaitez consulter.



3. DansContrôles non conformes, sousNom de la vérification, choisissez la vérification d'audit qui vous intéresse.

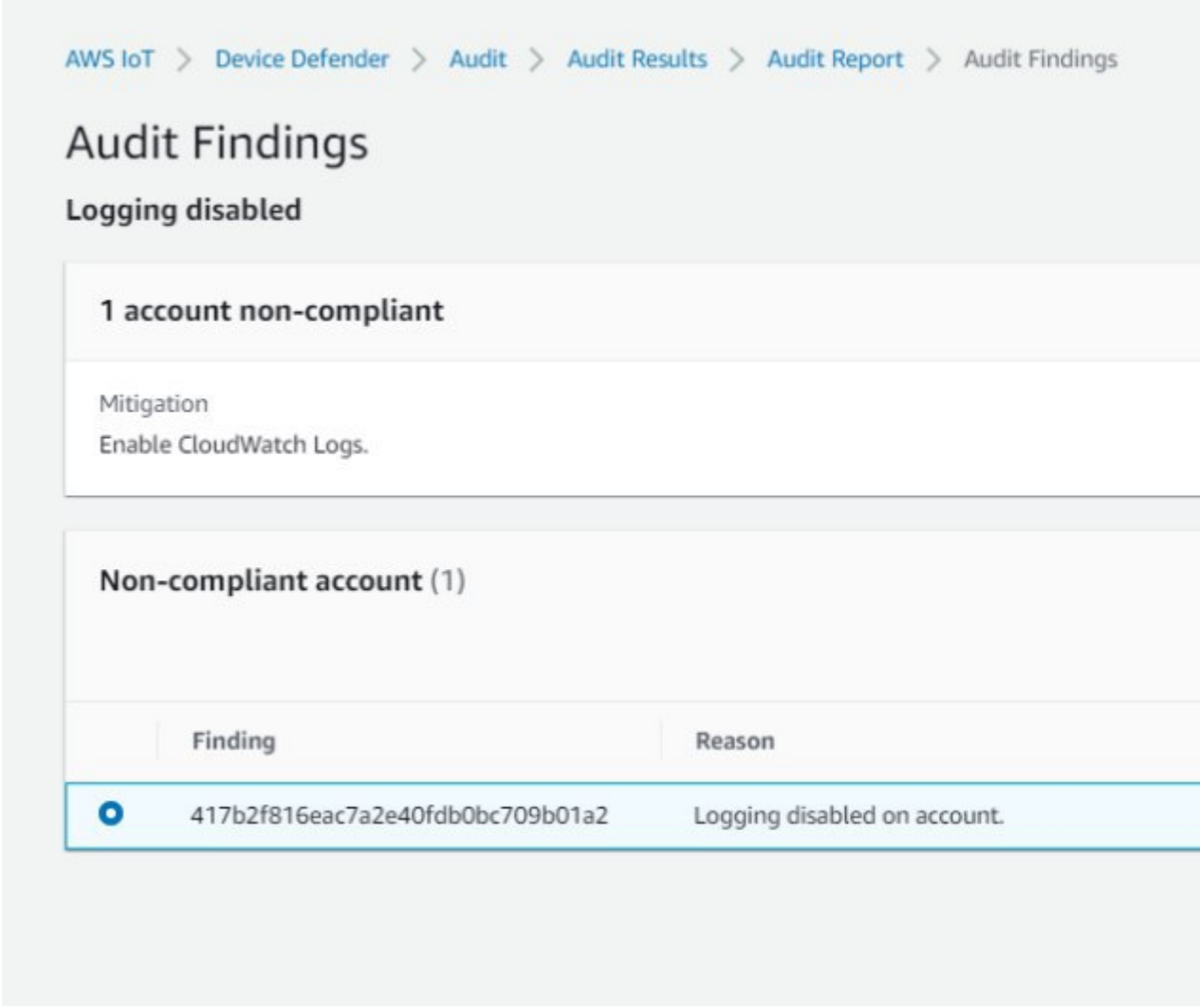


4. Dans l'écran Détails de la vérification, s'il y a des constatations que vous ne souhaitez pas voir, sélectionnez le bouton d'option en regard de la constatation. Ensuite, choisissezActions, puis

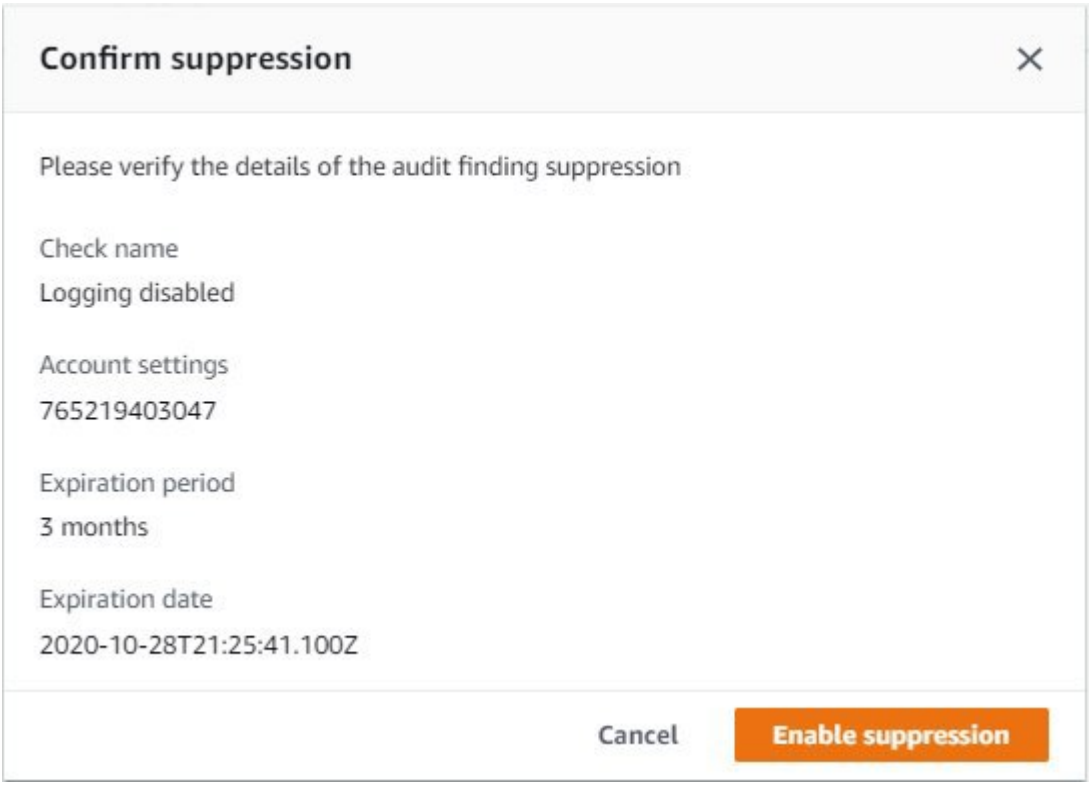
choisissez la durée pendant laquelle vous souhaitez que votre suppression de recherche d'audit persiste.

Note

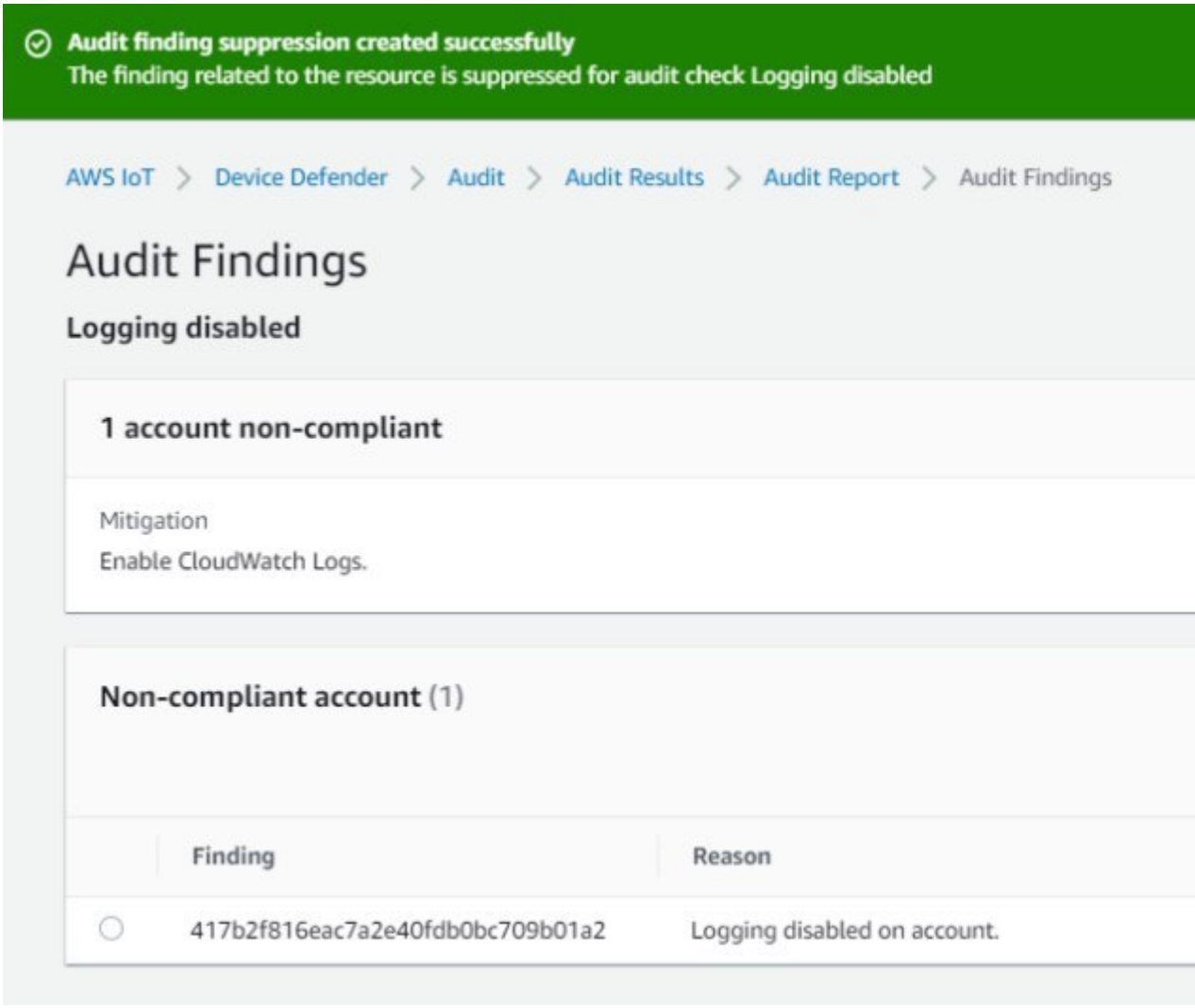
Dans la console, vous pouvez sélectionner1 semaine,1 mois,3 mois,6 mois, ouIndéfinimentcomme dates d'expiration pour votre suppression de recherche d'audit. Si vous souhaitez définir une date d'expiration spécifique, vous ne pouvez le faire que dans l'interface de ligne de commande ou l'API. Les suppressions de recherche d'audit peuvent également être annulées à tout moment, quelle que soit la date d'expiration.



5. Confirmer les détails de la suppression, puis choisissezActiver la suppression.

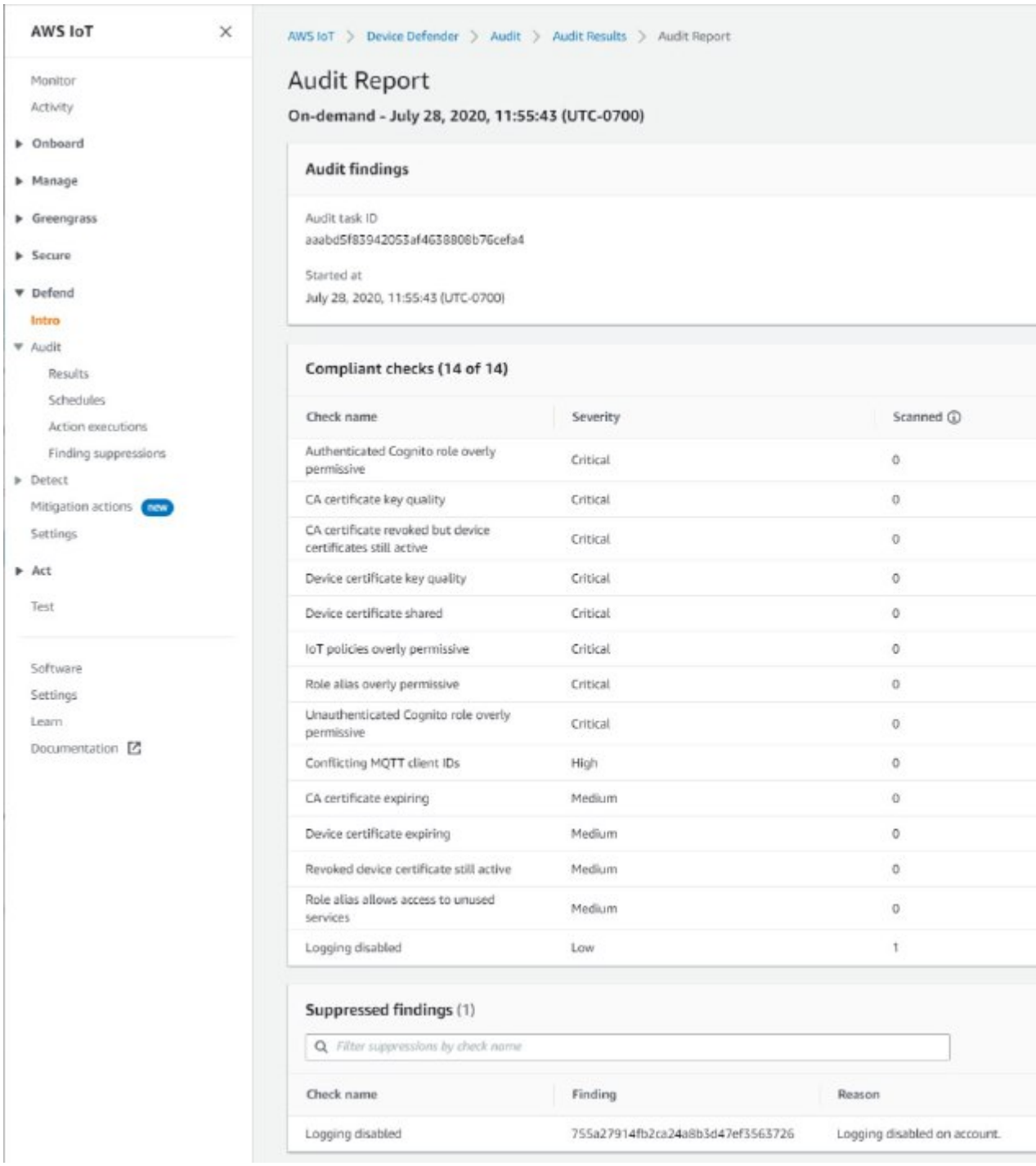


6. Une fois que vous avez créé la suppression des constatations d'audit, une bannière s'affiche confirmant que la suppression des constatations d'audit a été créée.



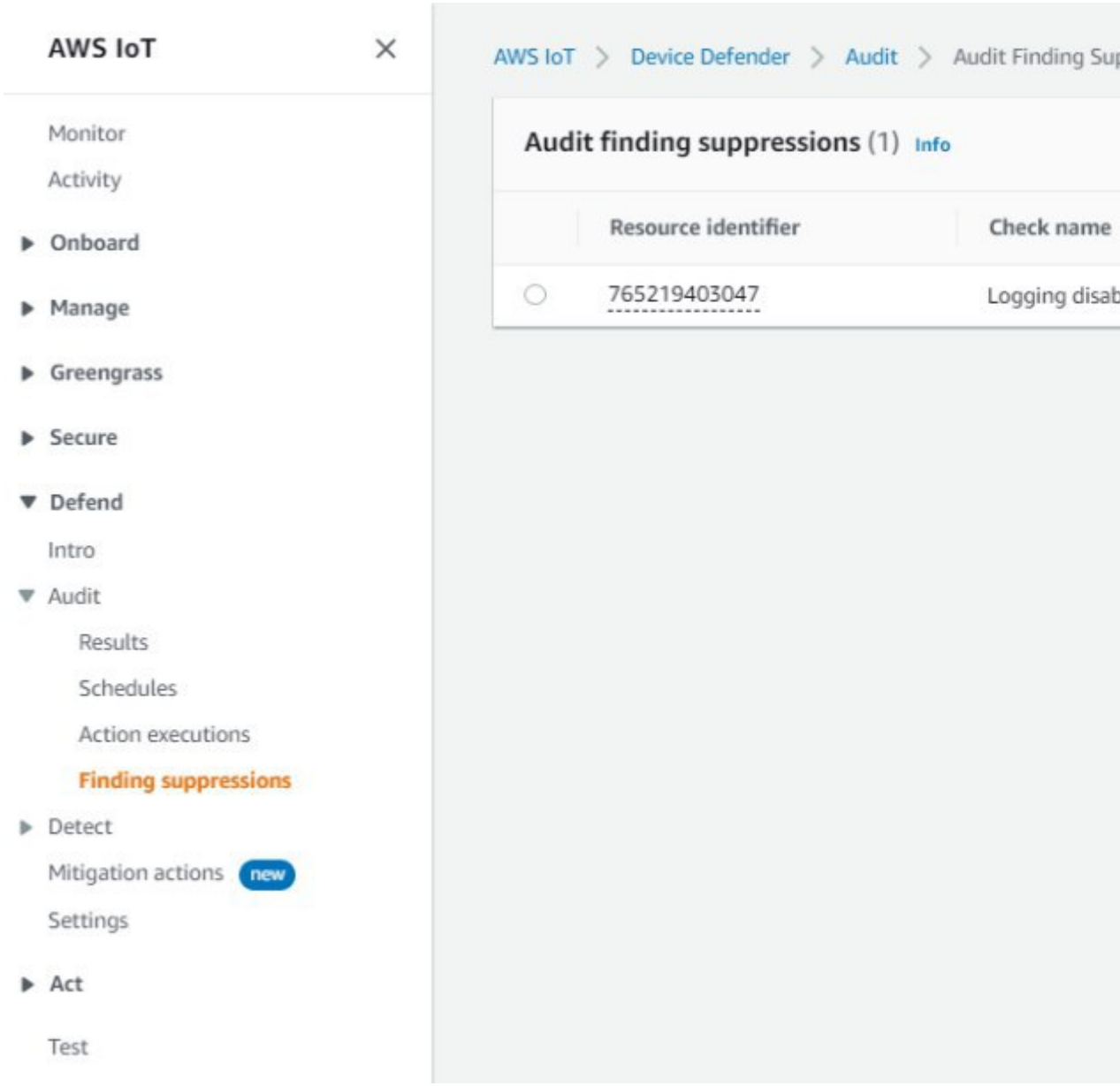
Pour afficher vos conclusions supprimées dans un rapport d'audit

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Audit,Résultats.
2. Sélectionnez un rapport d'audit que vous souhaitez consulter.
3. DansSuppression de résultats, affichez les constatations d'audit qui ont été supprimées pour le rapport d'audit que vous avez choisi.



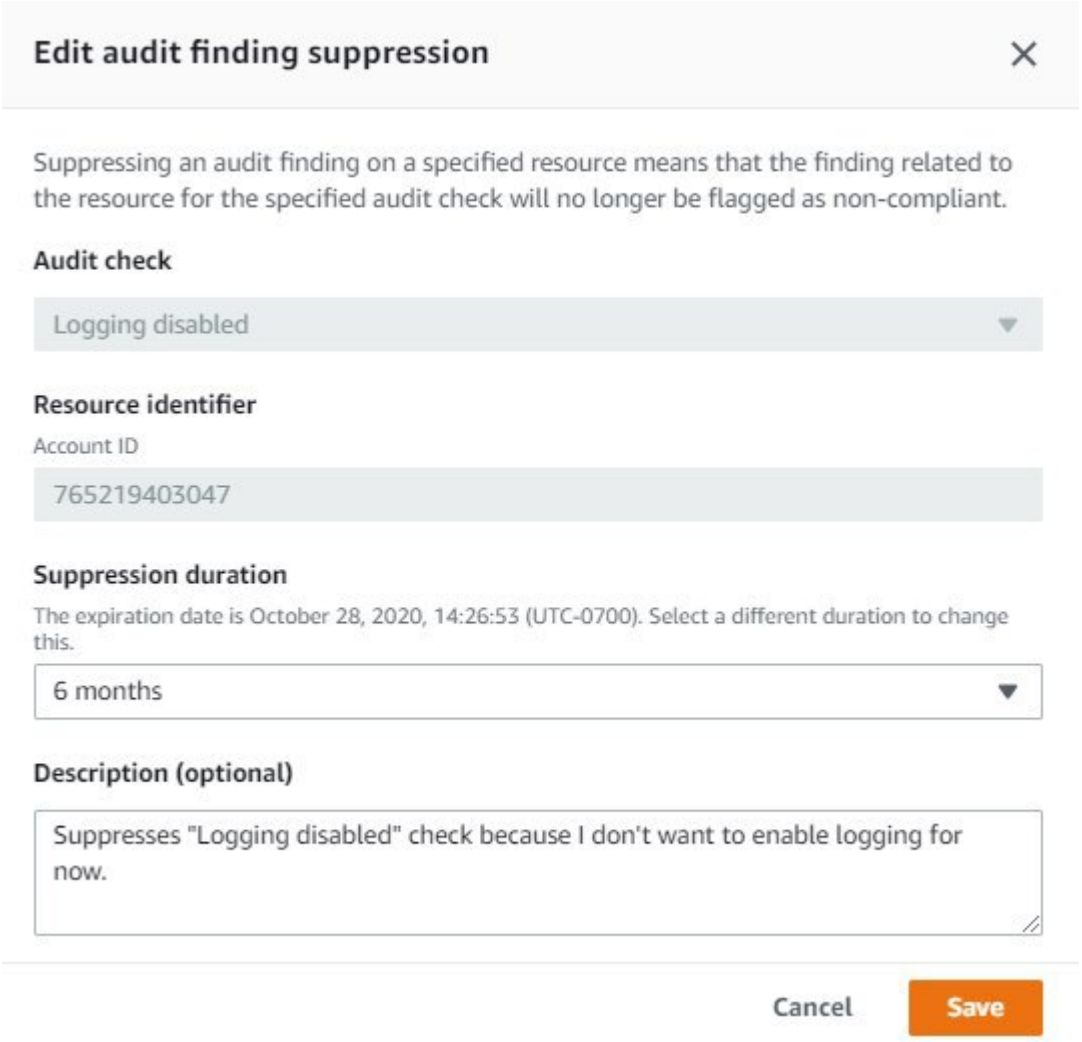
Pour répertorier vos suppressions de recherche d'audit

- DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Audit,Recherche de suppressions.



Pour modifier votre suppression de recherche d'audit

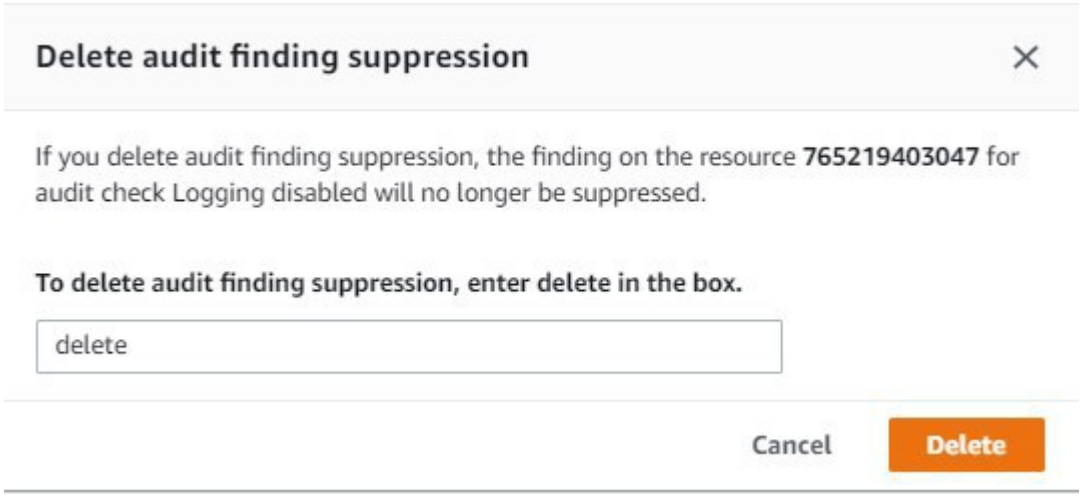
1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Audit,Recherche de suppressions.
2. Sélectionnez le bouton d'option en regard de la suppression de recherche d'audit que vous souhaitez modifier. Ensuite, choisissezActions,Modifier.
3. Dans la pageModifier la suppression des résultats d'audit, vous pouvez modifier laDurée de suppressionouDescription (facultative).



4. Une fois que vous avez apporté vos modifications, choisissezEnregistrer. La .Recherche de suppressionss'ouvre.

Pour supprimer une suppression de recherche d'audit

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Audit,Recherche de suppressions.
2. Sélectionnez le bouton d'option en regard de la suppression de recherche d'audit à supprimer, puis choisissezActions,Supprimer.
3. Dans la pageSupprimer la suppression de recherche d'auditfenêtre, entrez`delete`Dans la zone de texte pour confirmer votre suppression, puis choisissezSupprimer. La .Recherche de suppressionss'ouvre.



Comment utiliser les suppressions de recherche d'audit dans l'interface de ligne de commande

Vous pouvez utiliser les commandes CLI suivantes pour créer et gérer des suppressions de recherche d'audit.

- création-audit-suppression
- describe-audit-suppression
- update-audit-suppression
- supprimer-audit-suppression
- list-audit-suppressions

La .`resource-identifier`que vous entrez dépend de la`check-name`vous supprimez les conclusions pour. Le tableau suivant détaille les vérifications qui nécessitent`resource-identifier`pour créer et modifier des suppressions.

> **Note**
>
> Les commandes de suppression n'indiquent pas la désactivation d'un audit. Les audits s'exécuteront toujours sur votreAWS IoTAppareils. Les suppressions ne s'appliquent qu'aux constatations de la vérification.

check-name	**resource-identifier**
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	cognitoIdentityPoolId
CA_CERT_APPROACHING_EXPIRATION_CHECK	caCertificateId
CA_CERTIFICATE_KEY_QUALITY_CHECK	caCertificateId
CONFLICTING_CLIENT_IDS_CHECK	clientId
DEVICE_CERT_APPROACHING_EXPIRATION_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	deviceCertificateId
DEVICE_CERTIFICATE_SHARED_CHECK	deviceCertificateId

check-name	resource-identifier
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	policyVersionIdentifier
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	roleAliasArn
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	roleAliasArn
LOGGING_DISABLED_CHECK	account
REVOKED_CA_CERT_CHECK	caCertificateId
REVOKED_DEVICE_CERT_CHECK	deviceCertificateId
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	identityPoolId

Pour créer et appliquer une suppression de recherche d'audit

La procédure suivante vous montre comment créer une suppression de recherche d'audit dans la boîte de dialogueAWSINTERFACE DE LIGNE DE COMMANDE

- Utilisation de l'`create-audit-suppression`pour créer une suppression de recherche d'audit. L'exemple suivant crée une suppression de recherche d'audit pour Compte AWS *123456789012*sur la base de la vérificationConnexion désactivée.

```
aws iot create-audit-suppression \
    --check-name LOGGING_DISABLED_CHECK \
    --resource-identifier account=123456789012 \
    --client-request-token 28ac32c3-384c-487a-a368-c7bbd481f554 \
    --suppress-indefinitely \
    --description "Suppresses logging disabled check because I don't want to enable
 logging for now."
```

  Il n'y a pas de sortie pour cette commande.

Recherche d'audit suppressions API

Les API suivantes peuvent être utilisées pour créer et gérer les suppressions de recherche d'audit.

- CreateAuditSuppression
- DescribeEditSuppression
- UpdateAuditSuppression
- DeleteAuditSuppression
- ListAuditSuppressions

Pour filtrerpourdes résultats d'audit spécifiques, vous pouvez utiliser laListAuditFindingsAPI.

Detect

AWS IoT Device Defender Detect surveille le comportement de vos appareils afin d'identifier un comportement inhabituel pouvant indiquer qu'un appareil est endommagé. À l'aide d'une combinaison de métriques côté cloud (issues d'AWS IoT) et côté appareil (provenant d'agents que vous installez sur vos appareils), vous pouvez détecter les événements suivants :

- Changements dans les schémas de connexion.
- Périphériques qui communiquent avec des points de terminaison non autorisés ou non reconnus.
- Modifications des schémas de trafic entrant et sortant des appareils.

Vous créez des profils de sécurité qui contiennent des définitions de comportements d'appareils attendus et les affectent à un groupe d'appareils ou à tous les appareils de votre flotte.AWS IoT Device Defender Detect utilise ces profils de sécurité pour détecter les anomalies et envoyer des alarmes via les métriques Amazon CloudWatch et les notifications Amazon Simple Notification Service.

AWS IoT Device Defender Detect peut détecter les problèmes de sécurité fréquemment rencontrés dans les appareils connectés :

- Trafic d'un appareil vers une adresse IP malveillante connue ou un point de terminaison non autorisé qui indique un éventuel canal de contrôle et de commande malveillant.
- Trafic anormal, tel qu'un pic de trafic sortant, qui indique qu'un appareil participe à une attaque DDoS.
- Appareils dont les interfaces de gestion à distance et les ports sont accessibles à distance.
- Un pic de fréquence des messages envoyés à votre compte. (par exemple, à partir d'un appareil intrus qui peut entraîner des frais par message excessifs).

Cas d'utilisation :

Mesurer la surface d'attaque

Vous pouvez utiliser AWS IoT Device Defender Detect pour mesurer la surface d'attaque de vos appareils. Par exemple, vous pouvez identifier les appareils munis de ports de service souvent la cible de campagnes d'attaque (service telnet s'exécutant sur les ports 23/2323, service SSH s'exécutant sur le port 22, services HTTP/S s'exécutant sur les ports 80/443/8080/8081). Bien que ces ports de service aient de bonnes raisons d'être utilisés sur les appareils, ils font généralement partie de la surface d'attaque des adversaires et comportent des risques associés. AprèsAWS IoT Device DefenderDetect vous alerte sur la surface d'attaque, vous pouvez choisir de la réduire (en éliminant les services réseau inutilisés) ou d'effectuer des évaluations supplémentaires pour identifier les failles de sécurité (par exemple, telnet configuré avec des mots de passe courants, par défaut ou vulnérables).

Détecter les anomalies de comportement des appareils dont la cause possible est la sécurité

Vous pouvez utiliserAWS IoT Device DefenderDetect pour vous alerter des métriques de comportement des appareils (nombre de ports ouverts, nombre de connexions, port ouvert inattendu, connexions à des adresses IP inattendues) qui peuvent indiquer une faille de sécurité. Par exemple, un nombre plus élevé que prévu de connexions TCP peut indiquer qu'un appareil est utilisé pour une attaque DDoS. Une écoute de processus sur un port autre que celui attendu peut indiquer une backdoor installée sur un appareil pour un contrôle à distance. Vous pouvez utiliser AWS IoT Device Defender Detect pour tester l'état de vos flottes d'appareils et vérifier vos hypothèses de sécurité (par exemple, aucun appareil n'écoute sur le port 23 ou 2323).

Vous pouvez activer la détection des menaces basée sur le machine learning (ML) afin d’identifier automatiquement les menaces potentielles.

Détection d’un périphérique mal configuré

Un pic du nombre ou de la taille des messages envoyés d'un appareil vers votre compte peut indiquer un appareil mal configuré. Un tel appareil peut augmenter le montant de votre facture au message. De la même façon, un appareil présentant de nombreux échecs d'autorisation peut nécessiter une reconfiguration de sa stratégie.

Surveillance du comportement des appareils non enregistrés

AWS IoT Device Defender Detect permet d'identifier les comportements inhabituels pour les appareils qui ne sont pas enregistrés dans le registre d'AWS IoT. Vous pouvez définir des profils de sécurité spécifiques à un des types de cible suivants :

- Tous les appareils
- Tous les appareils enregistrés (objets dans le registre AWS IoT)
- Tous les appareils non enregistrés
- Les appareils appartenant à un groupe d'objets

Un profil de sécurité définit un ensemble de comportements attendus pour les appareils de votre compte et spécifie les actions à entreprendre lorsqu'une anomalie est détectée. Les profils de sécurité doivent être attachés aux cibles les plus spécifiques afin de vous donner un contrôle précis sur les appareils évalués par rapport à ce profil.

Les appareils non enregistrés doivent fournir un identifiant client ou un nom d'objet MQTT cohérent (pour les appareils qui notifient des métriques d'appareil) pendant la durée de vie de l'appareil afin de tous les violations et les métriques soient attribuées au même appareil.

> Important
>
> Les messages notifiés par les appareils sont rejetés si le nom d'objet contient des caractères de contrôle ou est composé de plus de 128 octets de caractères codés UTF-8.

Cas d'utilisation de sécurité

Cette section décrit les différents types d'attaques qui menacent votre parc d'appareils et les mesures recommandées que vous pouvez utiliser pour surveiller ces attaques. Nous vous recommandons d'utiliser les anomalies de mesure comme point de départ pour étudier les problèmes de sécurité, mais vous ne devez pas baser votre détermination des menaces de sécurité uniquement sur une anomalie de mesure.

Pour étudier une alarme d'anomalie, mettez en corrélation les détails de l'alarme avec d'autres informations contextuelles telles que les attributs du périphérique, les tendances historiques des métriques de périphérique, les tendances historiques des métriques du profil de sécurité, les mesures personnalisées et les journaux afin de déterminer si une menace de sécurité est présente.

Cas d'utilisation côté cloud

Device Defender peut surveiller les cas d'utilisation suivants sur leAWS IoTcôté nuage.

Vol de propriété intellectuelle :

> Le vol de propriété intellectuelle consiste à voler les propriétés intellectuelles d'une personne ou d'une entreprise, y compris les secrets commerciaux, le matériel ou les logiciels. Il se produit souvent au cours de la phase de fabrication des appareils. Le vol de propriété intellectuelle peut prendre la forme d'un piratage, d'un vol d'appareil ou d'un vol de certificat d'appareil. Le vol de propriété intellectuelle basé sur le cloud peut se produire en raison de la présence de politiques qui autorisent l'accès involontaire aux ressources IoT. Vous devez vérifier votreStratégies IoTet activezVérification des contrôles trop permissifspour identifier des politiques trop permissives.

métriques connexes :

Métrique	Justification
IP Source	Si le périphérique est volé, son adresse IP source se situerait en dehors de la plage d'adresses IP normalement attendue pour les périphériques circulant dans une chaîne d'approvisionnement normale.
Nombre de messages reçus	Étant donné qu'un attaquant peut utiliser un appareil dans le cadre d'un vol d'IP basé sur le cloud, les mesures liées au nombre de messages ou à la taille des messages envoyés à l'appareil depuisAWS IoTcloud peut augmenter, ce qui indique un problème de sécurité possible.
Taille de message	

Exfiltration de données basée sur MQT :

L'exfiltration de données se produit lorsqu'un acteur malveillant effectue un transfert de données non autorisé à partir d'un déploiement IoT ou d'un appareil. L'attaquant lance ce type d'attaques via MQTT contre des sources de données côté cloud.

métriques connexes :

Métrique	Justification
IP Source	Si un périphérique est volé, son adresse IP source se situe en dehors de la plage d'adresses IP normalement attendue pour les périphériques circulant dans une chaîne d'approvisionnement standard.
Nombre de messages reçus	Étant donné qu'un attaquant peut utiliser un périphérique dans une exfiltration de données basée sur MQT, les mesures liées au nombre de messages ou à la taille des messages envoyés à l'appareil depuisAWS IoTcloud peut augmenter, ce qui indique un problème de sécurité possible.
Taille de message	

Emprunt d'identité :

Une attaque par usurpation d'identité est le cas où les attaquants se présentent comme des entités connues ou fiables dans le but d'accéder àAWS IoTservices côté cloud, applications, données, ou s'engagent dans la commande et le contrôle des appareils IoT.

métriques connexes :

Métrique	Justification
Échecs d'autorisation	Lorsque des attaquants se présentent comme des entités de confiance à l'aide d'identités volées, les mesures liées à la connectivité augmentent souvent, car les informations d'identification peuvent ne plus être valides ou être déjà utilisées par un appareil de confiance. Les comportements anormaux dans les échecs
Tentatives de connexion	
Déconnecte	

Métrique	Justification
	d'autorisation, les tentatives de connexion ou les déconnexions pointent vers un scénario d'emprunt d'identité potentiel.

Abus de l'infrastructure cloud :

Abus deAWS IoTse produit lors de la publication ou de l'abonnement à des rubriques avec un volume de messages élevé ou avec des messages de grande taille. Les stratégies trop permissives ou l'exploitation d'une vulnérabilité de périphérique pour le contrôle et le contrôle peuvent également entraîner des abus d'infrastructure dans le cloud. L'un des principaux objectifs de cette attaque est d'augmenter votreAWSprojet de loi. Vous devez vérifier votreStratégies IoTet activezVérification des contrôles trop permissifspour identifier des politiques trop permissives.

métriques connexes :

Métrique	Justification
Nombre de messages reçus	L'objectif de cette attaque est d'augmenter votreAWS, les mesures qui surveillent les activités telles que le nombre de messages, les messages reçus et la taille des messages augmenteront.
Nombre de messages envoyés	
Taille de message	
IP Source	Des listes d'adresses IP source suspectes peuvent apparaître, à partir desquelles les attaquants génèrent leur volume de messagerie.

Cas d'utilisation côté périphérique

Device Defender peut surveiller les cas d'utilisation suivants du côté de votre appareil.

Attaque par déni de service :

Une attaque par déni de service (DoS) vise à arrêter un périphérique ou un réseau, ce qui rend le périphérique ou le réseau inaccessible aux utilisateurs auxquels ils sont destinés. Les attaques DoS bloquent l'accès en inondant la cible de trafic ou en lui envoyant des demandes qui démarrent un système ralentit ou provoquent une défaillance du système. Vos appareils IoT peuvent être utilisés dans les attaques DoS.

métriques connexes :

Métrique	Justification
Paquets sortis	Les attaques DoS impliquent généralement des taux de communication sortante plus élevés à partir d'un périphérique donné, et selon le type d'attaque DoS, il peut y avoir une augmentation du nombre de paquets sortants ou des deux.
Octets sortis	
IP de destination	Si vous définissez les adresses IP/plages CIDR avec lesquelles vos appareils doivent communiquer, une anomalie dans l'IP de destination peut indiquer une communication IP non authentifiée à partir de vos appareils.

Métrique	Justification
Ports TCP d'écoute	Une attaque DoS nécessite généralement une infrastructure de commande et de contrôle plus grande dans laquelle les logiciels malveillants installés sur vos appareils reçoivent des commandes et des informations sur les personnes à attaquer et à quel moment attaquer. Par conséquent, afin de recevoir de telles informations, le logiciel malveillant écoute généralement sur les ports qui ne sont pas normalement utilisés par vos appareils.
Nombre de ports TCP d'écoute	
Ports UDP d'écoute	
Nombre de ports UDP d'écoute	

Intensification latérale de la menace :

L'escalade des menaces latérales commence généralement par l'accès d'un attaquant à un point d'un réseau, par exemple un périphérique connecté. L'attaquant tente ensuite d'augmenter son niveau de privilèges ou son accès à d'autres appareils par des méthodes telles que des informations d'identification volées ou des exploits de vulnérabilité.

métriques connexes :

Métrique	Justification
Paquets sortis	Dans des situations typiques, l'attaquant devrait exécuter une analyse sur le réseau local afin d'effectuer une reconnexion et d'identifier les appareils disponibles afin de réduire la sélection de cibles d'attaque. Ce type d'analyse pourrait entraîner un pic d'octets et de paquets sortants.
Octets sortis	
IP de destination	Si un périphérique est censé communiquer avec un ensemble connu d'adresses IP ou de CIDR, vous pouvez identifier s'il tente de communiquer avec une adresse IP anormale, qui serait souvent une adresse IP privée sur le réseau local dans un cas d'utilisation d'escalade de menace latérale.
Échecs d'autorisation	Alors que l'attaquant tente d'augmenter son niveau de privilèges sur un réseau IoT, il peut utiliser des informations d'identification volées qui ont été révoquées ou qui ont expiré, ce qui entraînerait une augmentation des échecs d'autorisation.

Exfiltration ou surveillance des données :

L'exfiltration de données se produit lorsque un logiciel malveillant ou un acteur malveillant effectue un transfert de données non autorisé à partir d'un appareil ou d'un point de terminaison réseau. L'exfiltration de données sert normalement à deux fins pour l'attaquant, à savoir l'obtention de données ou de propriété intellectuelle, ou la reconnaissance d'un réseau. La surveillance signifie que le code malveillant est utilisé pour surveiller les activités des utilisateurs dans le but de voler des informations d'identification et de recueillir des informations. Les mesures ci-dessous peuvent fournir un point de départ pour étudier l'un ou l'autre des types d'attaques.

métriques connexes :

Métrique	Justification
Paquets sortis	En cas d'exfiltration de données ou d'attaques de surveillance, l'attaquant met souvent en miroir les données envoyées à partir de l'appareil plutôt que de simplement rediriger les données, qui seraient identifiées par le défenseur lorsqu'il ne voit pas les données prévues arriver. De telles données mises en miroir augmenteraient considérablement la quantité totale de données envoyées depuis le périphérique, ce qui entraînerait un pic de paquets et d'octets sortants.
Octets sortés	
IP de destination	Lorsqu'un attaquant utilise un appareil dans des attaques d'exfiltration de données ou de surveillance, les données doivent être envoyées à une adresse IP anormale contrôlée par l'attaquant. La surveillance de l'IP de destination peut aider à identifier une telle attaque.

CryptoCurrency

Les attaquants tirent parti de la puissance de traitement des appareils pour exploiter la crypto-monnaie. Crypto-Mining est un processus intensif en calcul, nécessitant généralement une communication réseau avec d'autres homologues et pools miniers.

métriques connexes :

Métrique	Justification
IP de destination	La communication réseau est généralement une exigence lors du cryptominage. Avoir une liste étroitement contrôlée des adresses IP avec lesquelles le périphérique doit communiquer peut aider à identifier les communications non intentionnelles sur un appareil, comme l'exploration de crypto-monnaies.
Utilisation de l'UCMétrique personnalisée	L'exploration de cryptomonnaies nécessite un calcul intensif, ce qui entraîne une utilisation élevée du processeur de l'appareil. Si vous choisissez de collecter et de surveiller cette mesure, une utilisation du processeur supérieure à la normale pourrait être un indicateur des activités de crypto-exploration.

Commande et contrôle, logiciels malveillants et rançongiciels

Les logiciels malveillants ou les ransomwares limitent votre contrôle sur vos appareils et limitent les fonctionnalités de votre appareil. Dans le cas d'une attaque par ransomware, l'accès aux données serait perdu en raison du cryptage utilisé par le ransomware.

métriques connexes :

Métrique	Justification
IP de destination	Les attaques réseau ou à distance représentent une grande partie des attaques sur les appareils IoT. Une liste étroitement contrôlée des adresses IP avec lesquelles l'appareil doit communiquer peut aider à identifier les adresses IP anormales de destination résultant d'une attaque de logiciels malveillants ou de rançongiciels.
Ports TCP d'écoute	Plusieurs attaques de logiciels malveillants impliquent le démarrage d'un serveur de commande et de contrôle qui envoie des commandes à exécuter sur un périphérique. Ce type de serveur est essentiel à un programme malveillant ou ransomware et peut être identifié en surveillant étroitement les ports TCP/UDP ouverts et le nombre de ports.
Nombre de ports TCP d'écoute	
Ports UDP d'écoute	
Nombre de ports UDP d'écoute	

Concepts

métrique

AWS IoT Device DefenderDetect utilise des métriques pour détecter les comportements anormaux des appareils.AWS IoT Device Defender Detect compare la valeur reportée d'une métrique à la valeur attendue fournie pas vous. Ces métriques peuvent provenir de deux sources : les métriques côté cloud et les métriques côté appareil. Il existe 17 mesures au total, dont 6 sont prises en charge par ML Detect. Pour obtenir une liste des métriques prises en charge par ML Detect, consultezMétriques prises en charge (p. 934).

Un comportement anormal sur le réseau AWS IoT est détecté grâce aux métriques côté cloud telles que le nombre d'échecs d'autorisation ou le nombre/la taille des messages envoyés par un appareil ou reçu via AWS IoT.

AWS IoT Device DefenderDetect peut également collecter, regrouper et surveiller les données de métriques générées parAWS IoT(par exemple, les ports qu'un appareil écoute, le nombre d'octets ou de paquets envoyés ou les connexions TCP de l'appareil).

Vous pouvez utiliser AWS IoT Device Defender Detect avec les seules métriques côté cloud. Pour utiliser des métriques côté appareil, vous devez d'abord déployer le kit de développement AWS IoT sur vos appareils ou passerelles d'appareils connectés à AWS IoT afin de collecter les métriques et les envoyer à AWS IoT. Voir Envoi de métriques à partir d'appareils (p. 956).

Profil de sécurité

Un profil de sécurité définit des comportements anormaux pour un groupe de périphériques (unGroupe d'objets (p. 210)) ou tous les appareils de votre compte, et spécifie les mesures à prendre lorsqu'une anomalie est détectée. Vous pouvez utiliser la stratégieAWS IoTPour créer un profil de sécurité et l'associer à un groupe d'appareils.AWS IoT Device Defender Detect commence à enregistrer les données relatives à la sécurité et utilise les comportements définis dans le profil de sécurité pour détecter des anomalies dans le comportement des appareils.

comportement

Un comportement indiqueAWS IoT Device DefenderDétecter comment reconnaître quand un appareil fait quelque chose d'anormal. Toute action d'un appareil ne correspondant pas à un comportement

déclenche une alerte. Un comportement de détection de règles consiste en une mesure et une valeur absolue ou un seuil statistique avec un opérateur (par exemple, inférieur ou égal à, supérieur ou égal à), qui décrivent le comportement attendu du périphérique. Un comportement ML Detect se compose d'une mesure et d'une configuration ML Detect, qui définissent un modèle ML pour apprendre le comportement normal des périphériques.

modèle ML

Un modèle ML est un modèle d'apprentissage automatique créé pour surveiller chaque comportement qu'un client configure. Le modèle s'entraîne sur les modèles de données métriques provenant de groupes de périphériques ciblés et génère trois seuils de confiance d'anomalie (élevé, moyen et faible) pour le comportement basé sur la mesure. Il déduit les anomalies en fonction des données métriques ingérées au niveau de l'appareil. Dans le contexte de ML Detect, un modèle ML est créé pour évaluer un comportement basé sur la mesure. Pour de plus amples informations, veuillez consulterDétection de détection (p. 932)

Niveau de fiabilité

ML Detect prend en charge trois niveaux de confiance :`High`,`Medium`, et`Low`.`High`confiance signifie une faible sensibilité dans l'évaluation des comportements anormaux et souvent un nombre plus faible d'alarmes.`Medium`confiance signifie une sensibilité moyenne et`Low`confiance signifie une sensibilité élevée et souvent un nombre plus élevé d'alarmes.

dimension

Vous pouvez définir une dimension pour ajuster la portée d'un comportement. Par exemple, vous pouvez définir une dimension de filtre de rubrique qui applique un comportement aux rubriques MQTT correspondant à un modèle. Pour plus d'informations sur la définition d'une dimension à utiliser dans un profil de sécurité, reportez-vous à la sectionCreateDimension.

alarme

Lorsqu'une anomalie est détectée, une notification d'alarme peut être envoyée via une métrique CloudWatch (consultezUtilisation de métriques AWS IoT (p. 359)) ou une notification SNS. Une notification d'alarme est également affichée dans leAWS IoTAvec des informations sur l'alarme et un historique des alarmes pour l'appareil. Une alarme est également envoyée lorsqu'un appareil surveillé s'arrête en présentant un comportement anormal ou lorsqu'il déclenche une alarme mais cesse les rapports pendant une période prolongée.

suppression d'alarme

Gérer les notifications de détection d'alarme SNS en définissant la notification de comportement sur`onousuppressed`. La suppression des alarmes n'empêche pas Detect d'effectuer des évaluations du comportement des appareils ; Détecter continue de signaler les comportements anormaux en tant qu'alarmes de violation. Toutefois, les alarmes supprimées ne seraient pas transmises pour notification SNS. Ils sont uniquement accessibles par l'intermédiaire de l'AWS IoTconsole ou API.

Behaviors

Un Profil de sécurité contient un ensemble de comportements. Chaque comportement contient une métrique qui spécifie le comportement normal pour un groupe d'appareils ou tous les appareils de votre compte. Les comportements se divisent en deux catégories : Règles Détecter les comportements et ML Detecter les comportements. Avec les comportements Rules Detect, vous définissez le comportement de vos appareils, alors que ML Detect utilise des modèles ML basés sur des données historiques de périphériques pour évaluer le comportement de vos appareils.

Un profil de sécurité peut être de l'un des deux types de seuil suivants : MLouBasé sur des règles. ML Security Profiles détecte automatiquement les anomalies opérationnelles et de sécurité au niveau de l'appareil sur l'ensemble de votre flotte en apprenant des données antérieures. Les profils de sécurité basés sur des règles exigent que vous définissiez manuellement des règles statiques pour surveiller les comportements de votre appareil.

La section suivante décrit certains des champs utilisés dans la définition d'un `behavior` :

Common to Rules Detect et ML Detection

`name`

Le nom du comportement.

`metric`

Le nom de la métrique utilisée (c'est-à-dire, ce qui est mesuré par le comportement).

`consecutiveDatapointsToAlarm`

Si un appareil est en violation du comportement pour le nombre spécifié de points de données consécutifs, une alarme se déclenche. Si la valeur n'est pas spécifiée, la valeur par défaut est 1.

`consecutiveDatapointsToClear`

Si une alarme s'est déclenchée et que l'appareil incriminé n'est plus en violation du comportement pour le nombre spécifié de points de données consécutifs, l'alarme est désactivée. Si la valeur n'est pas spécifiée, la valeur par défaut est 1.

`threshold type`

Un profil de sécurité peut être de l'un des deux types de seuil suivants : ML ou selon des règles. ML Security Profiles détecte automatiquement les anomalies opérationnelles et de sécurité au niveau de l'appareil sur l'ensemble de votre flotte en apprenant des données antérieures. Les profils de sécurité basés sur des règles exigent que vous définissiez manuellement des règles statiques pour surveiller les comportements de votre appareil.

`alarm suppressions`

Gérer les notifications de détection d'alarme SNS en définissant la notification de comportement sur`onousuppressed`. La suppression des alarmes n'empêche pas Detect d'effectuer des évaluations du comportement des appareils ; Détecter continue de signaler les comportements anormaux en tant qu'alarmes de violation. Toutefois, les alarmes supprimées ne sont pas transmises pour notification SNS. Ils ne peuvent être accessibles que par le biais de laAWS IoTconsole ou API.

Règles de détection

`dimension`

Vous pouvez définir une dimension pour ajuster la portée d'un comportement. Par exemple, vous pouvez définir une dimension de filtre de rubrique qui applique un comportement aux rubriques MQTT correspondant à un modèle. Pour définir une dimension à utiliser dans un profil de sécurité, consultezCreateDimension. S'applique uniquement à la détection des règles.

`criteria`

Les critères qui déterminent si un appareil se comporte normalement par rapport au `metric`.

`comparisonOperator`

L'opérateur qui lie l'objet mesuré (`metric`) aux critères (`value` ou `statisticalThreshold`).

Les valeurs possibles sont : « less-than », « less-than-equals », « greater-than », « greater-than-equals », « in-cidr-set », « not-in-cidr-set », « in-port-set » et « not-in-port-set ». Tous les opérateurs ne sont pas valides pour chaque métrique. Les opérateurs des ensembles CIDR et des ports sont uniquement utilisés avec des métriques impliquant de telles entités.

`value`

La valeur à comparer avec `metric`. Selon le type de métrique, ce champ doit contenir une valeur `count` (valeur), `cidrs` (liste de CIDR) ou `ports` (liste de ports).

`statisticalThreshold`

Le seuil de statistique par lequel une violation du comportement est déterminée. Ce champ contient un champ `statistic` qui peut prendre les valeurs suivantes : « p0 », « p0.1 », « p0.01 », « p1 », « p10 », « p50 », « p90 », « p99 », « p99.9 », « p99.99 » ou « p100 ».

Ce `statistic` indique un percentile. Il est résolu en une valeur par laquelle une violation du comportement est déterminée. Les mesures sont collectées une ou plusieurs fois sur la durée spécifiée (`durationSeconds`) à partir de tous les appareils de reporting associés à ce profil de sécurité, et les percentiles sont calculés à partir de ces données. Après quoi, des mesures sont recueillies pour un appareil données et cumulées pendant la même durée. Si la valeur obtenue pour l'appareil est supérieure ou inférieure à (`comparisonOperator`) la valeur associée au percentile spécifié, l'appareil est considéré comme étant conforme au comportement. Dans le cas contraire, l'appareil est en violation du comportement.

Un percentile indique le pourcentage de toutes les mesures étudiées qui atteignent une valeur inférieure à la valeur associée. Par exemple, si la valeur associée à «p90 » (le 90e percentile) est 123, cela signifie que 90 % de toutes les mesures étaient inférieures à 123.

`durationSeconds`

À utiliser pour spécifier la période de temps pendant laquelle le comportement est évalué pour les critères disposant d'une dimension temporelle (par exemple, `NUM_MESSAGES_SENT`). Pour une comparaison des métriques `statisticalThreshhold`, cela correspond à la période pendant laquelle les mesures sont effectuées pour tous les appareils afin de déterminer la valeur `statisticalThreshold`, puis pour chaque appareil individuellement en vue d'évaluer le classement de son comportement.

Détection de détection

`ML Detect confidence`

ML Detect prend en charge trois niveaux de confiance :`High`,`Medium`, et`Low`.`High`la confiance signifie une faible sensibilité dans l'évaluation des comportements anormaux et souvent un nombre plus faible d'alarmes,`Medium`la confiance signifie une sensibilité moyenne,`Low`confiance signifie une sensibilité élevée et souvent un nombre plus élevé d'alarmes.

Détection de détection

Avec Machine Learning Detect (ML Detect), vous créez des profils de sécurité qui utilisent l'apprentissage automatique pour apprendre les comportements attendus des périphériques en créant automatiquement des modèles basés sur les données historiques des périphériques et en affectant ces profils à un groupe de périphériques ou à tous les périphériques de votre flotte.AWS IoT Device Defenderidentifie ensuite les anomalies et déclenche des alarmes à l'aide des modèles ML.

Pour plus d'informations sur le démarrage de l'utilisation de ML Detect, consultezGuide de détection de (p. 811).

Ce chapitre contient les sections suivantes :

Cas d'utilisation de ML Detect

Vous pouvez utiliser ML Detect pour surveiller les appareils de votre flotte lorsqu'il est difficile de définir les comportements attendus des appareils. Par exemple, pour surveiller le nombre de mesures de déconnexion, il peut ne pas être clair ce qui est considéré comme un seuil acceptable. Dans ce cas, vous pouvez activer ML Detect pour identifier les points de données de mesure de déconnexion anormale basés sur les données historiques signalées à partir des périphériques.

Un autre cas d'utilisation de ML Detect consiste à surveiller les comportements des périphériques qui changent dynamiquement au fil du temps. ML Detect apprend périodiquement les comportements dynamiques attendus des périphériques en fonction de l'évolution des modèles de données des périphériques. Par exemple, le volume des messages envoyés par l'appareil peut varier entre les jours de semaine et les week-ends, et la détection de ML apprendra ce comportement dynamique.

Fonctionnement de ML Detect

À l'aide de ML Detect, vous pouvez créer des comportements pour identifier les anomalies opérationnelles et de sécurité dans6 métriques côté cloud (p. 934)and7 métriques côté appareil (p. 934). Après la période d'entraînement initiale du modèle, ML Detect actualise les modèles quotidiennement en fonction des 14 jours de données de fin. Il surveille les points de données pour ces mesures avec les modèles ML et déclenche une alarme si une anomalie est détectée.

ML Detect fonctionne mieux si vous attachez un profil de sécurité à une collection d'appareils avec des comportements attendus similaires. Par exemple, si certains de vos appareils sont utilisés chez les clients et d'autres appareils dans les bureaux d'entreprise, les modèles de comportement des appareils peuvent varier considérablement entre les deux groupes. Vous pouvez organiser les appareils dans unappareil domotique-machinegroupe d'objets et unappareil de bureauGroupe d'objets. Pour obtenir la meilleure efficacité de détection d'anomalies, associez chaque groupe de choses à un profil de sécurité ML Detect distinct.

Bien que ML Detect construit le modèle initial, il nécessite 14 jours et un minimum de 25 000 points de données par mesure sur la période de 14 jours pour générer un modèle. Par la suite, il met à jour le modèle tous les jours, il y a un nombre minimum de points de données métriques. Si la condition minimale n'est pas respectée, ML Detect tente de construire le modèle le lendemain et réessaye quotidiennement pendant les 30 prochains jours avant d'interrompre le modèle pour les évaluations.

Configuration requise

Pour la formation et la création du modèle ML initial, ML Detect a les exigences minimales suivantes.

Période de formation minimale

Il faut 14 jours pour que les modèles initiaux soient construits. Après cela, le modèle est actualisé tous les jours avec des données de mesure à partir d'une période de fin de 14 jours.

Minimum de points de données

Le minimum requis pour créer un modèle ML est de 25 000 points de données par mesure au cours des 14 derniers jours. Pour la formation continue et l'actualisation du modèle, ML Detect exige que les points de données minimales soient respectés à partir des appareils surveillés. C'est à peu près l'équivalent des configurations suivantes :

- 60 appareils se connectant et ayant une activité surAWS IoTÀ intervalles de 45 minutes.

- 40 appareils à intervalles de 30 minutes.
- 15 appareils à intervalles de 10 minutes.
- 7 appareils à intervalles de 5 minutes.

Cibles de groupe d'appareils

Pour que la collecte des données progresse, vous devez avoir des éléments dans les groupes de choses cibles pour le profil de sécurité.

Une fois le modèle initial créé, les modèles ML sont actualisés tous les jours et nécessitent au moins 25 000 points de données pour une période de fin de 14 jours.

Limitations

Vous ne pouvez actuellement pas utiliser ML Detect avec des dimensions ou avec des mesures personnalisées. Les mesures suivantes ne sont pas prises en charge avec ML Detect.

Mesures côté cloud non prises en charge avec ML Detect :

- Adresse IP source (aws:source-ip-address) (p. 962)

Mesures côté périphérique non prises en charge avec ML Detect :

- IP de destination (`aws:destination-ip-addresses`) (p. 948)
- Ports TCP d'écoute (`aws:listening-tcp-ports`) (p. 949)
- Ports UDP d'écoute (`aws:listening-udp-ports`) (p. 949)

Métriques prises en charge

Vous pouvez utiliser les métriques côté cloud suivantes avec ML Detect :

- Échecs d'autorisation (aws:num-authorization-échec) (p. 961)
- Tentatives de connexion (aws:num-connection-tentatives) (p. 963)
- Déconnecte (aws:num-déconnecte) (p. 964)
- Taille du message (aws:taille du message) (p. 958)
- Messages envoyés (aws:num-messages-envoyés) (p. 959)
- Messages reçus (aws:num-messages-reçus) (p. 960)

Vous pouvez utiliser les métriques côté appareil suivantes avec ML Detect :

- Octets sortés (`aws:all-bytes-out`) (p. 942)
- Octets dans (`aws:all-bytes-in`) (p. 943)
- Nombre de ports TCP d'écoute (`aws:num-listening-tcp-ports`) (p. 944)
- Nombre de ports UDP d'écoute (`aws:num-listening-udp-ports`) (p. 945)
- Paquets sortant (`aws:all-packets-out`) (p. 946)
- Paquets en (`aws:all-packets-in`) (p. 947)
- Nombre de connexions TCP établies (`aws:num-established-tcp-connections`) (p. 950)

Quotas de service

Pour plus d'informations sur les quotas de service et les limites de service, consultezAWS IoT Device DefenderPoints de terminaison et quotas.

Commandes de l'interface de commande ML

Vous pouvez utiliser les commandes CLI suivantes pour créer et gérer ML Detect.

- création-sécurité-profil
- attach-profil de sécurité
- liste-sécurité-profils
- profil de sécurité
- mise à jour-security-profile
- supprimer le profil de sécurité
- get-behavior-model-formation-résumés
- liste-active-violations
- liste-violation-événements

API de détection de

Les API suivantes peuvent être utilisées pour créer et gérer des profils de sécurité ML Detect.

- CreateSecurityProfile
- AttachSecurityProfile
- ListSecurityProfiles
- DescribeSecurityProfile
- UpdateSecurityProfile
- DeleteSecurityProfile
- GetBehaviorModelTrainingSummaries
- ListActiveViolations
- ListViolationEvents

Suspendre ou supprimer un profil de sécurité ML Detect

Vous pouvez suspendre votre profil de sécurité ML Detect pour arrêter temporairement la surveillance des comportements des périphériques, ou supprimer votre profil de sécurité ML Detect pour arrêter la surveillance des comportements des périphériques pendant une période prolongée.

Suspendre ML Detect Security Profile à l'aide de la

Pour suspendre un profil de sécurité ML Detect à l'aide de la console, vous devez d'abord disposer d'un groupe de choses vide. Pour créer un groupe de choses vide, consultezGroupes d’objets statiques (p. 210). Si vous avez créé un groupe de choses vide, définissez le groupe de choses vide comme cible du profil de sécurité ML Detect.

Note

Vous devez définir la cible de votre profil de sécurité à un groupe d'appareils avec des appareils dans un délai de 30 jours, sinon vous ne pourrez pas réactiver le profil de sécurité.

Supprimer ML Detect Security Profile à l'aide de la console

Pour supprimer un profil de sécurité, procédez comme suit :

1. DansAWS IoTnaviguez jusqu'à la barre latérale et choisissez l'optionDetectSection.
2. UnderDetect, choisissezDétecterpuisProfils de sécurité.
3. Choisissez le profil de sécurité ML Detect que vous souhaitez supprimer.

4. ChoisissezActions, puis dans les options, choisissezSupprimer.

   Note

   Après la suppression d'un profil de sécurité ML Detect, vous ne pourrez pas réactiver le profil de sécurité.

Suspendre un profil de sécurité ML Detect à l'aide de la CLI

Pour suspendre un profil de sécurité ML Detect à l'aide de l'interface de ligne de commande, utilisez l'outil`detach-security-security-profile`Commande de l' :

```
$aws iot detach-security-profile --security-profile-name SecurityProfileName --
security-profile-target-arn arn:aws:iot:us-east-1:123456789012:all/registered-things
```

Note

Cette option est uniquement disponible dansAWSINTERFACE DE LIGNE DE COMMANDE Comme pour le workflow de la console, vous devez définir la cible de votre profil de sécurité sur un groupe d'appareils avec des périphériques dans un délai de 30 jours, sinon vous ne pourrez pas réactiver le profil de sécurité. Pour attacher un profil de sécurité à un groupe de périphériques, utilisez l'outil`attach-security-profile`La commande.

Supprimer un profil de sécurité ML Detect à l'aide de l'interface de ligne de commande

Vous pouvez supprimer un profil de sécurité à l'aide de l'outil`delete-security-profile`Commande ci-dessous :

```
delete-security-profile --security-profile-name SecurityProfileName
```

Note

Après la suppression d'un profil de sécurité ML Detect, vous ne pourrez pas réactiver le profil de sécurité.

Métriques personnalisées

avecAWS IoT Device Defender, vous pouvez définir et surveiller des mesures propres à votre parc ou à votre cas d'utilisation, telles que le nombre d'appareils connectés à des passerelles Wi-Fi, les niveaux de charge des batteries ou le nombre de cycles d'alimentation pour les prises intelligentes. Les comportements de métriques personnalisés sont définis dans les profils de sécurité, qui spécifient les comportements attendus pour un groupe d'appareils (un groupe d'objets) ou tous les appareils. Vous pouvez surveiller les comportements en configurant des alarmes, que vous pouvez utiliser pour détecter les problèmes spécifiques aux appareils et y répondre.

Ce chapitre contient les sections suivantes :

Comment utiliser les métriques personnalisées dans la console

Didacticiels

AWS IoT Device DefenderKit SDK de l'agent (Python)

Pour commencer, téléchargez leAWS IoT Device DefenderAgent SDK (Python) exemple d'agent. L'agent rassemble les mesures et publie des rapports. Une fois que vos mesures côté périphérique sont publiées, vous pouvez afficher les mesures collectées et déterminer les seuils de configuration des alarmes. Les instructions relatives à la configuration de l'agent de périphérique sont disponibles sur leAWS IoTLisez-moi du kit SDK (Python) Device Defender Agent. Pour de plus amples informations, veuillez consulterAWS IoT Device DefenderKit SDK de l'agent (Python).

Créez une mesure personnalisée et ajoutez-la à un profil de sécurité

La procédure suivante vous montre comment créer une métriques personnalisée dans la console.

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Métriques.
2. Dans la pageMétriques personnalisées, choisissezCréer.
3. Dans la pageCréation de métrique personnalisée, procédez comme suit.

    1. UnderNom, entrez un nom pour votre métrique personnalisée. Vous ne pouvez pas modifier ce nom une fois que vous créez la métrique personnalisée.
    2. UnderNom d'affichage (facultatif), vous pouvez saisir un nom convivial pour votre métrique personnalisée. Il n'a pas besoin d'être unique et il peut être modifié après sa création.
    3. UnderType, choisissez le type de mesure que vous souhaitez surveiller. Les types de métrique incluentListe de chaîne,ip-address list,numéro-liste, etnumber. Le type ne peut pas être modifié après sa création.
    4. UnderTags (Balises), vous pouvez sélectionner les balises à associer à la ressource.

    Lorsque vous avez terminé, sélectionnezConfirmer.
4. Une fois que vous avez créé votre métrique personnalisée, la commandeMétriques personnaliséesS'affiche, où vous pouvez voir la métrique personnalisée que vous venez de créer.
5. Ensuite, vous devez ajouter votre métrique personnalisée à un profil de sécurité. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Profils de sécurité.
6. Choisissez le profil de sécurité auquel vous souhaitez ajouter votre mesure personnalisée.
7. Choisissez Actions, Edit.
8. ChoisissezMesures supplémentaires à conserver, puis choisissez votre métrique personnalisée. ChoisissezSuivantsur les écrans suivants jusqu'à ce que vous atteigniez leConfirmer. ChoisissezEnregistrerandContinuer. Une fois que votre mesure personnalisée a été ajoutée avec succès, la page Détails du profil de sécurité s'affiche.

    Note

    Les statistiques sur les centiles ne sont pas disponibles pour les métriques lorsque l'une des valeurs des métriques est un nombre négatif.

Affichage des détails de métrique personnalisée

La procédure suivante vous montre comment afficher les détails d'une métriques personnalisée dans la console.

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Métriques.
2. Cliquez sur l'ongletNom de métriquede la mesure personnalisée dont vous souhaitez afficher les détails.

Mise à jour d'une métrique personnalisée

La procédure suivante vous montre comment mettre à jour une métriques personnalisée dans la console.

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Métriques.
2. Cliquez sur le bouton d'option en regard de la mesure personnalisée que vous souhaitez mettre à jour. Puis, pourActions, choisissezModifier.
3. Dans la pageMise à jour de métrique personnalisée, vous pouvez modifier le nom complet et supprimer ou ajouter des balises.
4. Lorsque vous avez terminé, sélectionnezMise à jour. La .Métriques personnalisées.

Supprimer une métrique personnalisée

La procédure suivante vous montre comment supprimer une métriques personnalisée dans la console.

1. Tout d'abord, supprimez votre mesure personnalisée de tout profil de sécurité dans lequel elle est référencée. Vous pouvez afficher les profils de sécurité qui contiennent votre mesure personnalisée sur votre page de détails de mesure personnalisée. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Métriques.
2. Choisissez la mesure personnalisée que vous souhaitez supprimer. Supprimez la mesure personnalisée de tout profil de sécurité répertorié sousProfils de sécuritésur la page de détails de mesure personnalisée.
3. DansAWS IoTconsoleDans le volet de navigation, développezDetect, puis.Détecter,Métriques.
4. Choisissez le bouton d'option en regard de la métrique personnalisée que vous souhaitez supprimer. Puis, pourActions, choisissezSupprimer.
5. Dans la pageÊtes-vous sûr de vouloir supprimer une mesure personnalisée ?, choisissezSuppression personnalisée.

   Warning

   Une fois que vous avez supprimé une mesure personnalisée, vous perdez toutes les données associées à la mesure. Cette action ne peut pas être annulée.

Comment utiliser les métriques personnalisées à partir de l'interface de ligne de commande

Didacticiels

- AWS IoT Device DefenderKit SDK de l'agent (Python) (p. 938)
- Créez une mesure personnalisée et ajoutez-la à un profil de sécurité (p. 939)
- Affichage des détails de métrique personnalisée (p. 940)
- Mise à jour d'une métrique personnalisée (p. 940)
- Supprimer une métrique personnalisée (p. 940)

AWS IoT Device DefenderKit SDK de l'agent (Python)

Pour commencer, téléchargez leAWS IoT Device DefenderAgent SDK (Python) exemple d'agent. L'agent rassemble les mesures et publie des rapports. Une fois vos mesures côté périphérique publiées, vous

pouvez afficher les mesures collectées et déterminer les seuils de configuration des alarmes. Les instructions relatives à la configuration de l'agent de périphérique sont disponibles sur leAWS IoTLisez-moi du kit SDK (Python) Device Defender Agent. Pour de plus amples informations, veuillez consulterAWS IoT Device DefenderKit SDK de l'agent (Python).

Créez une mesure personnalisée et ajoutez-la à un profil de sécurité

La procédure suivante vous montre comment créer une métrique personnalisée et l'ajouter à un profil de sécurité à partir de l'interface de ligne de commande.

1. Utilisation de l'`create-custom-metric`pour créer votre mesure personnalisée. L'exemple suivant crée une métriques personnalisée qui mesure le pourcentage de batterie.

```
aws iot create-custom-metric \
    --metric-name "batteryPercentage" \
    --metric-type "number" \
    --display-name "Remaining battery percentage." \
    --region us-east-1
    --client-request-token "02ccb92b-33e8-4dfa-a0c1-35b181ed26b0" \
```

File d'attente:

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage"
}
```

2. Une fois que vous avez créé votre mesure personnalisée, vous pouvez soit ajouter la mesure personnalisée à un profil existant à l'aide de`update-security-profile`Ou créez un nouveau profil de sécurité pour ajouter la métrique personnalisée à l'aide de`create-security-profile`. Ici, nous créons un nouveau profil de sécurité nommé*BatteryUsage*pour ajouter notre*Pourcentage de batterie*métrique personnalisée à. Nous ajoutons également une mesure Rules Detect appelée*Bande passante cellulaireLargeur*.

```
aws iot create-security-profile \
    --security-profile-name batteryUsage \
    --security-profile-description "Shows how much battery is left in percentile."  \
    --behaviors "[{\"name\":\"great-than-75\",\"metric\":\"batteryPercentage\",
\"criteria\":{\"comparisonOperator\":\"greater-than\",\"value\":{\"number
\":75},\"consecutiveDatapointsToAlarm\":5,\"consecutiveDatapointsToClear
\":1}},{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size\",
\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
    --region us-east-1
```

File d'attente:

```
{
    "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
    "securityProfileName": "batteryUsage"
}
```

Note

Les statistiques sur les centiles ne sont pas disponibles pour les métriques lorsque l'une des valeurs des métriques est un nombre négatif.

Affichage des détails de métrique personnalisée

La procédure suivante vous montre comment afficher les détails d'une métriques personnalisée à partir de l'interface de ligne de commande.

- Utilisation de l'`list-custom-metrics`pour afficher toutes vos mesures personnalisées.

```
aws iot list-custom-metrics \
    --region us-east-1
```

La sortie de cette commande ressemble à ce qui suit.

```
{
    "metricNames": [
        "batteryPercentage"
    ]
}
```

Mise à jour d'une métrique personnalisée

La procédure suivante vous montre comment mettre à jour une métriques personnalisée à partir de l'interface de ligne de commande.

- Utilisation de l'`update-custom-metric`pour mettre à jour une métrique personnalisée. L'exemple suivant met à jour l'`display-name`.

```
aws iot update-custom-metric \
    --metric-name batteryPercentage \
    --display-name 'remaining battery percentage on device' \
    --region us-east-1
```

La sortie de cette commande ressemble à ce qui suit.

```
{
    "metricName": "batteryPercentage",
    "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage",
    "metricType": "number",
    "displayName": "remaining battery percentage on device",
    "creationDate": "2020-11-17T23:01:35.110000-08:00",
    "lastModifiedDate": "2020-11-17T23:02:12.879000-08:00"
}
```

Supprimer une métrique personnalisée

La procédure suivante vous montre comment supprimer une métriques personnalisée de l'interface de ligne de commande.

1. Pour supprimer une métrique personnalisée, commencez par la supprimer des profils de sécurité auxquels elle est attachée. Utilisation de l'`list-security-profiles`pour afficher les profils de sécurité avec une certaine mesure personnalisée.
2. Pour supprimer une mesure personnalisée d'un profil de sécurité, utilisez l'outil`update-security-profiles`La commande. Saisissez toutes les informations que vous souhaitez conserver, mais excluez la métriques personnalisée.

```
aws iot update-security-profile \
```

```
  --security-profile-name batteryUsage \
  --behaviors "[{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

La sortie de cette commande ressemble à ce qui suit.

```
{
  "behaviors": [{\"name\":\"cellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}],
  "securityProfileName": "batteryUsage",
  "lastModifiedDate": 2020-11-17T23:02:12.879000-09:00,
  "securityProfileDescription": "Shows how much battery is left in percentile.",
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-east-1:1234564789012:securityprofile/
batteryUsage",
  "creationDate": 2020-11-17T23:02:12.879000-09:00
}
```

3. Une fois la mesure personnalisée détachée, utilisez l'outil`delete-custom-metric`pour supprimer la mesure personnalisée.

```
aws iot delete-custom-metric  \
  --metric-name batteryPercentage \
  --region us-east-1
```

La sortie de cette commande ressemble à ce qui suit

```
HTTP 200
```

Commandes de l'interface de ligne de commande personnalisées

Vous pouvez utiliser les commandes CLI suivantes pour créer et gérer des métriques personnalisées.

- create-custom-metric
- describe-custom-metric
- list-custom-metrics
- mise à jour-mesure
- delete-custom-metric
- liste-sécurité-profils

API de métriques personnalisées

Les API suivantes peuvent être utilisées pour créer et gérer des métriques personnalisées.

- CreateCustomMetric
- DescribeCustommétrique
- ListCustomMetrics
- UpdateCustomMetric
- DeleteCustomMetric
- ListSecurityProfiles

Mesures côté périphérique

Lors de la création d'un profil de sécurité, vous pouvez spécifier le comportement attendu de votre appareil IoT en configurant les comportements et les seuils pour les mesures générées par les appareils IoT. Voici les métriques côté appareil, qui sont des métriques provenant d'agents que vous installez sur vos appareils.

Octets sortés (`aws:all-bytes-out`)

Le nombre d'octets sortants d'un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier la quantité maximum ou minimum (en octets) de trafic sortant qu'un appareil doit envoyer au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unité : Octets

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Outbound traffic ML behavior",
  "metric": "aws:all-bytes-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Octets dans (`aws:all-bytes-in`)

Le nombre d'octets entrants d'un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier la quantité maximum ou minimum (en octets) de trafic entrant qu'un appareil doit recevoir au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unité : Octets

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 4096
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Inbound traffic ML behavior",
  "metric": "aws:all-bytes-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Nombre de ports TCP d'écoute (`aws:num-listening-tcp-ports`)

Le nombre de ports TCP que l'appareil écoute.

Utilisez cette métrique pour spécifier le nombre maximum maximum de ports TCP que chaque appareil doit surveiller.

Compatible avec : Détection des règles | ML Detect

Unités : Échecs

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Échecs

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
```

```
  "name": "Max TCP Ports",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de la détection ML

```
{
  "name": "Max TCP Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Nombre de ports UDP d'écoute (`aws:num-listening-udp-ports`)

Le nombre de ports UDP que l'appareil écoute.

Utilisez cette métrique pour spécifier le nombre maximum maximum de ports UDP que chaque appareil doit surveiller.

Compatible avec : Détection des règles | ML Detect

Unités : Échecs

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Échecs

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 5
    },
```

```
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "Max UDP Ports",
  "metric": "aws:num-listening-udp-ports",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Max UPD Port ML behavior",
  "metric": "aws:num-listening-tcp-ports",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Paquets sortant (`aws:all-packets-out`)

Le nombre de paquets sortants d'un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier la quantité maximum ou minimum de trafic total sortant qu'un appareil doit envoyer au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Paquets

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
```

```
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "TCP outbound traffic",
  "metric": "aws:all-packets-out",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Outbound sent ML behavior",
  "metric": "aws:all-packets-out",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Paquets en (`aws:all-packets-in`)

Le nombre de paquets entrants d'un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier la quantité maximum ou minimum de trafic total entrant qu'un appareil doit recevoir au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Paquets

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 100
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example

Exemple d'utilisation de `statisticalThreshold`

```
{
  "name": "TCP inbound traffic",
  "metric": "aws:all-packets-in",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Inbound sent ML behavior",
  "metric": "aws:all-packets-in",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

IP de destination (`aws:destination-ip-addresses`)

Un ensemble de destinations IP.

Utilisez cette métrique pour spécifier un ensemble de gammes CIDR (auparavant sur liste blanche) ou non autorisés (auparavant sur liste noire) à partir duquel chaque appareil doit ou non se connecter àAWS IoT.

Compatible avec : Règles de détection

Opérateurs : in-cidr-set | not-in-cidr-set

Valeurs : une liste de CIDR

Unités : N/A

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:source-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
  },
  "suppressAlerts": true
}
```

Ports TCP d'écoute (`aws:listening-tcp-ports`)

Les ports TCP que l'appareil écoute.

Utilisez cette métrique pour spécifier un ensemble de ports TCP autorisés (auparavant sur liste blanche) ou non autorisés (auparavant sur liste noire) sur lequel chaque appareil doit ou non écouter.

Compatible avec : Règles de détection

Opérateurs : in-port-set | not-in-port-set

Valeurs : une liste de ports

Unités : N/A

Example

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 443, 80 ]
    }
  },
  "suppressAlerts": true
}
```

Ports UDP d'écoute (`aws:listening-udp-ports`)

Les ports UDP que l'appareil écoute.

Utilisez cette métrique pour spécifier un ensemble de ports UDP autorisés (auparavant sur liste blanche) ou non autorisés (auparavant sur liste noire) sur lequel chaque appareil doit ou non écouter.

Compatible avec : Règles de détection

Opérateurs : in-port-set | not-in-port-set

Valeurs : une liste de ports

Unités : N/A

Example

```
{
  "name": "Listening UDP Ports",
  "metric": "aws:listening-udp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 1025, 2000 ]
    }
  }
}
```

Nombre de connexions TCP établies (`aws:num-established-tcp-connections`)

Le nombre de connexions TCP pour un appareil.

Utilisez cette métrique pour spécifier le nombre maximum ou minimum de connexions TCP actives dont chaque appareil doit disposer (Tous les états TCP).

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Connexions

Example

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 3
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "TCP Connection Count",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 900,
    "consecutiveDatapointsToAlarm": 1,
```

```
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Connection count ML behavior",
  "metric": "aws:num-established-tcp-connections",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Spécifications des métriques d'appareil

Structure globale

Nom long	Nom court	Obligatoire	Type	Contraintes	Remarques
header	hed	O	Objet		Bloc complet requis pour rapport correct
metrics	met	O	Objet		Un rapport peut avoir les deux ou au moins un`metrics`ou`custom_metri`
custom_metrics	cmet	O	Objet		Un rapport peut avoir les deux ou au moins un`metrics`ou`custom_metri`

Bloc d'en-tête

Nom long	Nom court	Obligatoire	Type	Contraintes	Remarques
report_id	rid	O	Integer		Valeur augmentant de façon monotone. Horodatage epoch conseillé.
Version	v	O	Chaîne	Major.Minor	Incréments mineurs avec ajout de champ. Incréments majeurs si

Nom long	Nom court	Obligatoire	Type	Contraintes	Remarques
					métriques supprimées.

Blocs de métriques :

Connexions TCP

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
tcp_connections	tc	metrics	N	Objet		
established_connections	ec	tcp_connections	N	List<Connection>		État TCP établie
connections	cs	established_connections	N	List<Object>		
remote_addr	rad	connections	Y	Nombre	ip:port	IP peut être IPv6 ou IPv4
local_port	lp	connections	N	Nombre	>= 0	
local_interface	li	connections	N	Chaîne		Nom d'interface
total	t	established_connections	N	Nombre	>= 0	Nombre de connexions établies

Ports TCP d’écoute

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
listening_tcp_ports	lpts	metrics	N	Objet		
ports	pts	listening_tcp_ports	N	List<Port>	> 0	
port	pt	ports	N	Nombre	> 0	les ports doivent être des nombres supérieurs à 0
interface	if	ports	N	Chaîne		Nom d'interface
total	t	listening_tcp_ports	N	Nombre	>= 0	

Ports UDP d’écoute

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
listening_udp_ports	lupts	metrics	N	Objet		

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
ports	pts	listening_udp_ports	N	List<Port>	> 0	
port	pt	ports	N	Nombre	> 0	Les ports doivent être des nombres supérieurs à 0
interface	if	ports	N	Chaîne		Nom d'interface
total	t	listening_udp_ports	N	Nombre	>= 0	

Statistiques réseau

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
network_stats	ns	metrics	N	Objet		
bytes_in	bi	network_stats	N	Nombre	Delta Metric, >= 0	
bytes_out	bo	network_stats	N	Nombre	Delta Metric, >= 0	
packets_in	pi	network_stats	N	Nombre	Delta Metric, >= 0	
packets_out	po	network_stats	N	Nombre	Delta Metric, >= 0	

Example

La structure JSON suivante utilise des noms longs.

```
{
  "header": {
    "report_id": 1530304554,
    "version": "1.0"
  },
  "metrics": {
    "listening_tcp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 24800
        },
        {
          "interface": "eth0",
          "port": 22
        },
        {
          "interface": "eth0",
          "port": 53
        }
```

```
      ],
      "total": 3
    },
    "listening_udp_ports": {
      "ports": [
        {
          "interface": "eth0",
          "port": 5353
        },
        {
          "interface": "eth0",
          "port": 67
        }
      ],
      "total": 2
    },
    "network_stats": {
      "bytes_in": 29358693495,
      "bytes_out": 26485035,
      "packets_in": 10013573555,
      "packets_out": 11382615
    },
    "tcp_connections": {
      "established_connections": {
        "connections": [
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          },
          {
            "local_interface": "eth0",
            "local_port": 80,
            "remote_addr": "192.168.0.1:8000"
          }
        ],
        "total": 2
      }
    }
  },
  "custom_metrics": {
    "MyMetricOfType_Number": [
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
```

```
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

Example Exemple de structure JSON avec des noms courts :

```
{
  "hed": {
    "rid": 1530305228,
    "v": "1.0"
  },
  "met": {
    "tp": {
      "pts": [
        {
          "if": "eth0",
          "pt": 24800
        },
        {
          "if": "eth0",
          "pt": 22
        },
        {
          "if": "eth0",
          "pt": 53
        }
      ],
      "t": 3
    },
    "up": {
      "pts": [
        {
          "if": "eth0",
          "pt": 5353
        },
        {
          "if": "eth0",
          "pt": 67
        }
      ],
      "t": 2
    },
    "ns": {
      "bi": 29359307173,
      "bo": 26490711,
      "pi": 10014614051,
      "po": 11387620
    },
    "tc": {
      "ec": {
        "cs": [
          {
            "li": "eth0",
            "lp": 80,
            "rad": "192.168.0.1:8000"
          },
          {
            "li": "eth0",
            "lp": 80,
            "rad": "192.168.0.1:8000"
```

```
          }
        ],
        "t": 2
      }
    }
  },
  "cmet": {
    "MyMetricOfType_Number": [
      {
        "number": 1
      }
    ],
    "MyMetricOfType_NumberList": [
      {
        "number_list": [
          1,
          2,
          3
        ]
      }
    ],
    "MyMetricOfType_StringList": [
      {
        "string_list": [
          "value_1",
          "value_2"
        ]
      }
    ],
    "MyMetricOfType_IpList": [
      {
        "ip_list": [
          "172.0.0.0",
          "172.0.0.10"
        ]
      }
    ]
  }
}
```

Envoi de métriques à partir d'appareils

AWS IoT Device Defender Detect peut collecter, regrouper et surveiller les données de métriques générées par les appareils AWS IoT, pour identifier les appareils qui présentent un comportement anormal. Cette section vous explique comment envoyer des métriques d'un appareil vers AWS IoT Device Defender.

Vous devez déployer en toute sécurité leAWS IoTSDK version deux sur votreAWS IoTAppareils connectés ou passerelles d'appareils pour collecter des métriques côté appareil.AWS IoT Device Defenderfournit des exemples d'agents à utiliser comme exemples lorsque vous créez les vôtres. Si vous ne pouvez pas fournir de métriques d'appareil, vous pouvez toujours profiter des fonctionnalités limitées des métriques côté cloud. Voir la liste complète des kits SDKIci.

Il existe deux méthodes pour configurer votre appareil pour publier des mesures :AWS IoT Appareil de client etAWS IoT Device DefenderExemple d'agent. En général, vous devez utiliserAWS IoTDevice Client car il fournit un agent unique qui couvre les fonctionnalités présentes dansAWS IoT Device DefenderandAWS IoTGestion des périphériques. Ces caractéristiques comprennent les travaux, le tunnel sécurisé,AWS IoT Device Defenderla publication de mesures, et plus encore. Si vous définissez des mesures personnalisées pour votre appareil à surveiller, vous devez utiliserAWS IoT Device Defenderexemple d'agent en Python pour envoyer des données.

Utilisation du kitAWS IoTDevice Client pour publier des mesures

Pour installerAWS IoTDevice Client, vous pouvez la télécharger à partir deGithub. Une fois que vous avez installé leAWS IoTDevice Client sur le périphérique pour lequel vous souhaitez collecter des données côté périphérique, vous devez le configurer pour envoyer des mesures côté périphérique àAWS IoT Device Defender. Vérifiez que la stratégieAWS IoTClient de périphériqueFichier de configurationLes paramètres suivants sont définis dans l'`device-defender`Section:

```
"device-defender":    {
      "enabled":    true,
      "interval-in-seconds": 300
  }
```

> **Warning**
>
> Au minimum, vous devez définir l'intervalle de temps sur 300 secondes. Si vous définissez l'intervalle de temps sur moins de 300 secondes, vos données de mesure peuvent être limitées.

Après avoir mis à jour votre configuration, vous pouvez créer des profils et des comportements de sécurité dans laAWS IoT Device Defenderpour surveiller les mesures que vos appareils publient dans le cloud. Vous pouvez trouver les mesures publiées dans la AWS IoT Core en sélectionnant Défendre, Détecter, puis Métriques.

Utilisation du kitAWS IoTExemple d'agent Device Defender pour publier des mesures

Vous pouvez utiliser la stratégieAWS IoT Device Defenderexemple d'agent pour surveiller les mesures côté périphérique etmétriques personnaliséesfromAWS IoTAppareils.

Un exemple d'agent rapportant des métriques côté appareil est actuellement disponible en C à l'adresse https://github.com/aws-samples/aws-iot-device-defender-agent-c. De plus, un exemple d'agent rapportant des métriques d'appareil est actuellement disponible en Python sur GitHub à l'adresse https://github.com/aws-samples/aws-iot-device-defender-agent-sdk-python. Les mesures personnalisées sont prises en charge uniquement par l'agent d'exemple Python. Plus précisément, consultez lagreengrass_defender_agent.pypour un exemple d'agent utilisé avecAWS IoT GreengrassAppareils.

Pour utiliser les exemples d'agents ou créer vos propres agents personnalisés, vous devez installer le kit SDK pour appareils AWS IoT. Pour trouver des ressources pour votre langage de développement, consultez AWS IoT Core Ressources.

- Tous les agents doivent établir une connexion à AWS IoT et publier des métriques sur l'une de ces rubriques MQTT réservées à AWS IoT Device Defender :

  ```
  $aws/things/THING_NAME/defender/metrics/json
  ```

  ou

  ```
  $aws/things/THING_NAME/defender/metrics/cbor
  ```

  AWS IoT Device Defender utilise une de ces rubriques pour répondre en renvoyant l'état de réception de vos rapports de métriques :

  ```
  $aws/things/THING_NAME/defender/metrics/json/accepted
  ```

  ```
  $aws/things/THING_NAME/defender/metrics/cbor/accepted
  ```

```
$aws/things/THING_NAME/defender/metrics/json/rejected
```

```
$aws/things/THING_NAME/defender/metrics/cbor/rejected
```

- Afin de rapporter des métriques, un appareil doit être enregistré en tant qu'objet dans AWS IoT.
- Un appareil doit généralement envoyer un rapport de métriques une fois toutes les cinq minutes. Les appareils sont limités de sorte qu'ils ne peuvent pas effectuer plus d'un rapport de métriques toutes les cinq minutes.
- Les appareils doivent rapporter les métriques actuelles. Les rapports de métriques historiques ne sont pas pris en charge.
- Vous pouvez éventuellement utiliserJobs (p. 595)pour changer la fréquenceAWS IoT Device Defenderenvoie des métriques. Un exemple est fourni dans les exemples d'agent C AWS IoT Device Defender. Pour plus d'informations, consultez le fichier README.md.

Mesures côté cloud

Lors de la création d'un profil de sécurité, vous pouvez spécifier le comportement attendu de votre appareil IoT en configurant les comportements et les seuils pour les mesures générées par les appareils IoT. Les métriques suivantes sont des métriques côté cloud, qui sont des métriques d'AWS IoT.

Taille du message (aws:taille du message)

Nombre d'octets dans un message. Utilisez cette métrique pour spécifier la taille maximum ou minimum (en octets) de chaque message transmis à partir d'un appareil à AWS IoT.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unité : Octets

Example

```
{
  "name": "Max Message Size",
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 1024
    },
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{

  "name": "Large Message Size",
```

```
  "metric": "aws:message-byte-size",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p90"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Message size ML behavior",
  "metric": "aws:message-byte-size",
  "criteria": {
  "consecutiveDatapointsToAlarm": 1,
  "consecutiveDatapointsToClear": 1,
  "mlDetectionConfig": {
    "confidenceLevel": "HIGH"
   }
 },
  "suppressAlerts": true
}
```

Une alarme se déclenche pour un appareil si, pendant trois périodes de cinq minutes consécutives, il transmet des messages dont la taille cumulée dépasse celle mesurée pour 90 % de tous les autres appareils qui signalent ce comportement du profil de sécurité.

Messages envoyés (aws:num-messages-envoyés)

Nombre de messages envoyés par un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier le nombre maximum ou minimum de messages envoyés entreAWS IoTet chaque appareil au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Messages

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{

  "name": "Out bound message count",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 50
    },
```

```
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{

  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Messages sent ML behavior",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Messages reçus (aws:num-messages-reçus)

Nombre de messages reçus par un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier le nombre maximum ou minimum de messages reçus entreAWS IoTet chaque appareil au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Messages

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
```

```
  "name": "In bound message count",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
    },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "In bound message rate",
  "metric": "aws:num-messages-received",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Messages received ML behavior",
  "metric": "aws:num-messages-received",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Échecs d'autorisation (aws:num-authorization-échec)

Utilisez cette métrique pour spécifier le nombre maximum ou minimum d'échecs d'autorisation pour chaque appareil au cours d'une période donnée. Un échec d'autorisation se produit lorsqu'une demande d'un appareil vers AWS IoT est refusée, par exemple, si un appareil tente de publier dans une rubrique pour laquelle il ne dispose pas des autorisations suffisantes.

Compatible avec : Détection des règles | ML Detect

Unités : Échecs

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than",
    "value": {
      "count": 5
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "Authorization Failures",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p50"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Authorization failures ML behavior",
  "metric": "aws:num-authorization-failures",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Adresse IP source (aws:source-ip-address)

L'adresse IP à partir de laquelle un appareil s'est connecté à AWS IoT.

Utilisez cette métrique pour spécifier un ensemble de gammes CIDR (auparavant sur liste blanche) ou non autorisés (auparavant sur liste noire) à partir duquel chaque appareil doit ou non se connecter àAWS IoT.

Compatible avec : Règles de détection

Opérateurs : in-cidr-set | not-in-cidr-set

Valeurs : une liste de CIDR

Unités : N/A

Example

```
{
  "name": "Denied source IPs",
  "metric": "aws:source-ip-address",
  "criteria": {
    "comparisonOperator": "not-in-cidr-set",
    "value": {
      "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]
    }
  },
  "suppressAlerts": true
}
```

Tentatives de connexion (aws:num-connection-tentatives)

Nombre de tentatives de connexion d'un appareil au cours d'une période donnée.

Utilisez cette métrique pour spécifier le nombre maximum ou minimum de tentatives de connexion de chaque appareil. Les tentatives réussies et infructueuses sont comptabilisées.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Tentatives de connexion

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
  "name": "Connection Attempts",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p10"
```

```
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

Déconnecte (aws:num-déconnecte)

Nombre de fois où un appareil s'est déconnecté d'AWS IoT au cours d'une période donnée.

Utilisez cette métrique pour spécifier le nombre maximal ou minimal de fois qu'un appareil s'est déconnecté d'AWS IoT au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than | greater-than | greater-than | greater-than | greater-than-equal | greater-than-equal

Valeur : Nombre entier non négatif.

Unités : Déconnexions

Durée : Nombre entier non négatif. Les valeurs valides sont 300, 600, 900, 1 800 ou 3 600 secondes.

Example

```
{
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **`statisticalThreshold`**

```
{
```

```
  "name": "Disconnections",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equal",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple utilisant ML Detect

```
{
  "name": "Disconnects ML behavior",
  "metric": "aws:num-disconnects",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": true
}
```

Définition de la portée des métriques dans les profils de sécurité à l'aide de dimensions

Les dimensions sont des attributs que vous pouvez définir pour obtenir des données plus précises sur les métriques et les comportements dans votre profil de sécurité. Vous définissez la portée en fournissant une valeur ou un modèle servant de filtre. Par exemple, vous pouvez définir une dimension de filtre de rubrique qui applique une métrique uniquement aux rubriques MQTT qui correspondent à une valeur particulière, par exemple « data/bulb/+/activity ». Pour plus d'informations sur la définition d'une dimension à utiliser dans votre profil de sécurité, reportez-vous à la sectionCreateDimension.

Les valeurs de dimension prennent en charge les caractères génériques MQTT. Les caractères génériques MQTT vous aident à vous abonner à plusieurs rubriques simultanément. Il existe deux types différents de caractères génériques : à un seul niveau (+) et à plusieurs niveaux (#). Par exemple, la valeur de dimension `Data/bulb/+/activity` crée un abonnement qui correspond à toutes les rubriques qui existent au même niveau que le +. Les valeurs de dimension prennent également en charge la variable de substitution d'ID client MQTT $ {iot:ClientId}.

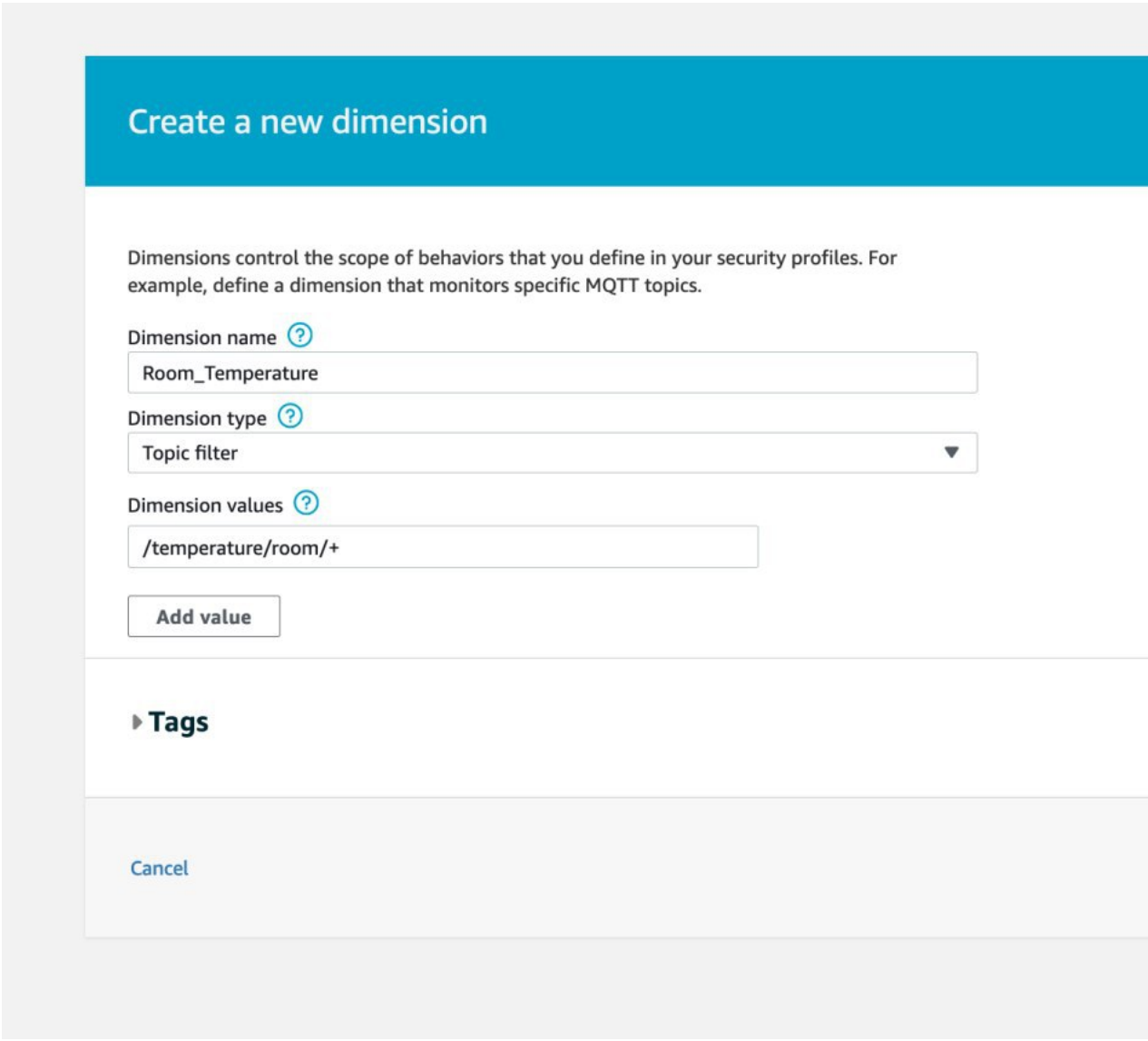
Les dimensions de type TOPIC_FILTER sont compatibles avec l'ensemble de métriques côté cloud suivant :

- Nombre de messages envoyés
- Nombre de messages reçus
- Taille en octets des messages
- Adresse IP source
- Nombre d'échecs d'autorisation

Comment utiliser les dimensions dans la console

Pour créer et appliquer une dimension à un comportement de profil de sécurité

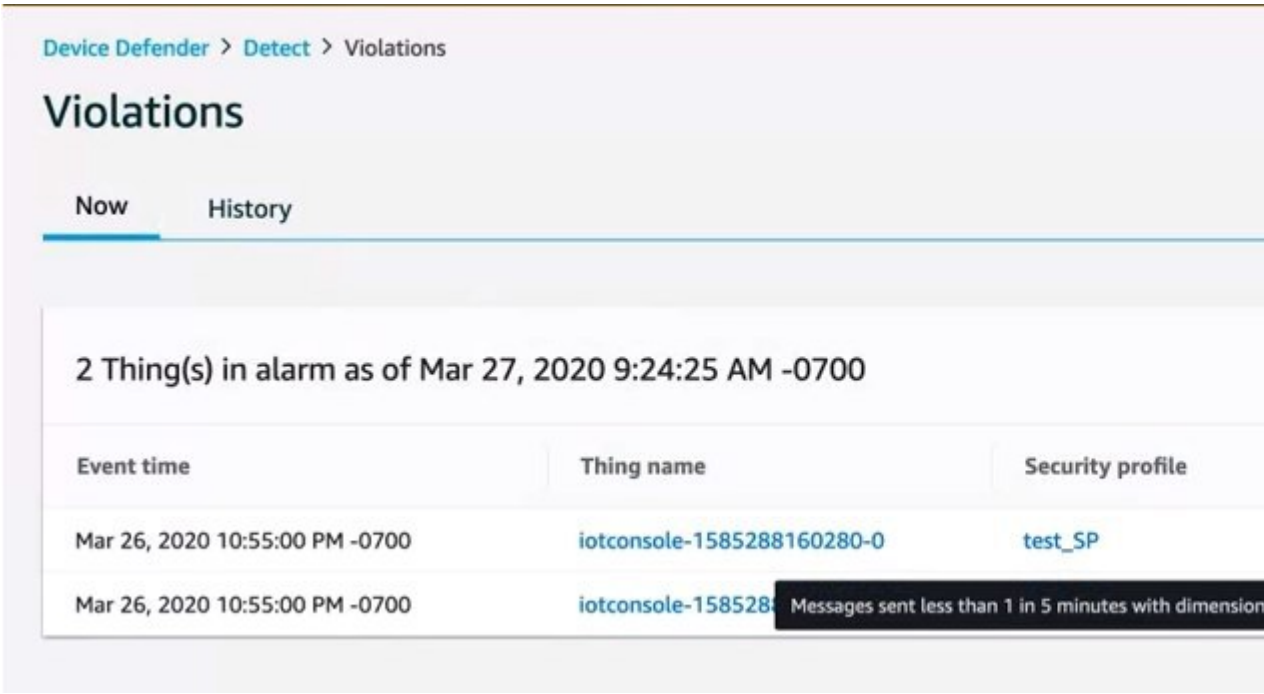
1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, DéveloppezDétecter, puis.Profils de sécurité.
2. Dans la page Profils de sécurité choisissez Créer pour ajouter un nouveau profil de sécurité ou Modifier pour appliquer une dimension à un profil de sécurité existant.
3. Dans la page Expected Behaviors (Comportements attendus) sélectionnez l'une des cinq dimensions de métriques côté cloud prises en charge sous Métriques. Les zones Dimension et Dimension operator (Opérateur de la dimension) s’affichent. Pour plus d'informations sur les métriques côté cloud prises en charge, reportez-vous à la section Définition de la portée des métriques dans les profils de sécurité à l'aide de dimensions (p. 965).
4. PourDimension, choisissezAjouter une dimension.
5. Dans la page Create a new dimension (Créer une nouvelle dimension) entrez les détails de votre nouvelle dimension. Les valeurs de dimension prennent en charge les caractères génériques MQTT # et + ainsi que la variable de substitution d'ID client MQTT ${iot:ClientId}.



6. Choisissez Enregistrer.
7. Vous pouvez éventuellement ajouter des dimensions aux métriques sousMesures supplémentaires à conserver.
8. Pour terminer la création du comportement, tapez les informations dans les autres champs obligatoires, puis choisissezSuivant.
9. Effectuez les étapes restantes pour terminer la création d'un profil de sécurité.

Pour afficher vos violations

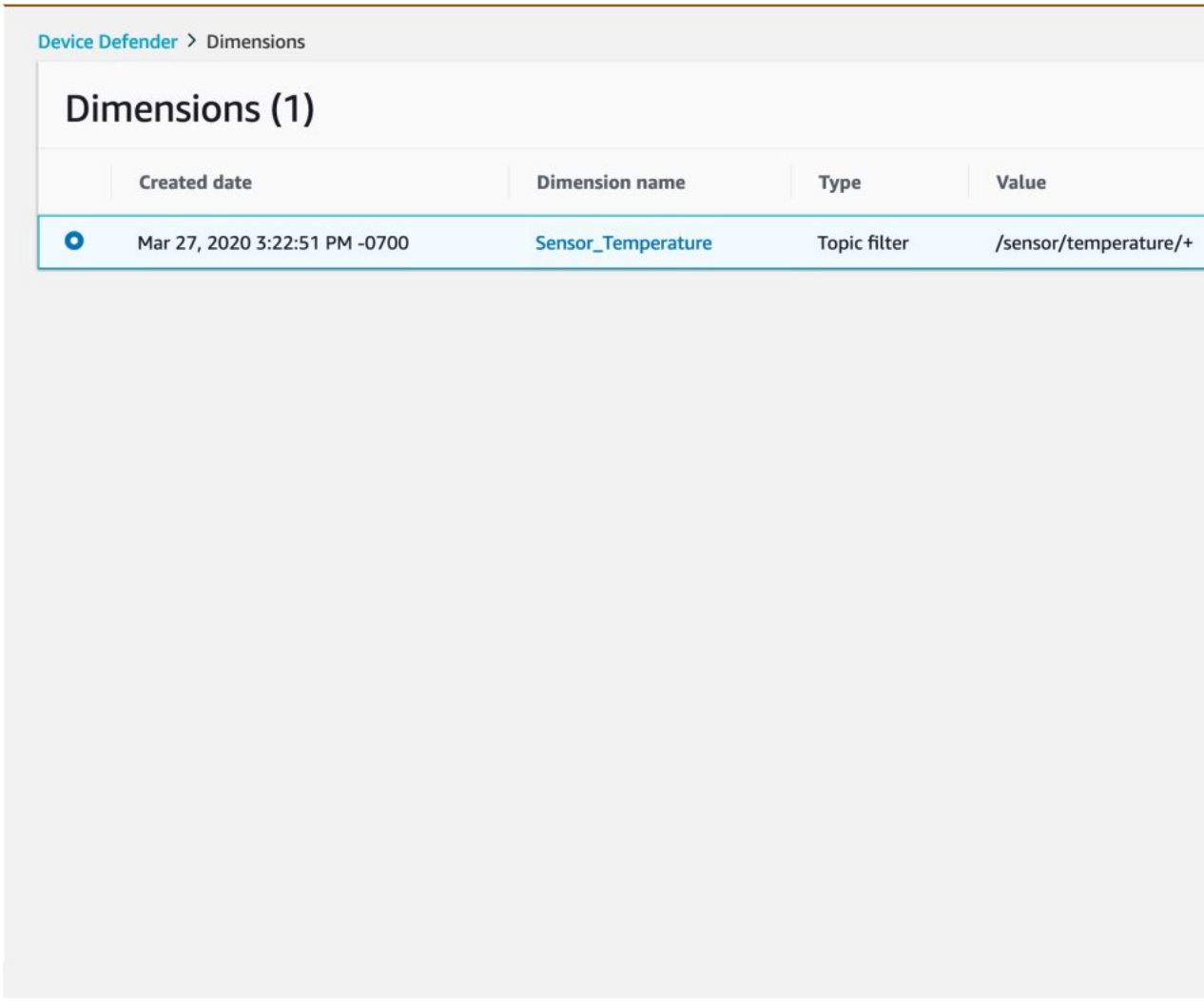
1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, DéveloppezDétecter, puis.Violations.



2. DansBehavior, faites une pause sur le comportement pour lequel vous souhaitez afficher les informations de violation.

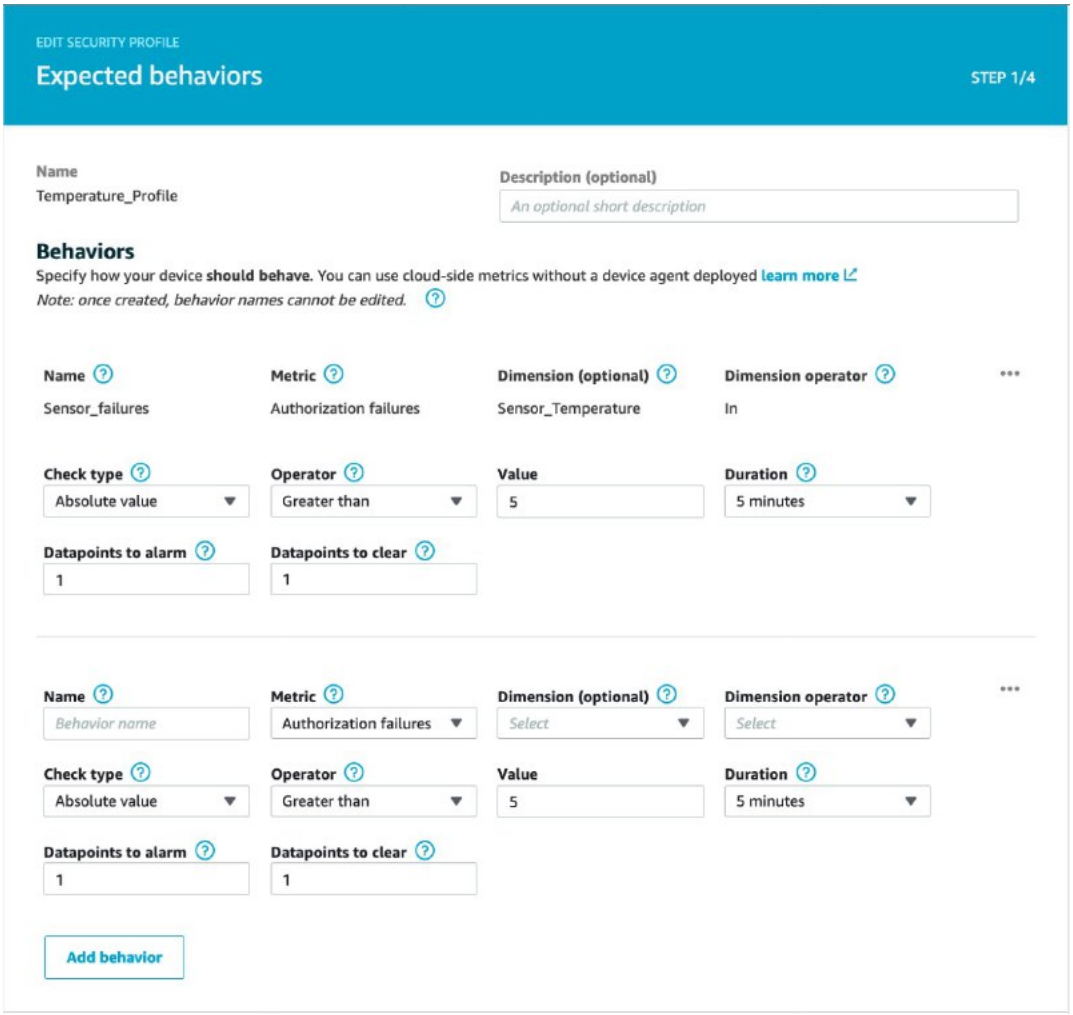
Pour afficher et mettre à jour vos dimensions

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, DéveloppezDétecter, puis.Dimensions.
2. Sélectionnez la dimension à modifier.
3. Choisissez Actions, puis Modifier.



Pour supprimer une dimension

1. DansAWS IoTconsoleDans le volet de navigation, développezDetect, DéveloppezDétecter, puis.Dimensions.
2. Sélectionnez la dimension à supprimer.
3. Vérifiez que la dimension n'est pas attachée à un profil de sécurité en cochant la colonne Used in (Utilisé dans). Si la dimension est attachée à un profil de sécurité, ouvrez la page Profils de sécurité sur la gauche et modifiez les profils de sécurité auxquels la dimension est attachée. Lorsque vous supprimez la dimension, vous supprimez également le comportement. Si vous souhaitez conserver le comportement, choisissez les points de suspension, puis Copier. Vous pouvez ensuite supprimer le comportement. Si vous souhaitez supprimer une autre dimension, suivez la procédure présentée dans cette section.



4. Choisissez Actions, puis choisissez Delete.

Comment utiliser les dimensions sur l'interface de ligne de commandeAWS CLI

Pour créer et appliquer une dimension à un comportement de profil de sécurité

1. Commencez par créer la dimension avant de l'attacher à un profil de sécurité. Utilisation de l'CreateDimensionPour créer une dimension :

```
aws iot create-dimension \
  --name TopicFilterForAuthMessages \
  --type TOPIC_FILTER \
  --string-values device/+/auth
```

La sortie de cette commande ressemble à ce qui suit :

```
{
    "arn": "arn:aws:iot:us-west-2:123456789012:dimension/TopicFilterForAuthMessages",
    "name": "TopicFilterForAuthMessages"
```

```
}
```

2. Ajoutez la dimension à un profil de sécurité existant à l'aide deUpdateSecurityProfileou ajoutez la dimension à un nouveau profil de sécurité à l'aide deCreateSecurityProfile. Dans l'exemple suivant, nous créons un nouveau profil de sécurité qui vérifie si les messages vers `TopicFilterForAuthMessages` font moins de 128 octets et qui conserve le nombre de messages envoyés à des rubriques non autorisées.

```
aws iot create-security-profile \
  --security-profile-name ProfileForConnectedDevice \
  --security-profile-description "Check to see if messages to
 TopicFilterForAuthMessages are under 128 bytes and retains the number of messages sent
 to non-auth topics." \
  --behaviors "[{\"name\":\"CellularBandwidth\",\"metric\":\"aws:message-byte-size
\",\"criteria\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":10},\"durationSeconds\":300,
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]" \
  --additional-metrics-to-retain-v2 "[{\"metric\": \"aws:num-authorization-failures\",
\"metricDimension\": {\"dimensionName\": \"TopicFilterForAuthMessages\",\"operator\":
 \"NOT_IN\"}}]"
```

La sortie de cette commande ressemble à ce qui suit :

```
{
    "securityProfileArn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice",
    "securityProfileName": "ProfileForConnectedDevice"
}
```

Pour gagner du temps, vous pouvez également charger un paramètre à partir d'un fichier au lieu de le taper comme valeur de paramètre de ligne de commande. Pour de plus amples informations, veuillez consulterchargement en coursAWS CLIParamètres d'un fichier. Le code suivant illustre le paramètre `behavior` au format JSON étendu :

```
[
  {
    "criteria": {
      "comparisonOperator": "less-than",
      "consecutiveDatapointsToAlarm": 1,
      "consecutiveDatapointsToClear": 1,
      "value": {
        "count": 128
      }
    },
    "metric": "aws:message-byte-size",
    "metricDimension": {
      "dimensionName:": "TopicFilterForAuthMessages"
    },
    "name": "CellularBandwidth"
  }
]
```

Pour afficher les profils de sécurité avec une dimension

- Utilisation de l'ListSecurityProfilesCommande pour afficher les profils de sécurité avec une certaine dimension :

```
aws iot list-security-profiles \
  --dimension-name TopicFilterForAuthMessages
```

La sortie de cette commande ressemble à ce qui suit :

```
{
    "securityProfileIdentifiers": [
        {
            "name": "ProfileForConnectedDevice",
            "arn": "arn:aws:iot:us-west-2:1234564789012:securityprofile/
ProfileForConnectedDevice"
        }
    ]
}
```

Pour mettre à jour votre dimension

- Utilisation de l'UpdateDimensionCommande pour mettre à jour une dimension :

```
aws iot update-dimension \
  --name TopicFilterForAuthMessages \
  --string-values device/${iot:ClientId}/auth
```

La sortie de cette commande ressemble à ce qui suit :

```
{
    "name": "TopicFilterForAuthMessages",
    "lastModifiedDate": 1585866222.317,
    "stringValues": [
        "device/${iot:ClientId}/auth"
    ],
    "creationDate": 1585854500.474,
    "type": "TOPIC_FILTER",
    "arn": "arn:aws:iot:us-west-2:1234564789012:dimension/TopicFilterForAuthMessages"
}
```

Pour supprimer une dimension

1. Pour supprimer une dimension, commencez par la détacher des profils de sécurité auxquels elle est attachée. Utilisation de l'ListSecurityProfilesCommande pour afficher les profils de sécurité avec une certaine dimension.
2. Pour supprimer une dimension d'un profil de sécurité, utilisez l'outilUpdateSecurityProfileLa commande. Saisissez toutes les informations que vous souhaitez conserver, mais excluez la dimension :

```
aws iot update-security-profile \
  --security-profile-name ProfileForConnectedDevice \
  --security-profile-description "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128" \
  --behaviors "[{\"name\":\"metric\":\"aws:message-byte-size\",\"criteria
\":{\"comparisonOperator\":\"less-than\",\"value\":{\"count\":128},
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}},{\"name
\":\"Authorization\",\"metric\":\"aws:num-authorization-failures\",\"criteria\":
{\comparisonOperator\":\"less-than\",\"value\"{\"count\":10},\"durationSeconds\":300,
\"consecutiveDatapointsToAlarm\":1,\"consecutiveDatapointsToClear\":1}}]"
```

La sortie de cette commande ressemble à ce qui suit :

```
{
  "behaviors": [
    {
      "metric": "aws:message-byte-size",
      "name": "CellularBandwidth",
      "criteria": {
        "consecutiveDatapointsToClear": 1,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 128
        }
      }
    },
    {
      "metric": "aws:num-authorization-failures",
      "name": "Authorization",
      "criteria": {
        "durationSeconds": 300,
        "comparisonOperator": "less-than",
        "consecutiveDatapointsToClear": 1,
        "consecutiveDatapointsToAlarm": 1,
        "value": {
          "count": 10
        }
      }
    }
  ],
  "securityProfileName": "ProfileForConnectedDevice",
  "lastModifiedDate": 1585936349.12,
  "securityProfileDescription": "Check to see if authorization fails 10 times in 5
 minutes or if cellular bandwidth exceeds 128",
  "version": 2,
  "securityProfileArn": "arn:aws:iot:us-west-2:123456789012:securityprofile/Preo/
ProfileForConnectedDevice",
  "creationDate": 1585846909.127
}
```

3. Une fois la dimension détachée, utilisez l'DeleteDimensionPour supprimer la dimension :

```
aws iot delete-dimension \
  --name TopicFilterForAuthMessages
```

Permissions

Cette section contient des informations sur la manière de configurer les rôles et les stratégies IAM requises pour gérerAWS IoT Device DefenderDétect. Pour plus d'informations, consultez le .Guide de l'utilisateur IAM.

DERAWS IoT Device Defenderdétecter l'autorisation de publier des alarmes dans une rubrique SNS

Si vous utilisez la stratégie`alertTargets`dansCreateSecurityProfile, vous devez spécifier un rôle IAM avec deux stratégies : une stratégie d'autorisation et une stratégie d'approbation. La stratégie d'autorisation accorde à AWS IoT Device Defender l'autorisation de publier des notifications dans votre rubrique SNS. La stratégie d'approbation accorde à AWS IoT Device Defender l'autorisation d'assumer le rôle requis.

Stratégie d'autorisation

```
{
  "Version":"2012-10-17",
  "Statement":[
      {
        "Effect":"Allow",
        "Action":[
          "sns:Publish"
        ],
        "Resource":[
          "arn:aws:sns:region:account-id:your-topic-name"
        ]
    }
  ]
}
```

Stratégie d'approbation

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

Stratégie de transmission de rôle

Vous avez également besoin d'une stratégie d'autorisations IAM attachée à l'utilisateur IAM qui permet à l'utilisateur de transférer des rôles. Consultez Octroi d'autorisations à un utilisateur pour transférer un rôle à un service AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Action": [
          "iam:GetRole",
          "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::account-id:role/Role_To_Pass"
    }
  ]
}
```

Commandes Detect

Vous pouvez utiliser les commandes d'Detect présentées dans cette section pour configurer ML Detect ou Rules Detect des profils de sécurité, afin d'identifier et surveiller les comportements inhabituels susceptibles d'indiquer qu'un périphérique est endommagé.

Commandes d'action d'atténuation

Lancer et gérer l'exécution Détecter
CancelDetectMitigationActionStask
DescribeDetectMitigationActionsTask
ListDetectMitigationActionStasks
DémarrerDetectMitigationActionStask
ListDetectMitigationActionsExecutions

Commandes d'action de dimension

Lancer et gérer l'exécution de dimension
CreateDimension
DescribeDimension
ListDimensions
DeleteDimension
UpdateDimension

Commandes d'action CustomMe

Lancer et gérer l'exécution CustomMetric
CreateCustomMetric
UpdateCustomMetric
DescribeCustommétrique
ListCustomMetrics
DeleteCustomMetric

Commandes d'action Profil de sécurité

Lancer et gérer l'exécution du profil de sécurité
CreateSecurityProfile
AttachSecurityProfile
DeleteSecurityProfile
DescribeSecurityProfile
ListTargetsForSecurityProfile
UpdateSecurityProfile
ValidateSecurityProfileBehaviors

Lancer et gérer l'exécution du profil de sécurité
ListSecurityProfilesForTarget

Commandes d'action d'alarme

Gestion des alarmes et des cibles
ListActiveViolations
ListViolationEvents
DetachSecurityProfile

Commandes d'action ML

Liste les données de formation de modèle ML
GetBehaviorModelTrainingSummaries

Utilisation d'AWS IoT Device Defender Detect

1. Vous pouvez utiliser AWS IoT Device Defender Detect avec uniquement les métriques côté cloud, mais si vous envisagez d'utiliser les métriques notifiées par les appareils, vous devez d'abord déployer le kit de développement SDK AWS IoT sur vos appareils connectés à AWS IoT ou sur vos passerelles d'appareil. Pour plus d'informations, consultez Envoi de métriques à partir d'appareils (p. 956).
2. Pensez à prendre connaissance des métriques que vos appareils génèrent avant de définir des comportements et de créer des alarmes. AWS IoT peut collecter les métriques à partir de vos appareils de sorte que vous puissiez d'abord identifier un comportement habituel ou inhabituel pour un groupe d'appareils ou pour tous les appareils de votre compte. UtiliserCreateSecurityProfile, mais spécifiez uniquement les`additionalMetricsToRetain`qui vous intéresse. Ne spécifiez pas`behaviors`À ce stade.

   Utilisez la console AWS IoT pour examiner vos métriques d'appareil et voir ce qui constitue un comportement normal pour vos appareils.
3. Créez un ensemble de comportements pour votre profil de sécurité. Les comportements contiennent des métriques qui spécifient un comportement normal pour un groupe d'appareils ou tous les appareils de votre compte. Pour plus d'informations et d'exemples, consultez Mesures côté cloud (p. 958) et Mesures côté périphérique (p. 942). Après avoir créé un ensemble de comportements, vous pouvez les valider avecValidateSecurityProfileBehaviors.
4. Utilisation de l'CreateSecurityProfileaction pour créer un profil de sécurité incluant vos comportements. Vous pouvez utiliser la stratégie`alertTargets`Pour envoyer des alarmes à une cible (une rubrique SNS) lorsqu'un appareil ne respecte pas un comportement. (Si vous envoyez des alarmes à l'aide de SNS, sachez que celles-ci sont comptabilisées pour votre Compte AWS du quota de sujet SNS. Il est possible qu'un grand nombre de violations dépasse votre quota de rubriques SNS. Vous pouvez également utiliser les métriques CloudWatch pour vérifier les violations. Pour plus d'informations, consultez

   Les métriques présentées par AWS IoT fournissent des informations qui permettent divers types d'analyses. Les cas d'utilisation suivants sont basés sur un scénario où vous avez dix objets qui se connectent à Internet une fois par jour. Chaque jour :

   - Dix objets se connectent à AWS IoT en même temps.

- Chaque objet s'abonne à un filtre de rubrique, puis attend une heure avant de déconnecter. Au cours de cette période, les objets communiquent entre eux pour en savoir plus sur l'état du monde.
- Chaque objet publie sa perception, d'après les données qu'il vient de détecter avec `UpdateThingShadow`.
- Chaque objet se déconnecte de AWS IoT.

Pour vous aider à démarrer, ces rubriques explorent certaines des questions que vous pourriez avoir.

- Comment puis-je être informé si mes objets ne se connectent pas chaque jour ? (p. 360)
- Comment puis-je être informé si mes objets ne publient pas de données chaque jour ? (p. 361)
- Comment puis-je être informé si les mises à jour du shadow de mon objet sont rejetées chaque jour ? (p. 361)
- Comment créer une alarme CloudWatch pour les travaux ? (p. 362)

(p. 359).

5. Utilisation de l'AttachSecurityProfileafin d'attacher le profil de sécurité à un groupe d'appareils (groupe d'objets), tous les objets enregistrés dans votre compte, tous les objets non enregistrés ou tous les appareils.AWS IoT Device Defender Detect lance le contrôle de comportements anormaux et, si des violations de comportement sont détectées, envoie des alarmes. Vous pouvez attacher un profil de sécurité à tous les objets non enregistrés si, par exemple, vous envisagez d'interagir avec les appareils mobiles qui ne font pas partie du registre d'objets de votre compte. Vous pouvez définir différents ensembles de comportements pour différents groupes d'appareils afin de répondre à vos besoins.

   Pour attacher un profil de sécurité à un groupe d'appareils, vous devez spécifier l'ARN du groupe d'objets qui les contient. L'ARN d'un groupe d'objets présente le format suivant :

   ```
   arn:aws:iot:region:account-id:thinggroup/thing-group-name
   ```

   Pour attacher un profil de sécurité à tous les objets enregistrés dans un Compte AWS (ignorant les objets non enregistrés), vous devez spécifier un ARN avec le format suivant :

   ```
   arn:aws:iot:region:account-id:all/registered-things
   ```

   Pour attacher un profil de sécurité à tous les objets non enregistrés, vous devez spécifier un ARN au format suivant :

   ```
   arn:aws:iot:region:account-id:all/unregistered-things
   ```

   Pour attacher un profil de sécurité à tous les appareils, vous devez spécifier un ARN au format suivant :

   ```
   arn:aws:iot:region:account-id:all/things
   ```

6. Vous pouvez également suivre les violations avec l'outilListActiveViolations, qui vous permet d'identifier les violations détectées pour un profil de sécurité ou un appareil cible donné.

   Utilisation de l'ListViolationEventsPour voir les violations détectées pendant une période donnée. Vous pouvez filtrer ces résultats par profil de sécurité ou par appareil.

7. Si vos appareils violent trop souvent ou trop rarement les comportements définis, vous pouvez peaufiner la définition de ces comportements.
8. Pour consulter les profils de sécurité que vous configurez et les appareils surveillés, utilisez laListSecurityProfiles,ListSecurityProfilesForTarget, etListTargetsForSecurityProfileActions.

   Utilisation de l'DescribeSecurityProfilePour obtenir plus de détails sur un profil de sécurité.
9. Pour mettre à jour un profil de sécurité, utilisez l'UpdateSecurityProfileaction. Utilisation de l'DetachSecurityProfilepour détacher un profil de sécurité d'un compte ou d'un groupe d'objets cible. Utilisation de l'DeleteSecurityProfileAction pour supprimer entièrement un profil de sécurité.

Actions d'atténuation

Vous pouvez utiliserAWS IoT Device DefenderPour prendre des mesures pour atténuer les problèmes détectés lors d'une constatation d'audit ou d'une alarme Détecter.

> Note
>
> Les mesures d'atténuation ne seront pas appliquées sur les constatations d'audit supprimées. Pour plus d'informations sur les suppressions de résultats d'audit, consultezSuppressions des résultats d'audit (p. 911).

Actions d'atténuation d'audit

AWS IoT Device Defenderfournit des actions prédéfinies pour les différentes vérifications d'audit. Vous configurez ces actions pour votre Compte AWS Pour ensuite les appliquer à un ensemble de résultats. Ces résultats peuvent être :

- Tous les résultats d'un audit. Cette option est disponible dans la console AWS IoT console et à l'aide de l'interface de ligne de commande AWS CLI.
- Une liste des résultats individuels. Cette option est uniquement disponible à l'aide de l'interface de ligne de commande AWS CLI.
- Un ensemble filtré de résultats à partir d'un audit.

Le tableau suivant répertorie les types de contrôles d'audit et les actions d'atténuation pris en charge pour chacun :

Contrôle d'audit pour cartographie d'actions d'atténuation

Contrôle d'audit	Actions d'atténuation prises en charge
REVOKED_CA_CERT_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
DEVICE_CERTIFICATE_SHARED_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
CA_CERT_APPROACHING_EXPIRATION_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE

Contrôle d'audit	Actions d'atténuation prises en charge
CONFLICTING_CLIENT_IDS_CHECK	PUBLISH_FINDING_TO_SNS
DEVICE_CERT_APPROACHING_EXPIRATION_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
REVOKED_DEVICE_CERT_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
LOGGING_DISABLED_CHECK	PUBLISH_FINDING_TO_SNS, ENABLE_IOT_LOGGING
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
CA_CERTIFICATE_KEY_QUALITY_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS
IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK	PUBLISH_FINDING_TO_SNS

Tous les contrôles d'audit prennent en charge la publication des résultats d'audit dans Amazon SNS, afin que vous puissiez effectuer des actions personnalisées pour répondre à la notification. Chaque type de contrôle d'audit peut prendre en charge d'autres actions d'atténuation :

REVOKED_CA_CERT_CHECK

- Modifiez l'état du certificat pour le marquer comme inactif dans AWS IoT.

DEVICE_CERTIFICATE_SHARED_CHECK

- Modifiez l'état du certificat de l'appareil pour le marquer comme inactif dans AWS IoT.
- Ajoutez les appareils qui utilisent ce certificat à un groupe d’objets.

UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- Aucune action supplémentaire prise en charge.

AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK

- Aucune action supplémentaire prise en charge.

IOT_POLICY_OVERLY_PERMISSIVE_CHECK

- Ajoutez une version de stratégie AWS IoT pour limiter les autorisations.

CA_CERT_APPROACHING_EXPIRATION_CHECK

- Modifiez l'état du certificat pour le marquer comme inactif dans AWS IoT.

CONFLICTING_CLIENT_IDS_CHECK

- Aucune action supplémentaire prise en charge.

DEVICE_CERT_APPROACHING_EXPIRATION_CHECK

- Modifiez l'état du certificat de l'appareil pour le marquer comme inactif dans AWS IoT.
- Ajoutez les appareils qui utilisent ce certificat à un groupe d’objets.

DEVICE_CERTIFICATE_KEY_QUALITY_CHECK

- Modifiez l'état du certificat de l'appareil pour le marquer comme inactif dans AWS IoT.
- Ajoutez les appareils qui utilisent ce certificat à un groupe d’objets.

CA_CERTIFICATE_KEY_QUALITY_CHECK

- Modifiez l'état du certificat pour le marquer comme inactif dans AWS IoT.

REVOKED_DEVICE_CERT_CHECK

- Modifiez l'état du certificat de l'appareil pour le marquer comme inactif dans AWS IoT.
- Ajoutez les appareils qui utilisent ce certificat à un groupe d’objets.

LOGGING_DISABLED_CHECK

- Activez la journalisation

AWS IoT Device Defenderprend en charge les types d'actions d'atténuation suivants à l'égard des résultats de l'audit :

Type d'action	Remarques
ADD_THINGS_TO_THING_GROUP	Vous spécifiez le groupe auquel vous souhaitez ajouter les appareils. Vous pouvez également spécifier si l'adhésion à un ou plusieurs groupes dynamiques doit être remplacée si cela risque de dépasser le nombre maximum de groupes auxquels l'objet peut appartenir.
ENABLE_IOT_LOGGING	Vous spécifiez le niveau de journalisation et le rôle avec les autorisations pour la journalisation. Vous ne pouvez pas spécifier un niveau de journalisation `DISABLED`.
PUBLISH_FINDING_TO_SNS	Vous spécifiez la rubrique dans laquelle le résultat doit être publiée.
REPLACE_DEFAULT_POLICY_VERSION	Vous spécifiez le nom du modèle. Remplace la version de stratégie avec une stratégie vide ou par défaut. Seule une valeur `BLANK_POLICY` est actuellement prise en charge.
UPDATE_CA_CERTIFICATE	Vous spécifiez le nouvel état pour le certificat CA. Seule une valeur `DEACTIVATE` est actuellement prise en charge.
UPDATE_DEVICE_CERTIFICATE	Vous spécifiez le nouvel état pour le certificat d'appareil. Seule une valeur `DEACTIVATE` est actuellement prise en charge.

En configurant des actions standard lorsque des problèmes sont trouvés lors d'un audit, vous pouvez les résoudre de manière cohérente. L’utilisation de ces actions d'atténuation définies vous aide également à résoudre les problèmes plus rapidement et avec des risques réduits d'erreur humaine.

> **Important**
>
> L’application d’actions d'atténuation qui modifient des certificats, ajoutent des objets à un nouveau groupe d’objets ou remplacent la stratégie peut avoir un impact sur vos appareils et applications. Par exemple, les appareils peuvent s’avérer incapalbes de se connecter. Avant de les appliquer, prenez en compte les implications des actions d’atténuation. Vous devrez peut-être effectuer d'autres actions pour corriger les problèmes avant que vos appareils et applications fonctionnent normalement. Par exemple, il se peut que vous deviez fournir des certificats de l'appareil mis à jour. Les actions d’atténuation peuvent vous aider à limiter rapidement vos risques, mais vous devez tout de même prendre des mesures correctives pour résoudre les problèmes sous-jacents.

Certaines actions, comme la réactivation d'un certificat d'appareil, peuvent uniquement être effectuées manuellement. AWS IoT Device Defender ne fournit pas un mécanisme pour restaurer automatiquement les actions d'atténuation qui ont été appliqués.

Détection d'actions d'atténuation

AWS IoT Device Defenderprend en charge les types d'actions d'atténuation suivants sur Detect alarmes :

Type d'action	Remarques
ADD_THINGS_TO_THING_GROUP	Vous spécifiez le groupe auquel vous souhaitez ajouter les appareils. Vous pouvez également spécifier si l'adhésion à un ou plusieurs groupes dynamiques doit être remplacée si cela risque de dépasser le nombre maximum de groupes auxquels l'objet peut appartenir.

Comment définir et gérer des actions d’atténuation

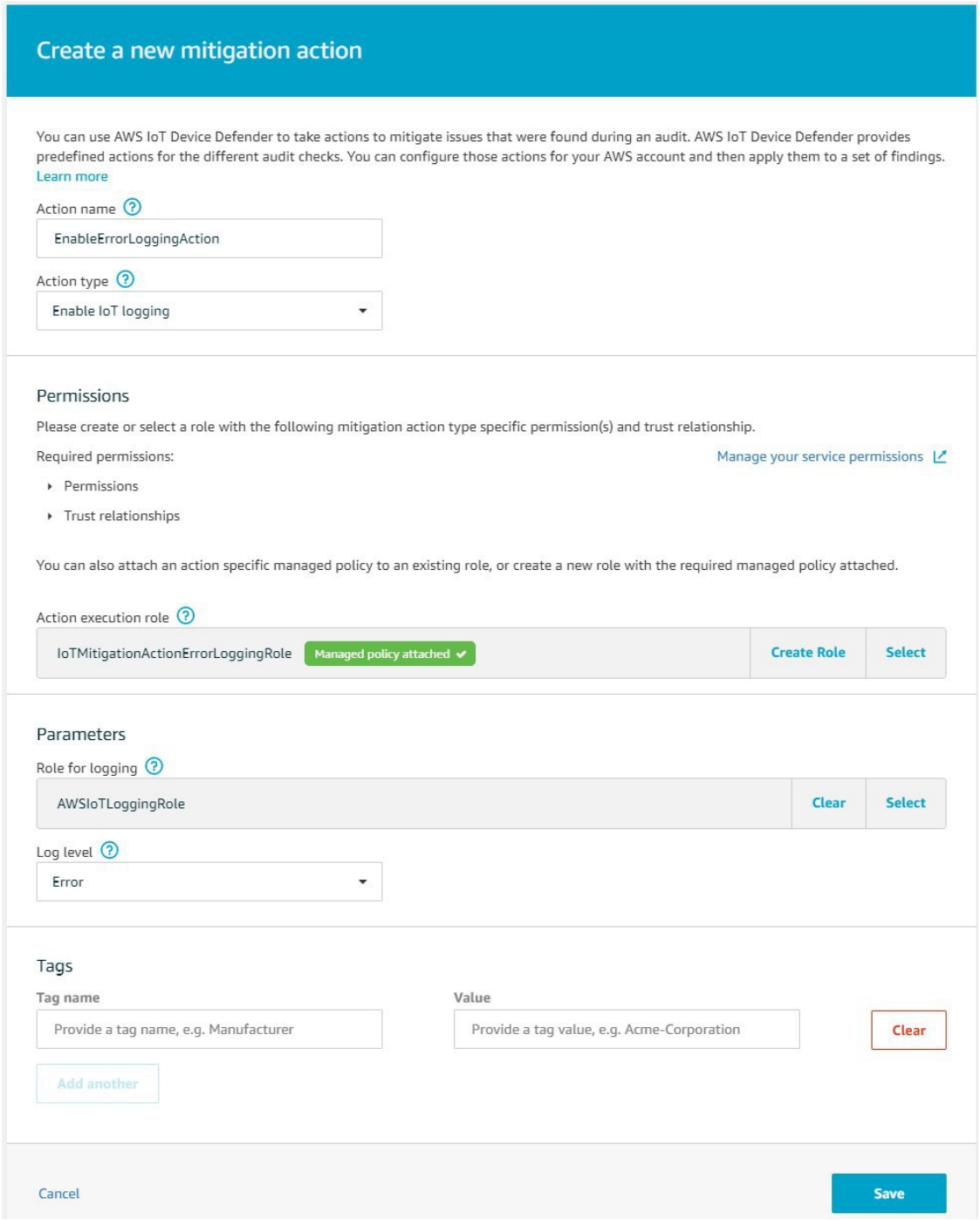
Vous pouvez utiliser la stratégieAWS IoTou la consoleAWS CLIPour définir et gérer des actions d'atténuation pour votre Compte AWS .

Créez des actions d’atténuation

Chaque action d'atténuation que vous définissez est une combinaison d'un type d'action prédéfinie et des paramètres spécifiques à votre compte.

Utiliser la console AWS IoT pour créer des actions d'atténuation

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Mitigation Actions (Actions d’atténuation).
3. Dans la page Mitigation Actions (Actions d'atténuation), choisissez Create (Créer).



4. Dans la page Create a Mitigation Action (Créer une action d'atténuation), dans Action name (Nom de l'action), saisissez un nom unique pour votre action d'atténuation.
5. Dans Action Type (Type d'action), spécifiez le type d'action que vous souhaitez définir.
6. Chaque type d'action demande un ensemble différent de paramètres. Saisissez les paramètres pour l'action. Par exemple, si vous choisissez le type d'action Add things to tring group(Ajouter des objets au groupe d'objets), choisissez le groupe de destination et sélectionnez ou désélectionnez Override dynamic groups (Remplacer groupes dynamiques).

7. Dans Action execution role (Rôle d’exécution d’action), choisissez le rôle sous les autorisations duquel l’action est appliquée.
8. Choisissez Save (Enregistrer) pour enregistrer votre action d'atténuation pour votre compte AWS.

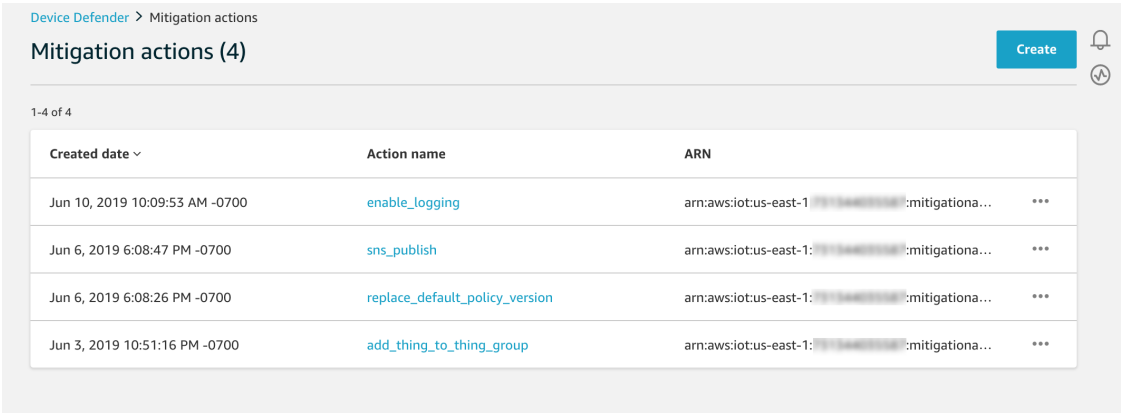
Utiliser l’interface de ligne de commance AWS CLI pour créer des actions d'atténuation

- Utilisation de l'CreateMitigationActionpour créer votre action d'atténuation. Le nom unique que vous attribuez à l'action est utilisé lorsque vous appliquez cette action aux résultats d’audit. Choisissez un nom descriptif.

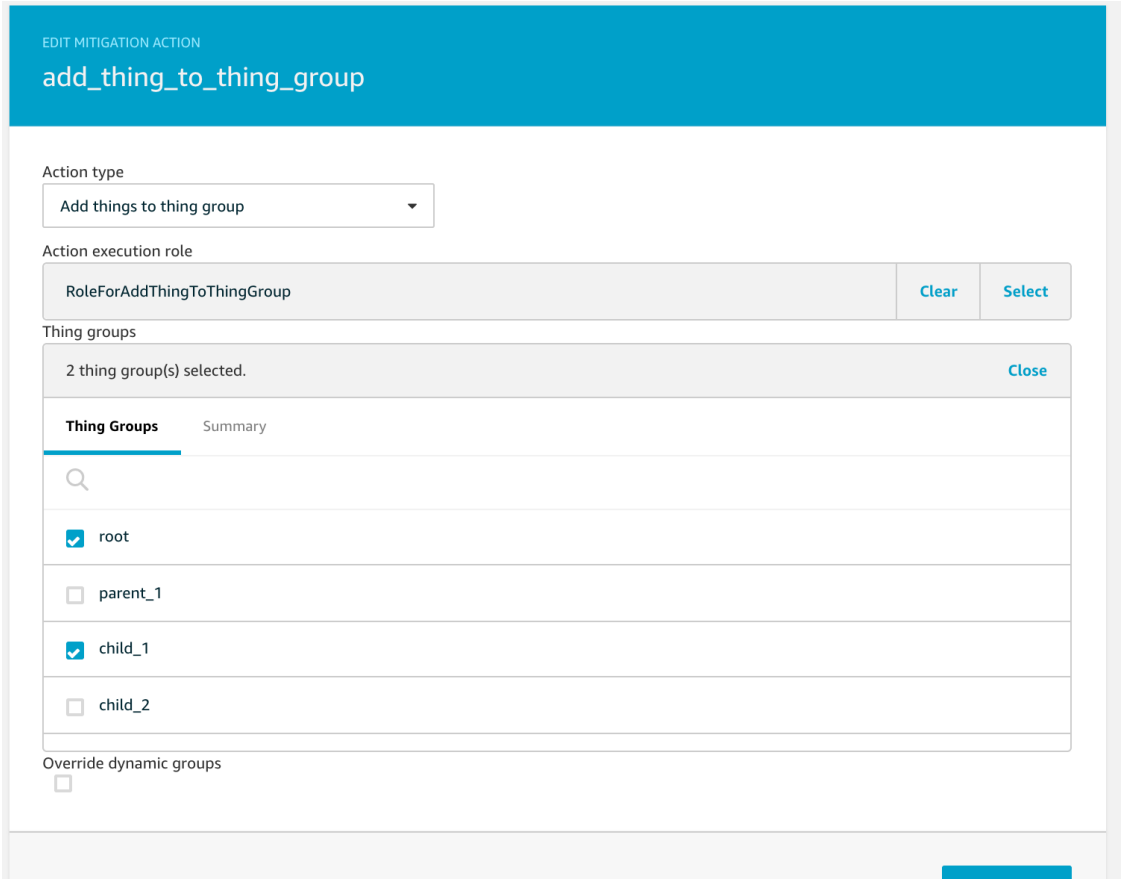
Utiliser la console AWS IoT pour afficher et modifier les actions d'atténuation

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Mitigation Actions (Actions d’atténuation).

   La .Actions d'atténuationaffiche une liste de toutes les actions d'atténuation qui sont définies pour votre Compte AWS .



3. Choisissez le lien du nom de l'action d'atténuation que vous voulez modifier.
4. Apportez vos modifications à l'action d'atténuation. Étant donné que le nom de l'action d'atténuation est utilisée pour l’identifier, vous ne pouvez pas modifier le nom.



5. ChoisissezEnregistrerPour enregistrer les modifications apportées à l'action d'atténuation dans votre Compte AWS .

Pour utiliser l'interface de ligne de commande AWS CLI pour répertorier une action d'atténuation

- Utilisation de l'ListMitigationActionpour répertorier vos actions d'atténuation. Si vous souhaitez modifier ou supprimer une action d'atténuation, notez le nom.

Pour utiliser l'AWS CLI pour mettre à jour une action d'atténuation

- Utilisation de l'UpdateMitigationActionpour modifier votre action d'atténuation.

Pour utiliser la console AWS IoT pour supprimer une action d'atténuation

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Mitigation Actions (Actions d'atténuation).

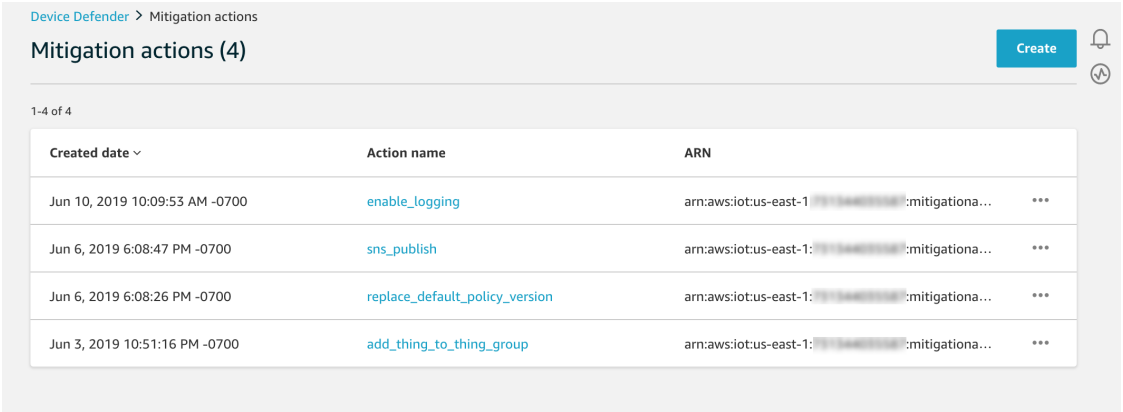
   La .Actions d'atténuationaffiche toutes les actions d'atténuation qui sont définies pour votre Compte AWS .
3. Choisissez les points de suspension (...) pour supprimer l'action d'atténuation voulue, puis choisissez Delete (Supprimer).

Utiliser l’interface de ligne de commande AWS CLI pour supprimer des actions d'atténuation

- Utilisation de l'UpdateMitigationActionpour modifier votre action d'atténuation.

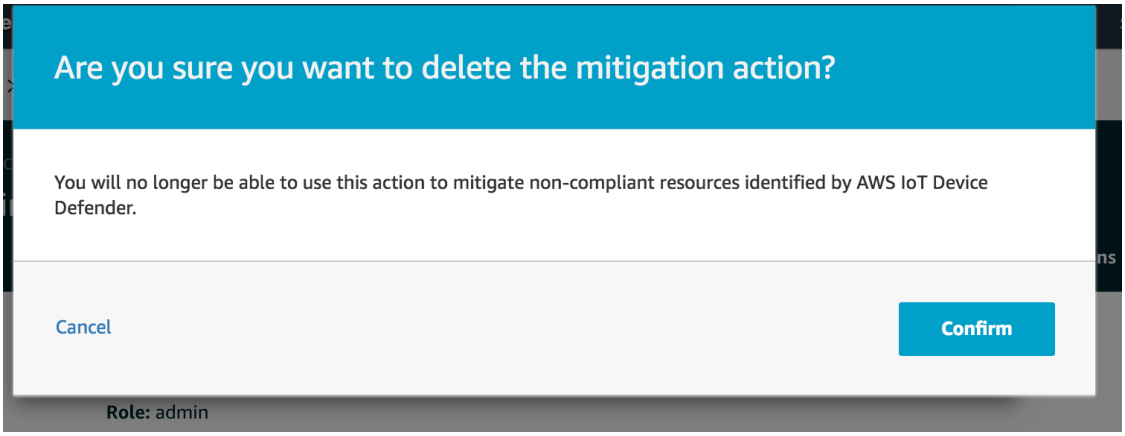
Utiliser la console AWS IoT pour afficher les détails d'une action d'atténuation

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Mitigation Actions (Actions d’atténuation).



La .Actions d'atténuationaffiche toutes les actions d'atténuation qui sont définies pour votre Compte AWS .

3. Choisissez le lien du nom de l'action d'atténuation que vous voulez modifier.
4. Dans la fenêtre Are you sure you want to delete the mitigation action (Êtes-vous sûr de vouloir supprimer l’action d’atténuation ?), choisissez Confirm (Confirmer).



Utiliser l’interface de ligne de commande AWS CLI pour afficher les détails de l'action d'atténuation

- Utilisation de l'DescribeMitigationActionPour afficher les détails de votre action d'atténuation.

Appliquer des actions d’atténuation

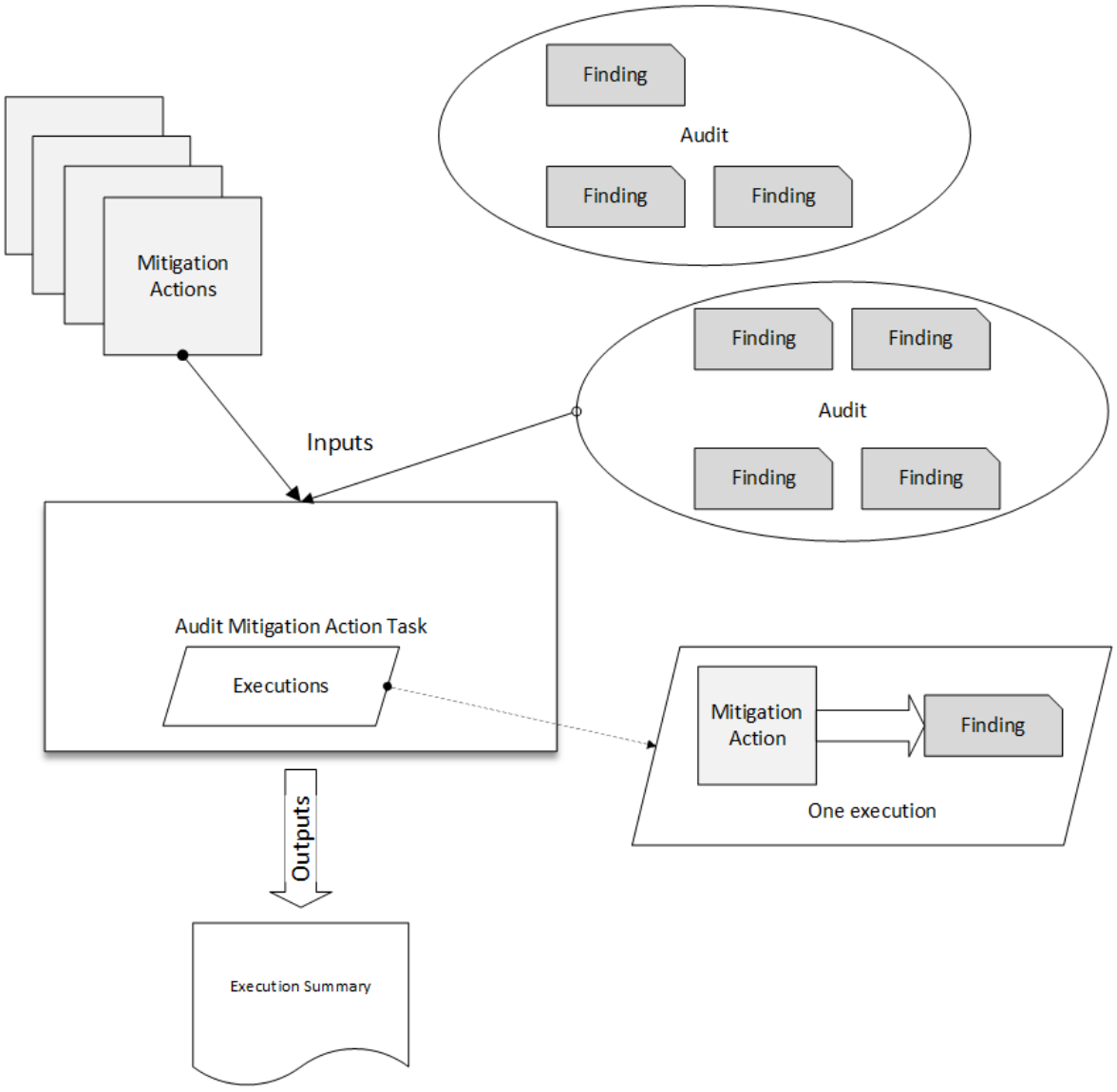
Une fois que vous avez défini un ensemble d’actions d'atténuation, vous pouvez les appliquer aux résultats d'un audit. Lorsque vous appliquez des actions, vous lancez une tâche d’actions d'atténuation d’audit. Cette tâche peut prendre un certain temps, en fonction de l'ensemble de résultats et des actions que vous leur

appliquez. Par exemple, si vous avez un grand groupe d'appareils dont les certificats ont expiré, cela peut prendre un certain temps de désactiver l'ensemble de ces certificats ou de déplacer ces appareils vers un groupe de quarantaine. D'autres actions, telles que l'activation de la journalisation, peuvent se réaliser rapidement.

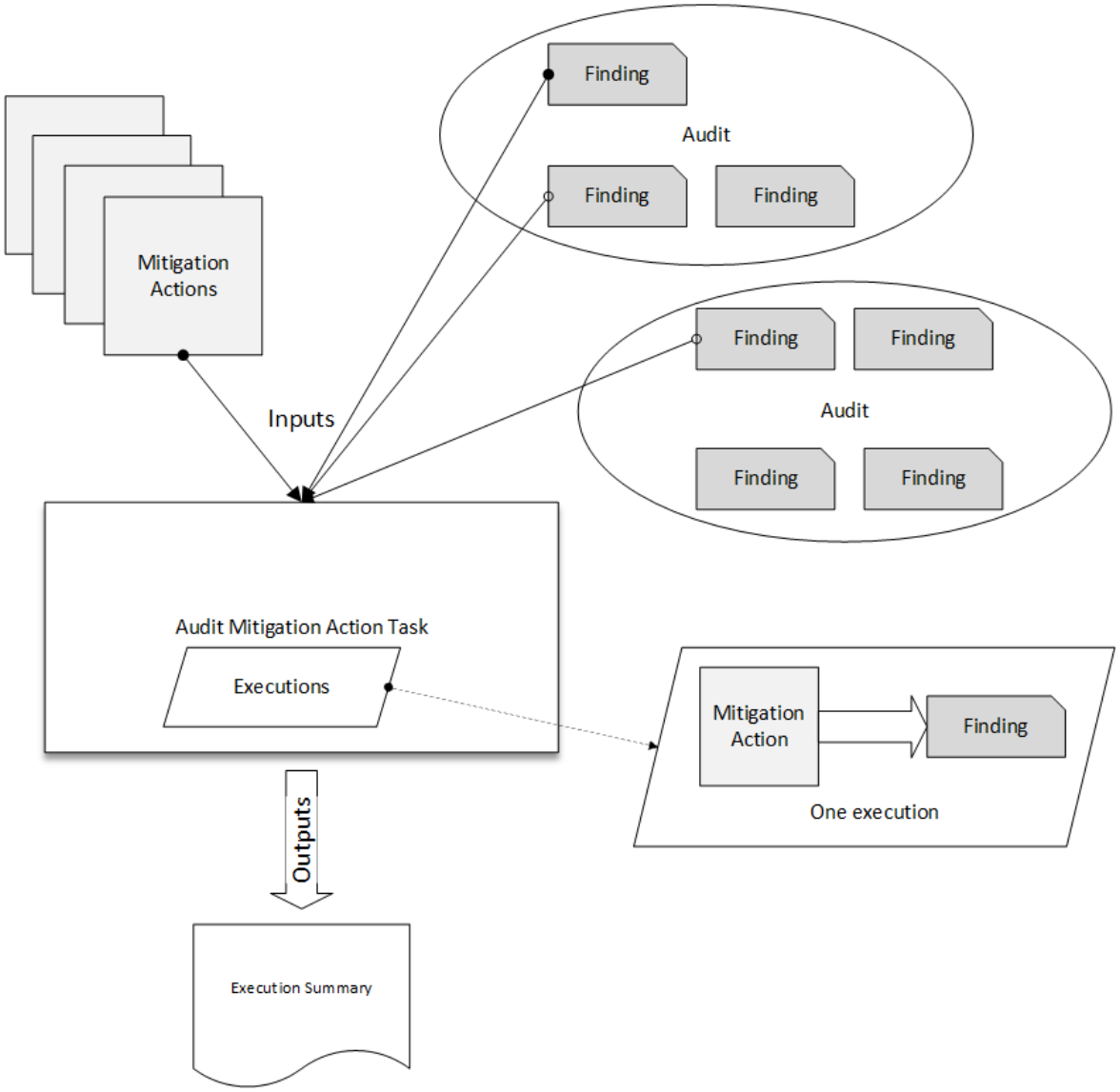
Vous pouvez afficher la liste des exécutions d'actions et annuler une exécution qui n'est pas encore terminée. Les actions déjà effectuée dans le cadre de l'exécution de l'action annulée ne sont pas restaurées. Si vous appliquez plusieurs actions à un ensemble de résultats et l'une de ces actions a échoué, les actions suivantes sont ignorées pour ce résultat (mais sont néanmoins appliquées à d'autres résultats). L'état de la tâche pour le résultat est FAILED. Le `taskStatus` est défini comme ayant échoué, si une ou plusieurs des actions ont échoué lors de l'application aux résultats. Les actions sont appliquées dans l'ordre dans lequel elles sont spécifiées.

Chaque action d'exécution applique un ensemble d'actions à une cible. Cette cible peut être une liste de résultats ou tous les résultats d'un audit.

Le schéma suivant montre comment vous pouvez définir une tâche d'atténuation d'audit qui accepte tous les résultats d'un audit et leur applique un ensemble d'actions. Une seule exécution applique une action à un résultat. La tâche d'actions d'atténuation d'audit génère un résumé d'exécution.



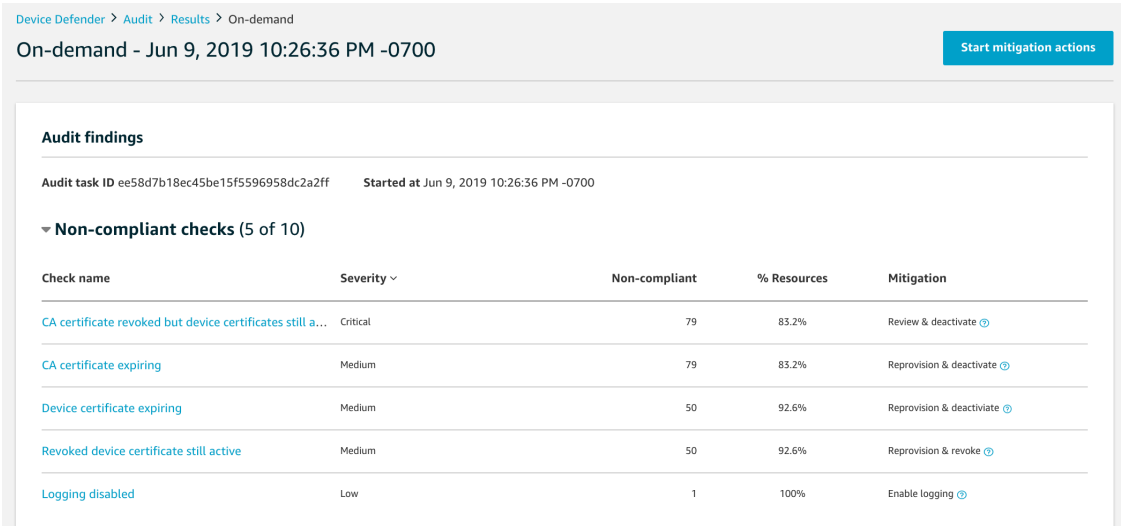
Le schéma suivant montre comment vous pouvez définir une tâche d'atténuation d'audit qui accepte une liste de résultats individuels à partir d'un ou plusieurs audits et applique un ensemble d'actions à ces résultats. Une seule exécution applique une action à un résultat. La tâche d'actions d'atténuation d'audit génère un résumé d'exécution.



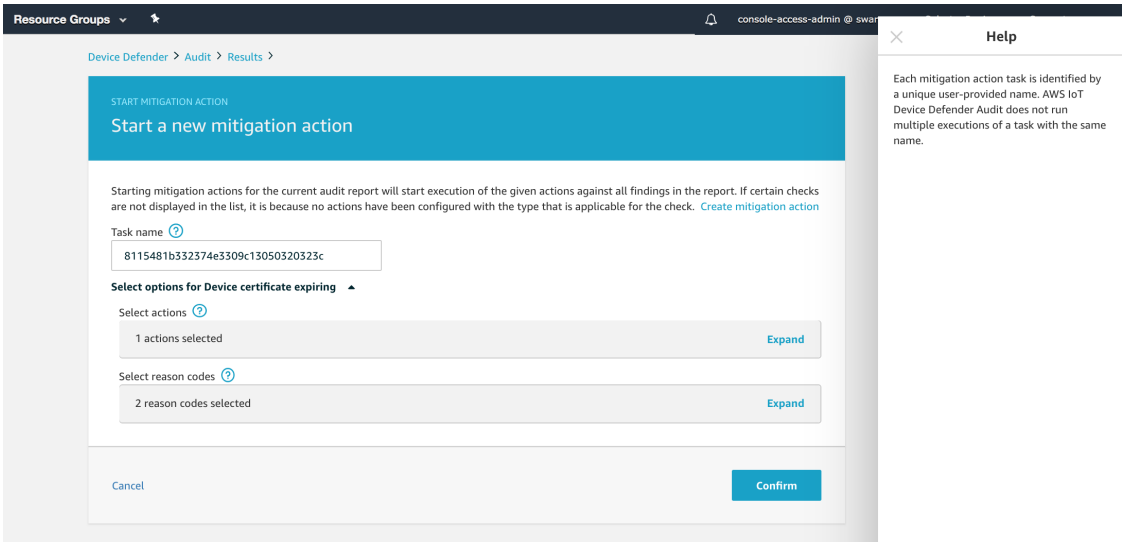
Vous pouvez utiliser la console AWS IoT console ou l'interface de ligne de commande AWS CLI pour appliquer des actions d'atténuation.

Utiliser la console AWS IoT pour appliquer des actions d'atténuation en lançant une action d'exécution

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Audit, puis Results (Résultats).



3. Choisissez le nom pour l'audit auquel vous souhaitez appliquer des actions.



4. Choisissez Start Mitigation Actions (Lancer des actions d'atténuation). Ce bouton n'est pas disponible si toutes vos vérifications sont conformes.

5. Dans Are you sure you want to start mitigation action task (Êtes-vous sûr de vouloir lancer la tâche d'actions d'atténuation ?), le nom de la tâche est par défaut le nom de l'ID d'audit, mais vous pouvez le changer par quelque chose de plus descriptif.
6. Pour chaque type de contrôle qui présente un ou plusieurs résultats non conformes dans l'audit, vous pouvez choisir une ou plusieurs actions à appliquer. Seules les actions qui sont valables pour le type de vérification sont affichées.

   Note

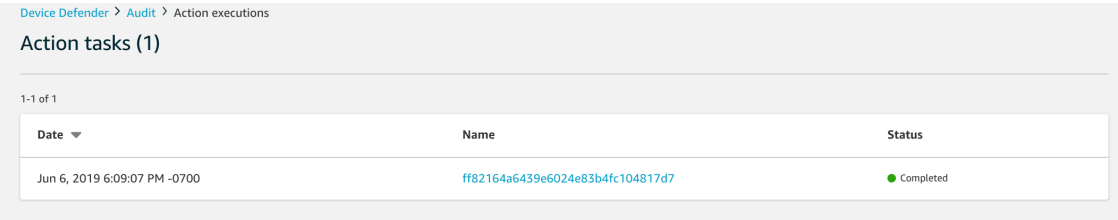
   Si vous n'avez pas configuré d'actions pour votre Compte AWS , la liste des actions disponibles est vide. Vous pouvez choisir le lien click here (cliquer ici) pour créer une ou plusieurs actions d'atténuation.
7. Lorsque vous avez indiqué toutes les actions que vous souhaitez appliquer, choisissez Confirm (Confirmer).

Utiliser l’interface de ligne de commande AWS CLI pour appliquer des actions d'atténuation en lançant une exécution d’actions d'atténuation d'audit

1. Si vous souhaitez appliquer des actions à toutes les conclusions de l'audit, utilisez laListAuditTaskspour trouver l'ID de tâche.
2. Si vous souhaitez appliquer uniquement des actions à des résultats sélectionnés, utilisez laListAuditFindingspour obtenir les ID de recherche.
3. Utilisation de l'ListMitigationActionset notez les noms des actions d'atténuation que vous souhaitez appliquer.
4. Utilisation de l'StartAuditMitigationActionsTaskpour appliquer des actions à la cible. Prenez note de l'ID de la tâche. Vous pouvez utiliser l'ID pour vérifier l'état de l'exécution de l’action, consulter les détails ou l’annuler.

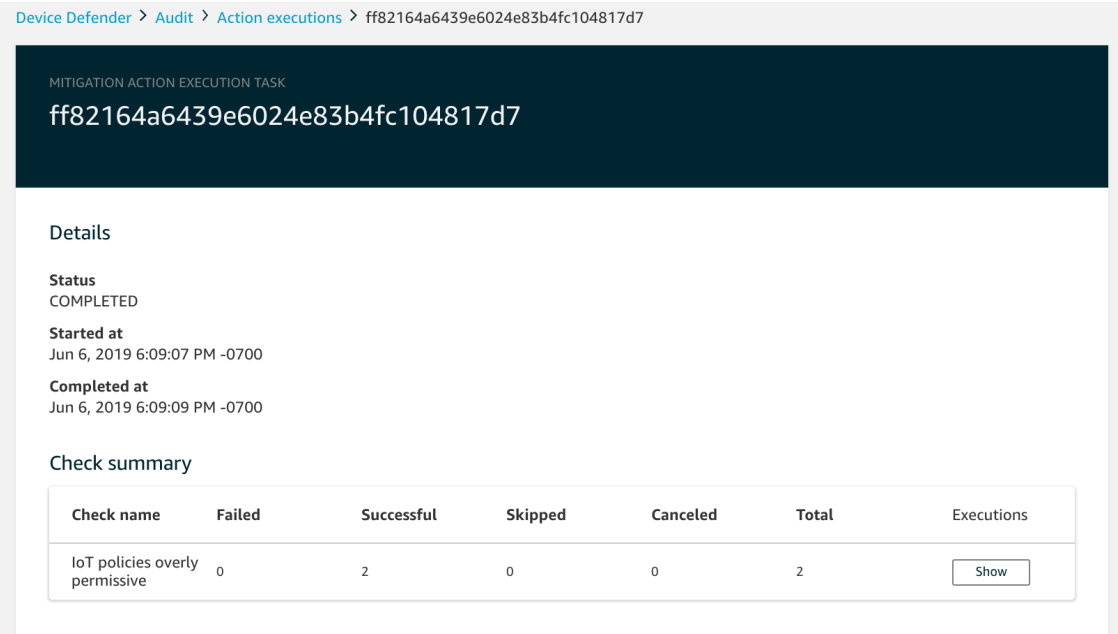
Utiliser la console AWS IoT pour voir vos exécutions d'actions

1. Ouvrez la console AWS IoT.
2. Dans le volet de navigation de gauche, choisissez Defend (Défendre), puis Action Executions (Exécutions d’actions).



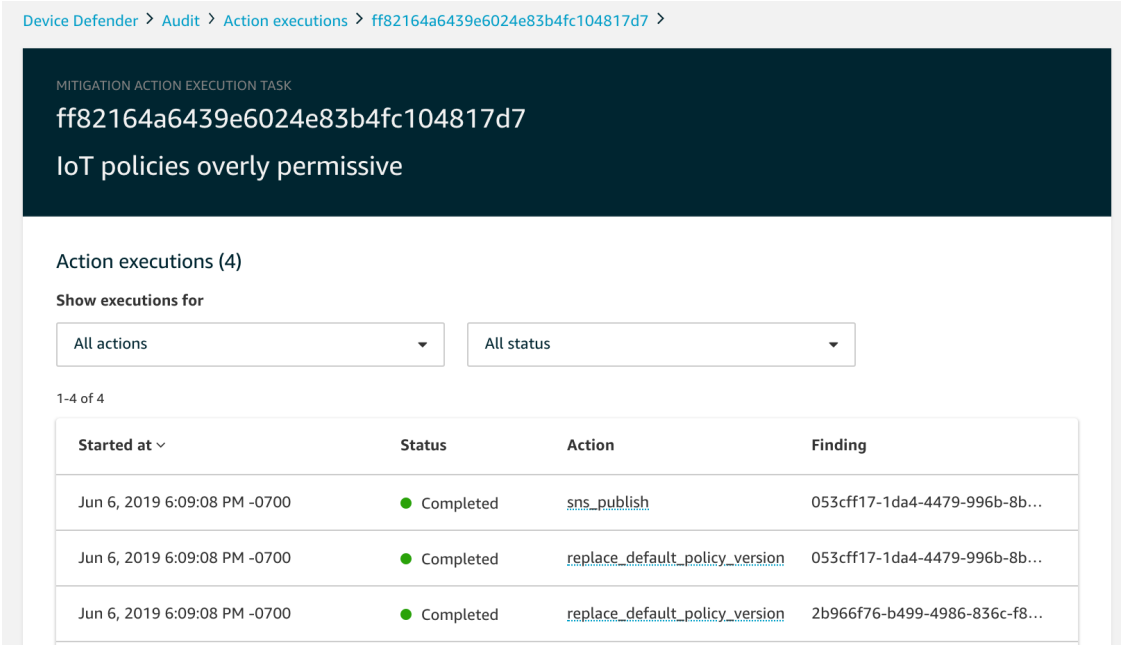
Une liste des tâches d'action indique le moment où chacune a été lancée et l’état actuel.

3. Choisissez le lien Name (Nom) pour voir les détails de la tâche. Les détails comprennent toutes les actions qui sont appliquées par la tâche, leur cible et leur état.



Vous pouvez utiliser les filtres Show executions for (Afficher les exécutions pour) pour vous concentrer sur les types d'actions ou les états d'action.

4. Pour afficher les détails de la tâche, dans Executions (Exécutions), choisissez Afficher.



Utiliser l’interface de ligne de commande AWS CLI pour répertorier vos tâches lancées

1. UtiliserListAuditMitigationActionsTasksPour afficher vos tâches d'actions d'atténuation d'audit. Vous pouvez fournir des filtres pour affiner les résultats. Si vous souhaitez afficher les détails de la tâche, notez l'ID de la tâche.
2. UtiliserListAuditMitigationActionsExecutionsPour afficher les détails d'exécution d'une tâche d'actions d'atténuation d'audit particulière.
3. UtiliserDescribeAuditMitigationActionsTaskPour afficher des détails sur la tâche, tels que les paramètres spécifiés lorsqu'elle a été lancée.

Utiliser l'interface de ligne de commandeAWS CLI pour annuler une tâche d’actions d’atténuation d’audit

1. Utilisation de l'ListAuditMitigationActionsTasksPour trouver l'ID de tâche pour la tâche dont vous souhaitez annuler l'exécution. Vous pouvez fournir des filtres pour affiner les résultats.
2. Utilisation de l'ListDetectMitigationActionsExecutionsÀ l'aide de l'ID de tâche, pour annuler votre tâche d'actions d'atténuation d'audit. Vous ne pouvez pas annuler des tâches qui ont été terminées. Lorsque vous annulez une tâche, les actions restantes ne sont pas appliquées, mais les actions d’atténuation déjà appliquées ne sont pas restaurées.

Permissions

Pour chaque action d'atténuation que vous définissez, vous devez fournir le rôle utilisé pour appliquer cette action.

Autorisations pour les actions d'atténuation

Type d'action	Modèle de stratégie d'autorisations	

UPDATE_DEVICE_CERTIFICATE	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:UpdateCertificate"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

UPDATE_CA_CERTIFICATE	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:UpdateCACertificate"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

ADD_THINGS_TO_THING_GROUP	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"iot:ListPrincipalThings",
	"iot:AddThingToThingGroup"
	],
	"Resource":[
	"*"
	]
	}
	]
	}	

Type d'action	Modèle de stratégie d'autorisations	
REPLACE_DEFAULT_POLICY_VERSION	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "iot:CreatePolicyVersion" ], "Resource":[ "*" ] } ] }`	
ENABLE_IOT_LOGGING	`{ "Version":"2012-10-17", "Statement":[ { "Effect":"Allow", "Action":[ "iot:SetV2LoggingOptions" ], "Resource":[ "*" ] }, { "Effect":"Allow", "Action":[ "iam:PassRole" ], "Resource":[ "<IAM role ARN used for setting up logging>" ] } ] }`	

Type d'action	Modèle de stratégie d'autorisations	

PUBLISH_FINDING_TO_SNS	{
	"Version":"2012-10-17",
	"Statement":[
	{
	"Effect":"Allow",
	"Action":[
	"sns:Publish"
	],
	"Resource":[
	"<The SNS topic to which the finding is published>"
	]
	}
	]
	}	

Pour tous les types d'action d'atténuation, utilisez le modèle de stratégie d'approbation suivant :

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

Commandes d'action d'atténuation

Vous pouvez utiliser ces commandes d'action d'atténuation pour définir un ensemble d'actions pour votre Compte AWS Vous pouvez ensuite appliquer à un ou plusieurs ensembles de résultats d'audit. Il existe trois catégories de commande :

- Celles utilisées pour définir et gérer des actions.
- Celles utilisées pour démarrer et gérer l'application de ces actions pour les résultats d'audit.
- Celles utilisées pour démarrer et gérer l'application de ces actions pour détecter les alarmes.

Commandes d'action d'atténuation

Définir et gérer des actions	Lancer et gérer l'exécution d'audit	Lancer et gérer l'exécution Détecter
CreateMitigationAction	CancelAuditMitigationActionsTask	CancelDetectMitigationActionStask

Définir et gérer des actions	Lancer et gérer l'exécution d'audit	Lancer et gérer l'exécution Détecter
DeleteMitigationAction	DescribeAuditMitigationActionsTask	DescribeDetectMitigationActionsTask
DescribeMitigationAction	ListAuditMitigationActionsTasks	ListDetectMitigationActionStasks
ListMitigationActions	StartAuditMitigationActionsTask	DémarrerDetectMitigationActionStask
UpdateMitigationAction	ListAuditMitigationActionsExecutions	ListDetectMitigationActionsExecutions

Intégration de l'appareil àAWS IoT Greengrass

AWS IoT Device Defender peut être utilisé avec AWS IoT Greengrass. L'intégration d'un agent d'appareil suit le modèle de déploiement de la fonction Lambda AWS IoT Greengrass standard, ce qui vous permet d'ajouter une sécurité AWS IoT Device Defender à vos appareils de noyau AWS IoT Greengrass. Procédez comme suit pour intégrer un agent d'appareil.

Prérequis :

- Configuration de votre environnement AWS IoT Greengrass
- Configurez et exécutez votre noyau AWS IoT Greengrass.
- Assurez-vous de pouvoir déployer et exécuter une fonction Lambda sur votre noyau AWS IoT Greengrass.

En général, le processus décrit ici suit la section Création et empaquetage d'une fonction Lambda du Manuel du développeur AWS IoT Greengrass.

Pour créer un package Lambda

1. Clonez le référentiel d'échantillons Python AWS IoT Device Defender.

```
git clone https://github.com/aws-samples/aws-iot-device-defender-agent-sdk-python.git
```

2. Créez et activez un environnement virtuel (facultatif, mais conseillé).

```
pip install virtualenv
virtualenv metrics_lambda_environment
source metrics_lambda_environment/bin/activate
```

3. Installez l'exemple d'agent AWS IoT Device Defender dans l'environnement virtuel. Installez depuis PyPi.

```
pip install AWSIoTDeviceDefenderAgentSDK
```

4. Installez la source téléchargée.

```
cd aws-iot-device-defender-agent-sdk-python
#This must be run from the same directory as setup.py
pip install .
```

5. Créez un répertoire vide pour assembler votre fonction Lambda. Il s'agit de votre répertoire Lambda.

```
mkdir metrics_lambda
cd metrics_lambda
```

6. Effectuez les étapes 1 à 4 de la section Création et empaquetage d'une fonction Lambda.
7. Décompressez le kit de développement SDK Python AWS IoT Greengrass dans votre répertoire Lambda.

```
unzip ../aws_greengrass_core_sdk/sdk/python_sdk_1_1_0.zip
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrass_common .
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrasssdk .
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrass_ipc_python_sdk .
```

8. Copiez le module `AWSIoTDeviceDefenderAgentSDK` à la racine de votre répertoire Lambda.

```
cp -R ../aws-iot-device-defender-agent-sdk-python/AWSIoTDeviceDefenderAgentSDK .
```

9. Copiez l'agent AWS IoT Greengrass à la racine de votre répertoire Lambda.

```
cp ../aws-iot-device-defender-agent-sdk-python/samples/greengrass/
greengrass_core_metrics_agent/greengrass_defender_agent.py .
```

10. Personnalisez l'agent AWS IoT Greengrass de manière à inclure le nom de votre appareil noyau AWS IoT Greengrass et l'intervalle de rapport de métriques souhaité :

    - Remplacez `GREENGRASS_CORENAME` par le nom de votre noyau AWS IoT Greengrass.
    - Définissez `SAMPLE_RATE_SECONDS` sur votre intervalle de rapport de métriques souhaité. L'intervalle de rapport le plus court pris en charge par AWS IoT Device Defender est de 5 minutes (300 secondes).

11. Copiez les dépendances de votre environnement virtuel (ou votre système) dans la racine de votre répertoire Lambda.

```
cp -R ../metrics_lambda_environment/lib/python2.7/site-packages/psutil .
cp -R ../metrics_lambda_environment/lib/python2.7/site-packages/cbor .
```

12. Créez votre fichier zip de fonction Lambda. Exécutez cette commande à la racine de votre répertoire Lambda.

```
rm *.zip
zip -r greengrass_defender_metrics_lambda.zip *
```

Pour configurer et déployer votreAWS IoT GreengrassFonction Lambda

1. Chargez votre fichier zip Lambda.
2. Sélectionnez l'environnement d'exécution Python 2.7, et dans le champ Handler (Gestionnaire), entrez `greengrass_defender_agent.function_handler`.
3. Configurez votre fonction Lambda sous la forme d'une fonction Lambda à longue durée de vie.
4. Configurez un abonnement de votre fonction Lambda versAWS IoTcloud. PourAWS IoT Device Defender, un abonnement de laAWSLe cloud vers votre fonction Lambda n'est pas requis.
5. Créez une ressource locale pour permettre à votre fonction Lambda de collecter des métriques à partir de votre hôte AWS IoT Greengrass :

    - Suivez les instructions fournies dans la section Accès aux ressources locales avec les fonctions Lambda. Utilisez les paramètres suivants :
        - Nom de la ressource : `Core Proc`
        - Type: `Volume`
        - Chemin source : `/proc`
        - Chemin de la destination : `/host_proc`

- Autorisation d'accès fichier pour le propriétaire du groupe : Ajouter automatiquement les autorisations de groupe de système d'exploitation du groupe Linux qui possède la ressource
- Associez la ressource à votre fonction Lambda de métriques.

6. Déployez votre fonction Lambda sur votre groupe AWS IoT Greengrass.

Pour vérifier votreAWS IoT Device Defendermesures de périphérique à l'aide de laAWS IoTconsole

1. Remplacez temporairement la rubrique de publication dans votre fonction Lambda AWS IoT Greengrass par « métriques/test ».
2. Déployez la fonction Lambda.
3. Pour voir les métriques émises par votre noyau AWS IoT Greengrass, sur la page Test de la console AWS IoT, ajoutez un abonnement à la rubrique temporaire (« métriques/test »).

Bonnes pratiques de sécurité pour les agents d'appareil

Principe de moindre privilège

Les autorisations minimum nécessaires doivent être accordées au processus d'agent pour qu'il exécute ses tâches.

Mécanismes de base

- L'agent doit être exécuté en tant qu'utilisateur non-racine.
- L'agent doit être exécuté en tant qu'utilisateur dédié dans son propre groupe.
- Les utilisateurs/groupes doivent disposer des autorisations en lecture seule sur les ressources nécessaires, afin de rassembler et de transmettre des métriques.
- Exemple : lecture seule activée/proc/sys pour l'exemple d'agent.
- Pour obtenir un exemple de la façon de configurer un processus à exécuter avec des autorisations réduites, consultez les instructions de configuration incluses avec l'exemple d'agent Python.

Il existe un certain nombre de mécanismes Linux connus qui peuvent vous aider à restreindre ou isoler davantage votre processus d'agent :

Mécanismes avancés

- CGroups
- SELinux
- Chroot
- Espaces de noms Linux

Résilience opérationnelle

Un processus d'agent doit résister aux exceptions et aux erreurs opérationnelles inattendues et ne doit pas planter ou se fermer en permanence. Le code doit gérer correctement les exceptions et, par précaution, être configuré pour redémarrer automatiquement en cas de mise hors service inattendue (par exemple, en cas de redémarrage du système ou d'exceptions non interceptées).

Principe de moindre dépendance

Un agent doit utiliser un nombre de dépendances minimum (c'est-à-dire des bibliothèques tierces) dans son implémentation. Si l'utilisation d'une bibliothèque est justifiée en raison de la complexité

d'une tâche (par exemple, le protocole TLS), utilisez uniquement les dépendances bien gérées et mettez en place un mécanisme pour les conserver à jour. Si les dépendances ajoutées contiennent des fonctionnalités non utilisées par l'agent et actives par défaut (par exemple, l'ouverture des ports, le socket de domaine), désactivez-les dans votre code ou via les fichiers de configuration de la bibliothèque.

Isolation du processus

Un processus d'agent doit uniquement contenir des fonctionnalités nécessaires pour rassembler et transmettre les métriques de l'appareil. Il ne doit pas s'intégrer à d'autres processus système en tant que conteneur ou implémenter des fonctionnalités pour d'autres cas d'utilisation hors de portée. En outre, le processus d'agent ne doit pas créer de canaux de communication entrants, tels que des sockets de domaine et des ports de service réseau, qui permettraient aux processus locaux ou distants d'influer sur son fonctionnement et d'impacter son intégrité et son isolement.

Furtivité

Un processus d'agent ne doit pas être nommé avec des mots-clés, tels que sécurité, surveillance ou audit qui indiquent son objectif et la valeur de la sécurité. Les noms de code génériques ou aléatoires ainsi que les noms de processus propres à chaque appareil sont plébiscités. Le même principe doit être suivi pour nommer le répertoire dans lequel résident les binaires de l'agent ainsi que les noms et les valeurs des arguments du processus.

Principe de moindre informations partagées

Les artefacts d'agent déployés vers des appareils ne doivent pas contenir d'informations sensibles, telles que des informations d'identification privilégiées, un code de débogage et un code mort, des fichiers de documentation ou des commentaires en ligne révélant des détails sur le traitement côté serveur des métriques rassemblées par l'agent ou d'autres détails sur les systèmes backend.

: acte de révision dans un pipeline se poursuivant d'une étape à l'autre dans un flux de travail.

Pour mettre en place des canaux sécurisés TLS pour la transmission des données, un processus d'agent doit appliquer toutes les validations côté client, telles que la chaîne de certificats et la validation du nom de domaine, au niveau de l'application si elles ne sont pas activées par défaut. En outre, un agent doit utiliser un magasin de certificats racines contenant des autorités de confiance, mais pas de certificats appartenant à des émetteurs de certificats compromis.

Déploiement sécurisé

Les mécanismes de déploiement d'agent, tels que la synchronisation ou la transmission de code ainsi que les référentiels contenant leurs binaires, les codes sources et les fichiers de configuration (y compris les certificats racines de confiance) doivent disposer d'un accès contrôlé pour empêcher l'injection ou la falsification de code non autorisé. Si le mécanisme de déploiement s'appuie sur la communication réseau, utilisez des méthodes cryptographiques pour protéger l'intégrité des artefacts de déploiement en transit.

Suggestions de lecture

- Sécurité dans AWS IoT (p. 231)
- Présentation desAWS IoTModèle de sécurité
- Redhat : Une morsure de Python
- 10 pièges de sécurité courants dans Python et comment les éviter
- En quoi consiste le principe de moindre privilège et pourquoi est-il nécessaire ?
- Top 10 de la sécurité embarquée OWASP
- Projet IoT OWASP

Device Advisor

Device Advisorest une fonctionnalité de test entièrement gérée basée sur le cloud pour valider les périphériques IoT pendant le développement de logiciels de périphériques. Device Advisor fournit des tests prédéfinis que vous pouvez utiliser pour valider les périphériques IoT pour une connectivité fiable et sécurisée avec AWS IoT Core , avant de déployer des périphériques en production. Les tests prédéfinis de Device Advisor vous aident à valider le logiciel de votre appareil par rapport aux meilleures pratiques d'utilisation deTLS,MQTT,Device Shadow, etEmplois IoT. Vous pouvez également télécharger des rapports de qualification signés à soumettre auAWSPartner Network pour que votre appareil soit qualifié pourAWSCatalogue des appareils partenairessans avoir à envoyer votre appareil et attendre qu'il soit testé.

> Note
>
> Device Advisor est pris en charge dans les régions us-east-1, us-west-1 et eu-west-1. Device Advisor prend en charge MQTT avec les certificats clients X509.

Ce chapitre contient les sections suivantes :

- Configuration de (p. 998)
- Démarrer avec Device Advisor dans la console (p. 1002)
- Device Advisor Flux (p. 1014)
- Workflow détaillé de la console Device (p. 1018)
- Device Advisor (p. 1036)

Tout appareil qui a été construit pour se connecter à AWS IoT Core peut tirer parti de Device Advisor. Vous pouvez accéder à Device Advisor à partir de laAWS IoTconsole, ou en utilisant la méthodeAWS CLIou kit SDK. Lorsque vous êtes prêt à tester votre appareil, enregistrez-le avec AWS IoT Core et configurez le logiciel du périphérique avec le point de terminaison Device Advisor. Choisissez ensuite les tests prédéfinis, configurez-les, exécutez les tests sur votre appareil et obtenez les résultats des tests avec des journaux détaillés ou un rapport de qualification.

Device Advisor est un point de terminaison de test dans leAWScloud. Vous pouvez tester vos périphériques en les configurant pour se connecter au point de terminaison de test fourni par Device Advisor. Une fois qu'un périphérique est configuré pour se connecter au point de terminaison de test, vous pouvez visiter la console du Device Advisor ou utiliser leAWSSDK pour choisir les tests que vous voulez exécuter sur vos appareils. Device Advisor gère ensuite le cycle de vie complet d'un test, y compris le provisionnement des ressources, la planification du processus de test, la gestion de la machine d'état, l'enregistrement du comportement du périphérique, l'enregistrement des résultats et la fourniture des résultats définitifs sous la forme d'un rapport de test.

Configuration de

Avant d'utiliser Device Advisor pour la première fois, effectuez les tâches suivantes.

Prérequis

- Créer un rôle IAM à utiliser comme rôle de périphérique (p. 999)
- Créer une stratégie gérée sur mesure pour votre compte d'utilisateur Device Advisor (p. 1000)
- Créer un utilisateur IAM à utiliser pour exécuter les tests Device Advisor (p. 1000)
- Création d'unAWS IoTchose et certificat (p. 1001)

- Configurer votre appareil de test (p. 1001)

Créer un rôle IAM à utiliser comme rôle de périphérique

Cette section vous montre comment créer un Compte AWS et ajoutez des autorisations à un utilisateur IAM que vous pouvez utiliser pour exécuter des tests Device Advisor sur vos appareils.

1. Accédez à laAWSConsole IAMet connectez-vous au compte que vous utilisez pour les tests Device Advisor.
2. Tout d'abord, créez une stratégie qui limite les autorisations de rôle aux stratégies de chose de périphérique. Dans le panneau de navigation de gauche, choisissezStratégies.
3. Choisissez Créer une stratégie.
4. UnderCréer une stratégie, procédez comme suit :
    1. ChoisissezIoTpourService.
    2. UnderAction, vérifiezToutes les actions IoT (IoT : *).
    3. UnderRessources, vous pouvez sélectionner toutes les ressources. Toutefois, pour les meilleures pratiques en matière de sécurité, limitezclient,topic, ettopicfilter. Si vous choisissez de restreindre ces ressources, suivez les étapes ci-dessous.
        a. ChoisissezSpécifier l'ARN de ressource client pour l'action Connect.
            i. Choisissez Add ARN (Ajouter un ARN).
            ii. Spécifiez laregion,accountId, etclientIdDans l'éditeur d'ARN visuel ou spécifiez manuellement l'Amazon Resource Names (ARN) des rubriques IoT que vous souhaitez utiliser pour exécuter des cas de test. La .clientIdest le MQTTclientIdvotre appareil utilise pour interagir avec Device Advisor.
            iii. Choisissez Add (Ajouter).
        b. ChoisissezSpécifier l'ARN de ressource de rubrique pour l'action Recevoir et 1 autre.
            i. Choisissez Add ARN (Ajouter un ARN).
            ii. Spécifiez laregion,accountId, etNom de la rubriquedans l'éditeur ARN visuel ou spécifiez manuellement les ARN des rubriques IoT que vous souhaitez utiliser pour exécuter des cas de test. La .Nom de la rubriqueest le MQTTtopicl'utilisation de votre appareil pour publier des messages sur.
            iii. Choisissez Add (Ajouter).
        c. ChoisissezSpécifier l'ARN de ressource topicfilter pour l'action S'abonner.
            i. Choisissez Add ARN (Ajouter un ARN).
            ii. Spécifiez laregion,accountId, etNom de la rubriquedans l'éditeur ARN visuel ou spécifiez manuellement les ARN des rubriques IoT que vous souhaitez utiliser pour exécuter des cas de test. La .Nom de la rubriqueest le MQTTtopicvotre appareil utilise pour vous abonner.
            iii. Choisissez Add (Ajouter).
5. Choisissez Review policy (Examiner une stratégie).
6. UnderExaminer la stratégie, saisissez unNom.
7. Choisissez Créer une stratégie.
8. Dans le panneau de navigation de gauche, choisissezRôles.
9. Choisissez Create Role (Créer le rôle).
10. UnderOu sélectionnez un service pour afficher ses cas d'utilisation, choisissezIoT.
11. Sous Select your use case (Sélectionner votre cas d'utilisation), choisissez IoT.
12. Choisissez Next (Suivant) Permissions (Autorisations).

13. (Facultatif)UnderDéfinir la limite des autorisations, choisissezUtiliser une limite d'autorisations pour contrôler les autorisations maximum de rôle, puis choisissez la stratégie que vous venez de créer.
14. Choisissez Next (Suivant) Tags (Balises).
15. Choisissez Next (Suivant) Review (Examiner).
16. Saisissez unNom de rôleet unDescription du rôle.
17. Sélectionnez Créer un rôle.
18. Accédez au rôle que vous avez créé.
19. ChoisissezAttacher des stratégiesdans leAutorisations, puis choisissez la stratégie que vous avez créée dansÉtape 4.
20. Choisissez Attacher la stratégie.
21. ChoisissezRelations d'approbationet choisissezModifier la relation d'approbation.
22. Entrez cette politique :

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

23. Choisissez Update Trust Policy.

Créer une stratégie gérée sur mesure pour votre compte d'utilisateur Device Advisor

1. Accédez à la console IAM à l'adressehttps://console.aws.amazon.com/iam/et connectez-vous à votre compte.
2. Dans le volet de navigation de gauche, choisissez Policies (Stratégies).
3. ChoisissezCréation d'une stratégie, puis.JSONOnglet.
4. Ajoutez les autorisations nécessaires pour utiliser Device Advisor. Le document de politique est disponible sousBonnes pratiques de sécurité.
5. Choisissez Review Policy (Examiner une stratégie).
6. Saisissez un Name (Nom) et une Description.
7. Choisissez Create Policy (Créer une stratégie).

Créer un utilisateur IAM à utiliser pour exécuter les tests Device Advisor

> Note
>
> Nous vous recommandons de créer un utilisateur IAM à utiliser lorsque vous exécutez des tests Device Advisor. Bien que nous ne le recommandons pas, vous pouvez également utiliser un utilisateur IAM Admin.

1. Accédez à la console IAM à l'adressehttps://console.aws.amazon.com/iam/et connectez-vous à votre compte.
2. Dans le panneau de navigation de gauche, choisissezUsers.
3. Sélectionnez Add User (Ajouter un utilisateur).
4. Saisissez unNom d'utilisateur.
5. Tâche de sélectionAccès programmatique.
6. Choisissez Next (Suivant) Permissions (Autorisations).
7. Choisissez Attacher directement les stratégies existantes.
8. Entrez le nom de la stratégie personnalisée que vous avez créée dans la zone de recherche, puis cochez la case située à gauche deNom de la stratégie.
9. Choisissez Next (Suivant) Tags (Balises).
10. Choisissez Next (Suivant) Review (Examiner).
11. Choisissez Create user (Créer un utilisateur).
12. Choisissez Close (Fermer).

Device Advisor a besoin d'accéder à votreAWS(objets, certificats, point de terminaison) en votre nom. Votre utilisateur IAM doit disposer des autorisations nécessaires. Device Advisor publiera également des journaux sur Amazon CloudWatch si vous associez la stratégie d'autorisations nécessaire à votre utilisateur IAM.

Création d'unAWS IoTchose et certificat

1. Accédez à la console AWS IoT Core . Connectez-vous au compte que vous utilisez pour le test Device Advisor, puis choisissezGérer.
2. Si vous avez des éléments existants, choisissezCréerpour créer une nouvelle chose. Dans le cas contraire, sur leVous n'avez pas encore d'objetsPage, choisissezEnregistrer un objet.
3. Sur la page Création d'objets AWS IoT, choisissez Créer un objet unique.
4. Dans la pageAjouter votre appareil au registre d'objets, entrez unNomPour votre objet. Choisissez Suivant.
5. Sur la page Add a certificate for your thing (Ajouter un certificat pour votre objet), choisissez Create certificate (Créer un certificat). Des notifications s'affichent confirmant que votre objet et un certificat pour votre objet sont créés.
6. Copiez le certificat que vous avez créé à l'étape précédente sur votre appareil. L'emplacement correct du certificat dépend des références au certificat dans le logiciel ou le firmware de votre appareil.

Configurer votre appareil de test

Device Advisor utilise l'extension TLS d'indication de nom de serveur (SNI) pour appliquer des configurations TLS. Les appareils doivent utiliser cette extension lors de la connexion et transmettre un nom de serveur identique au point de terminaison de test Device Advisor.

Device Advisor autorise la connexion TLS lorsque le test est en cours d'exécution et refuse la connexion TLS avant et après chaque exécution de test. Pour cette raison, nous vous recommandons également d'utiliser le mécanisme de nouvelle tentative de connexion de périphérique pour bénéficier d'une expérience de test entièrement automatisée avec Device Advisor. Si vous exécutez une suite de tests avec plus d'un cas de test, par exemple TLS connect, MQTT connect et MQTT publication, nous vous recommandons de disposer d'un mécanisme conçu pour que votre appareil essaie de se connecter à notre point de terminaison de test toutes les cinq secondes. Vous pouvez ensuite exécuter plusieurs cas de test en séquence de manière automatisée.

Note

Pour que le logiciel de votre appareil soit prêt à être testé, nous vous recommandons de disposer d'un SDK pouvant se connecter à AWS IoT Core et mettez à jour le SDK avec le point de terminaison de test Device Advisor fourni pour votre compte.

La commande pour obtenir le point de terminaison de test est :

```
aws iot describe-endpoint --endpoint-type iot:DeviceAdvisor --region region
```

Démarrer avec Device Advisor dans la console

Ce didacticiel vous aide à démarrer rapidement avec Device Advisor dans la console. Device Advisor offre des fonctionnalités telles que les tests requis et les rapports de qualification signés pour qualifier et répertorier les périphériques dans leAWSCatalogue des appareils partenairesconformément à la AWS IoT Core Programme de qualification.

Pour plus d'informations sur l'utilisation de Device Advisor, consultezDevice Advisor Flux (p. 1014)andWorkflow détaillé de la console Device (p. 1018).

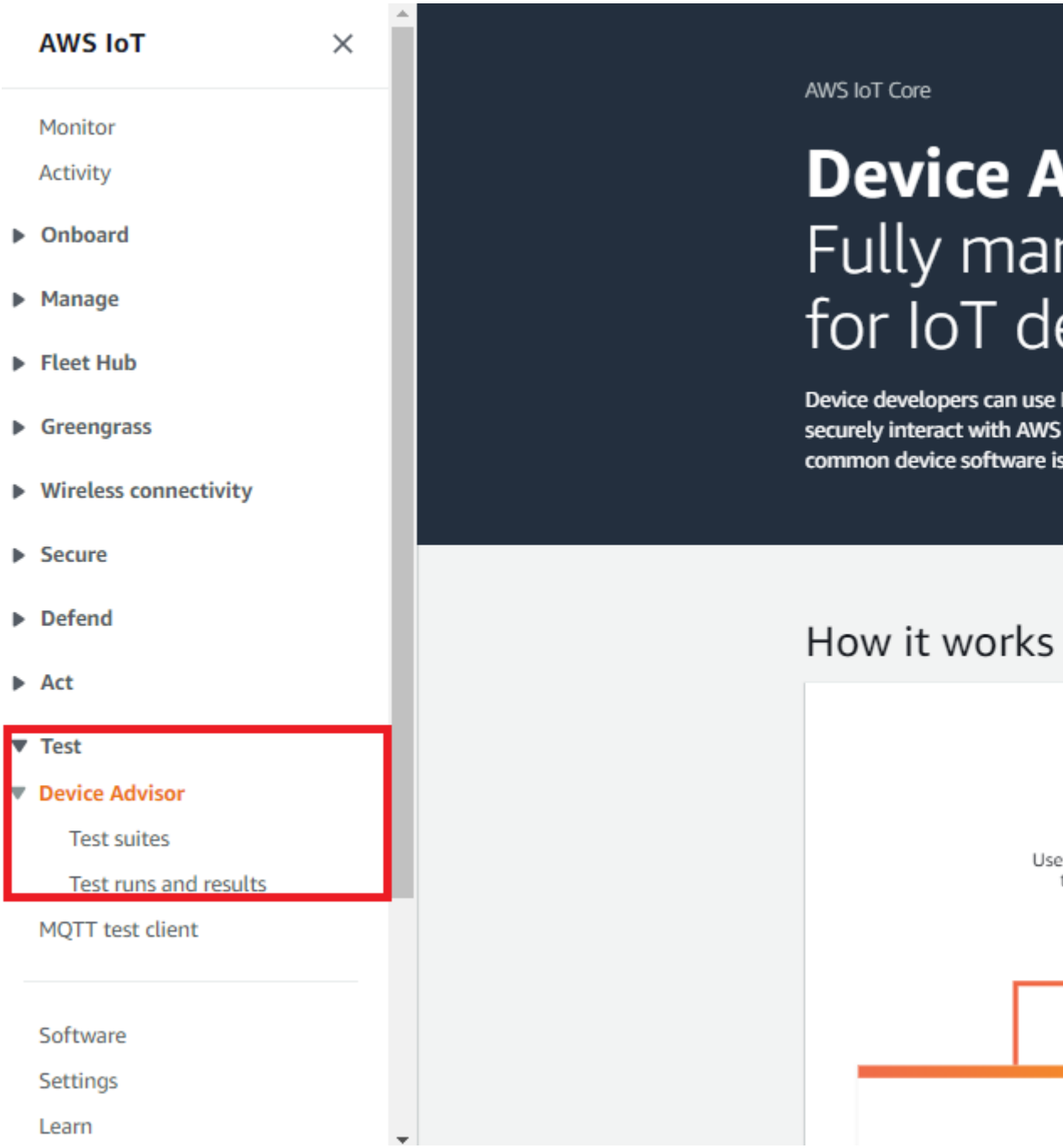
Pour suivre ce didacticiel, suivez les étapes décrites dansConfiguration de (p. 998).

Note

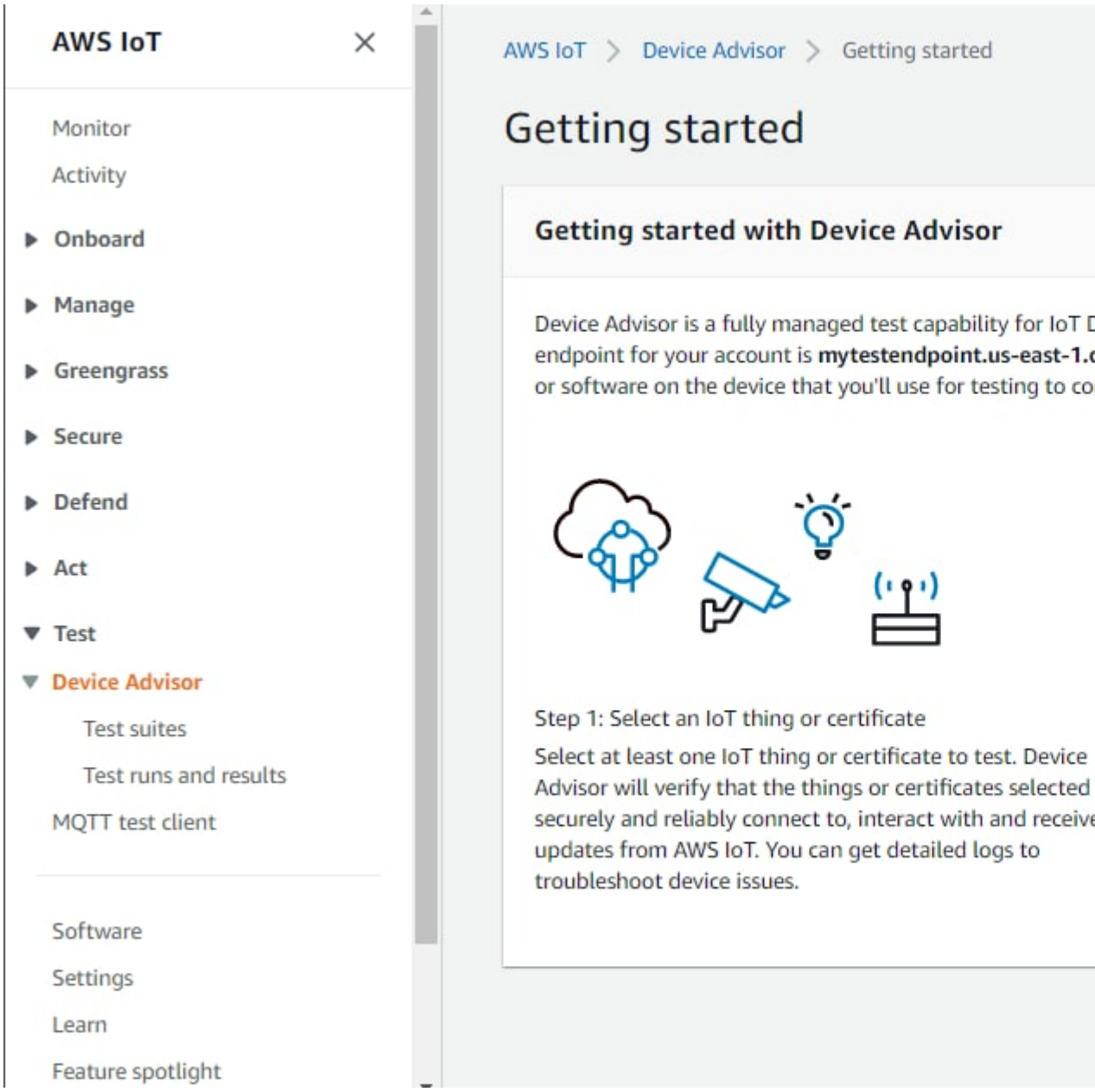
Device Advisor est pris en charge dans les régions us-east-1, us-west-1 et eu-west-1.

Mise en route

1. DansAWS IoTconsoleDans le volet de navigation, développezTest,Device Advisor, puis.Démarrer la procédure pas à pas.

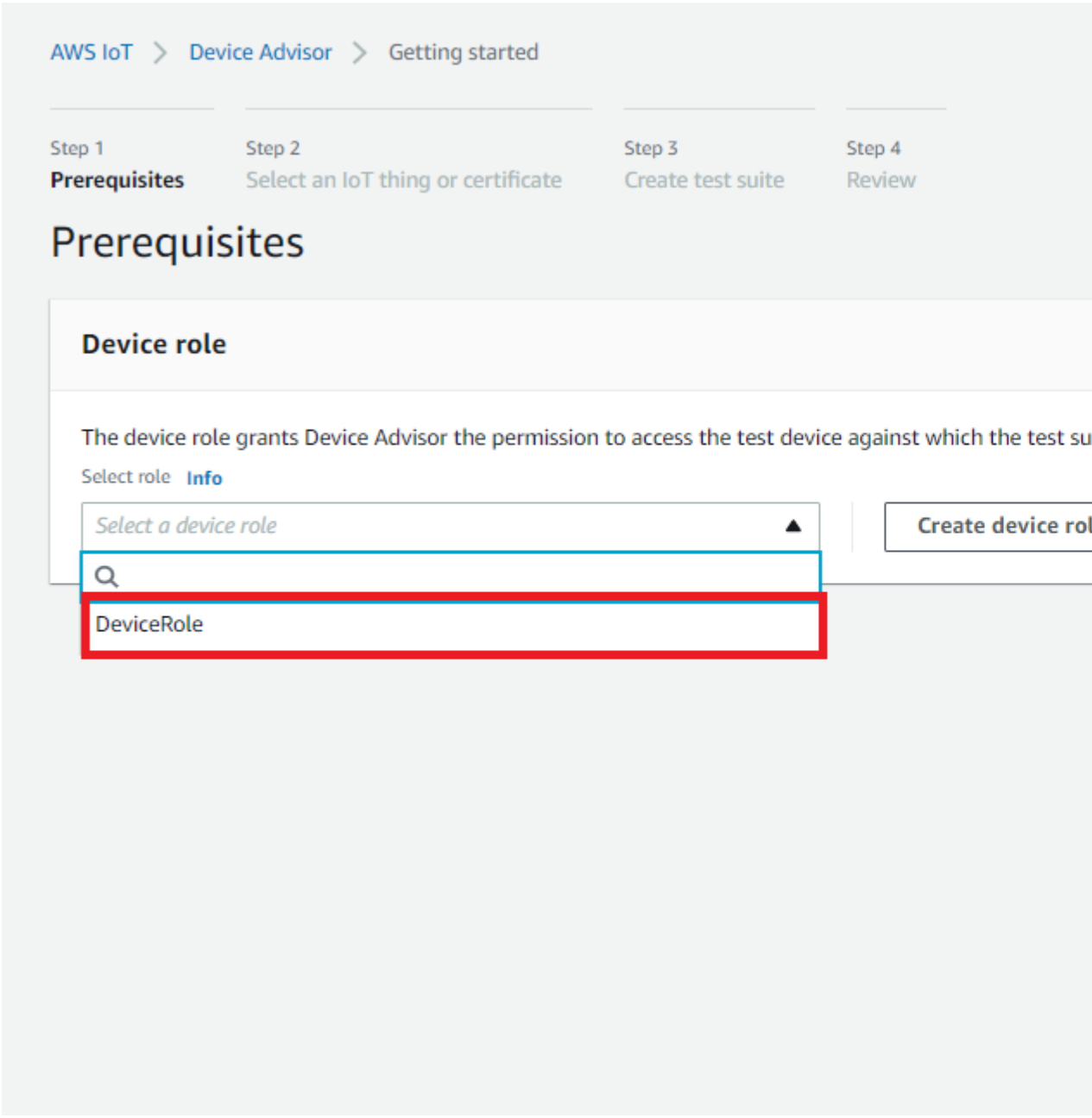


2. La .Mise en route avec Device Advisordonne un aperçu des étapes requises pour créer une suite de tests et exécuter des tests sur votre appareil. Vous pouvez également trouver le point de terminaison de test Device Advisor pour votre compte. Vous devez configurer le firmware ou le logiciel sur le périphérique que vous utiliserez pour les tests afin de vous connecter à ce point de terminaison de test.



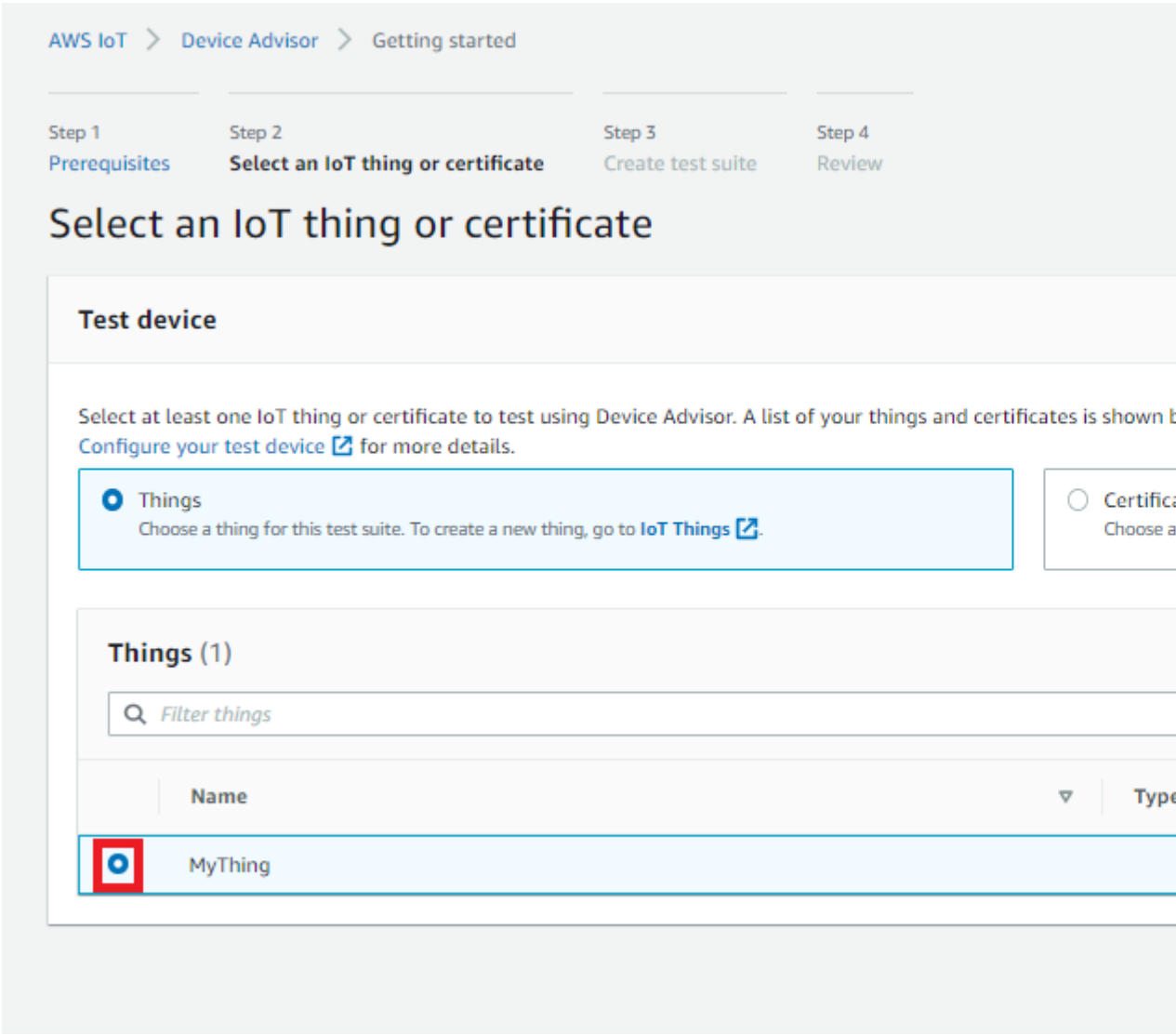
Après avoir examiné les informations, choisissezSuivant.

3. DansÉtape 1, sélectionnez le rôle de périphérique qui a été créé dans le cadre deConfiguration de.

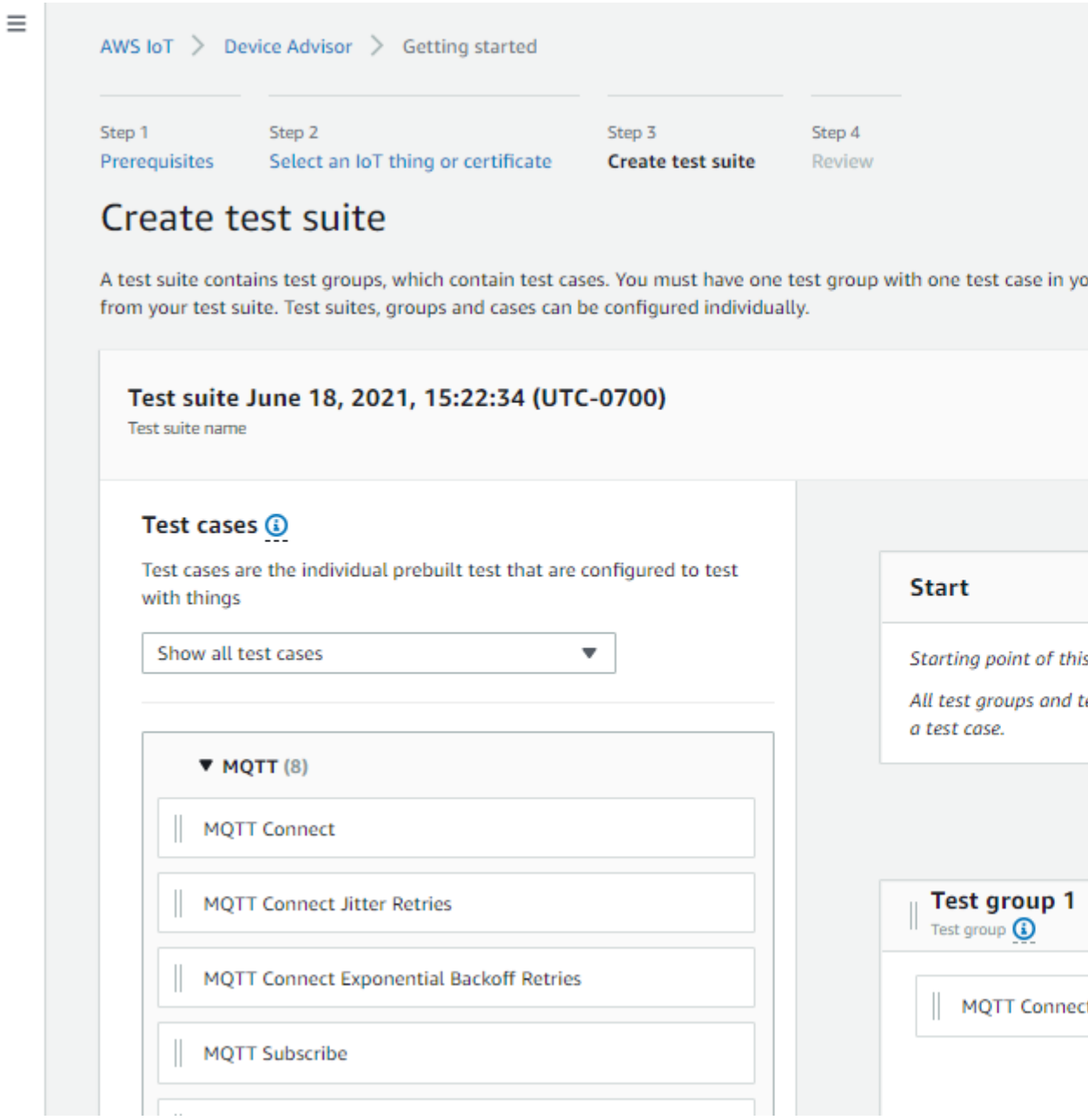


4. DansÉtape 1, vous avez sélectionné unAWS IoTchose ou certificat à tester à l'aide de Device Advisor. Si vous ne disposez d'aucun objet ou certificat existant, consultez.Configuration de (p. 998).

   Si la valeur est affichéeAWS IoTchose ou certificat que vous avez configuré avec votre point de terminaison de test Device Advisor, choisissez ce périphérique dans la liste, puis choisissezSuivant.

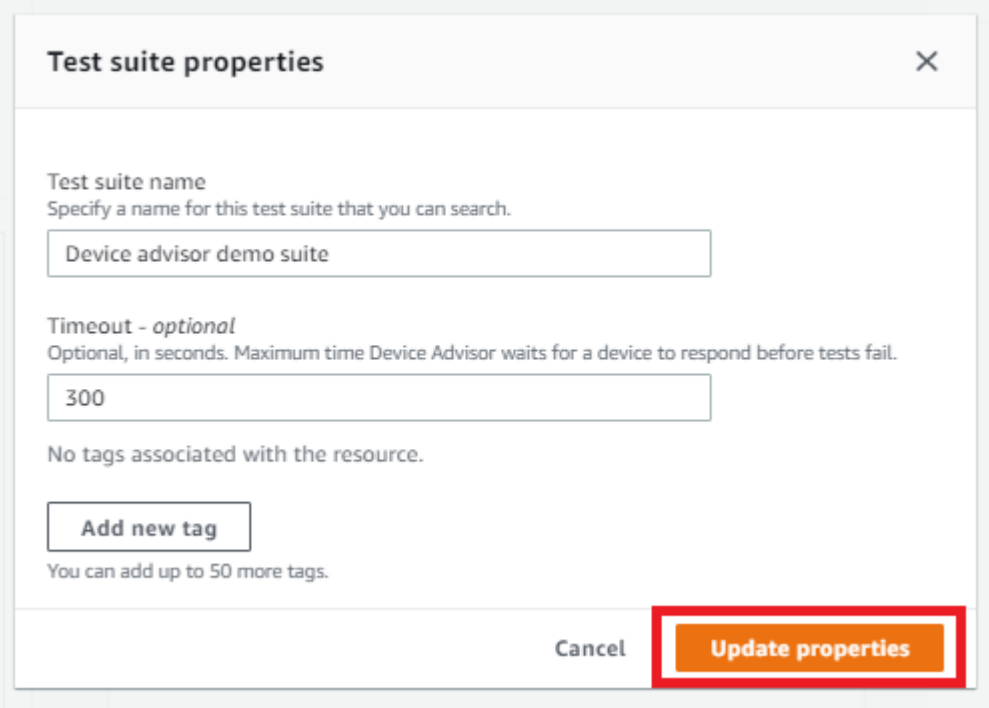


5. DansÉtape 2, vous pouvez créer et configurer une suite de tests personnalisée. Une suite de tests personnalisée doit comporter au moins un groupe de tests et chaque groupe de tests doit comporter au moins un cas de test. Nous avons ajouté leConnect MQTTcas de test pour vous de commencer.
6. ChoisissezPropriétés de la suite de tests. Vous devez fournir les propriétés de la suite de tests lorsque vous créez votre suite de tests.



Vous pouvez configurer les propriétés de niveau suite ici.

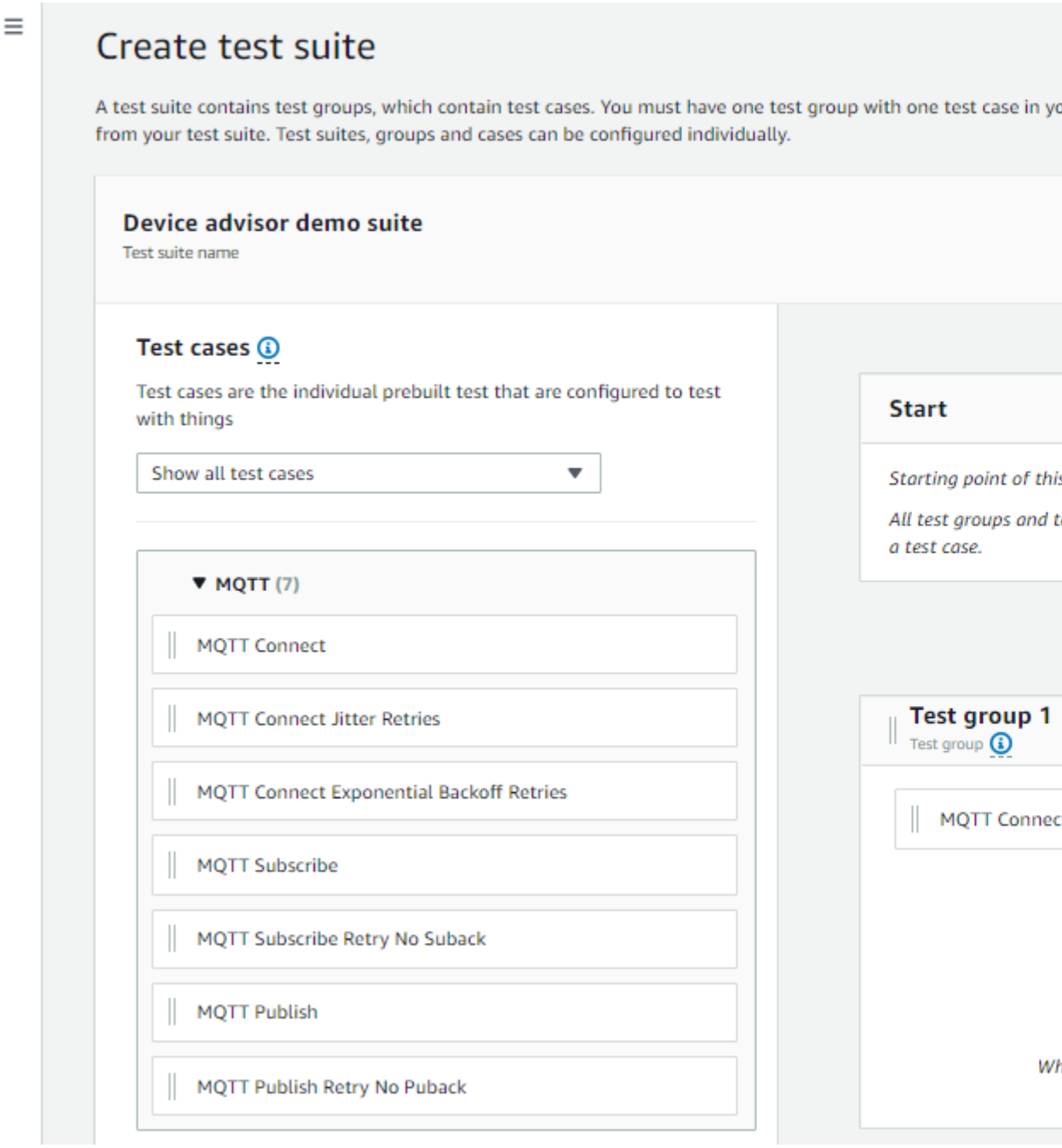
- Nom de la suite de tests : Vous pouvez créer la suite avec un nom personnalisé.
- Timeout (Expiration)(facultatif) : Délai d'expiration en secondes pour chaque cas de test dans la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.
- Tags (Balises)(facultatif) : Ajoutez des balises à la suite de tests que vous allez créer.



Lorsque vous avez terminé, choisissezMettre à jour les.

7. (Facultatif) Vous pouvez mettre à jour la configuration du groupe de la suite de tests en sélectionnantModifierEn regard du nom du groupe de test.

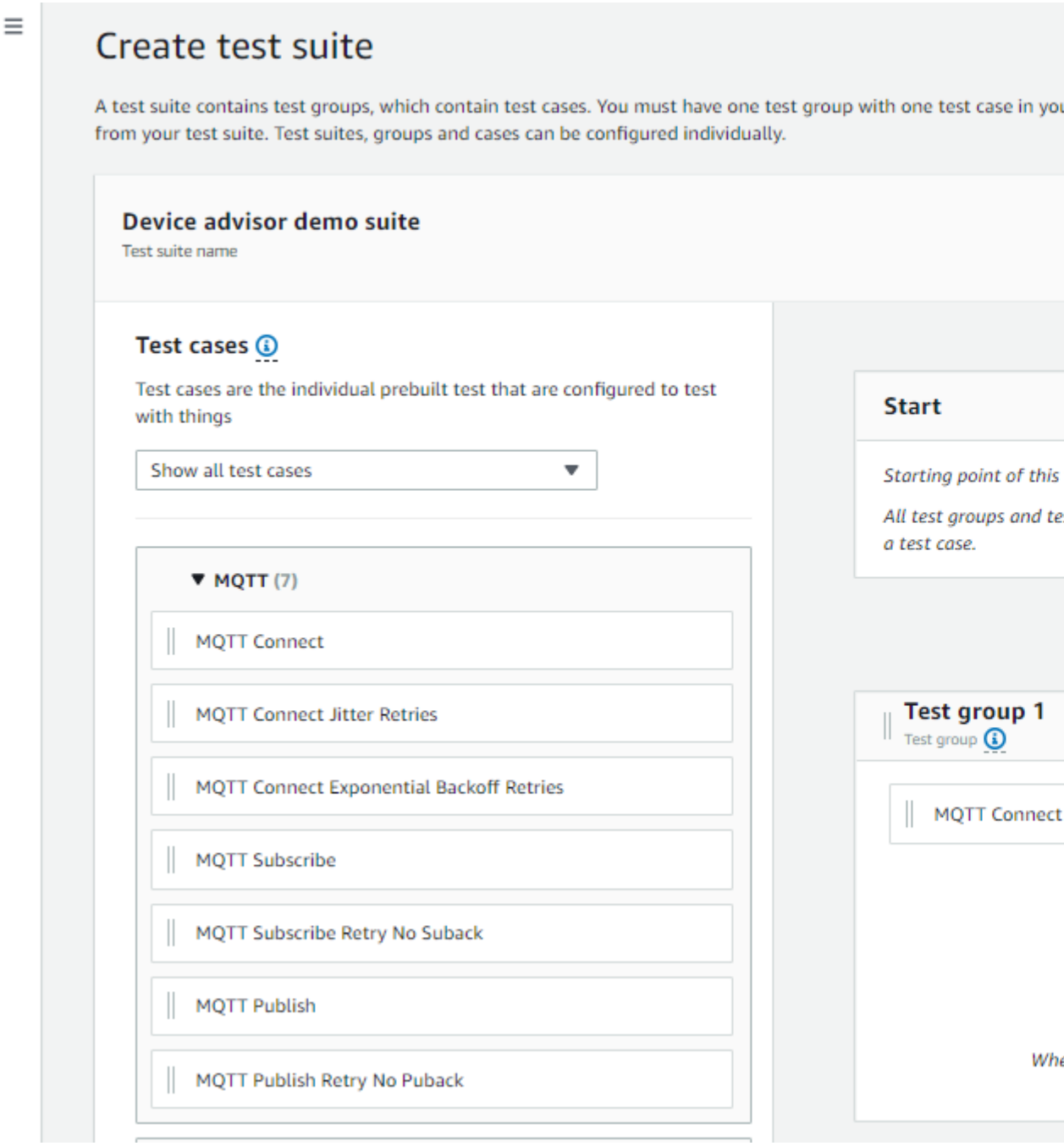
   - Nom : Vous pouvez mettre à jour le groupe avec un nom personnalisé.
   - Timeout (Expiration)(facultatif) : Délai d'expiration en secondes pour chaque cas de test dans la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.



Sélectionnez Done (Effectué).

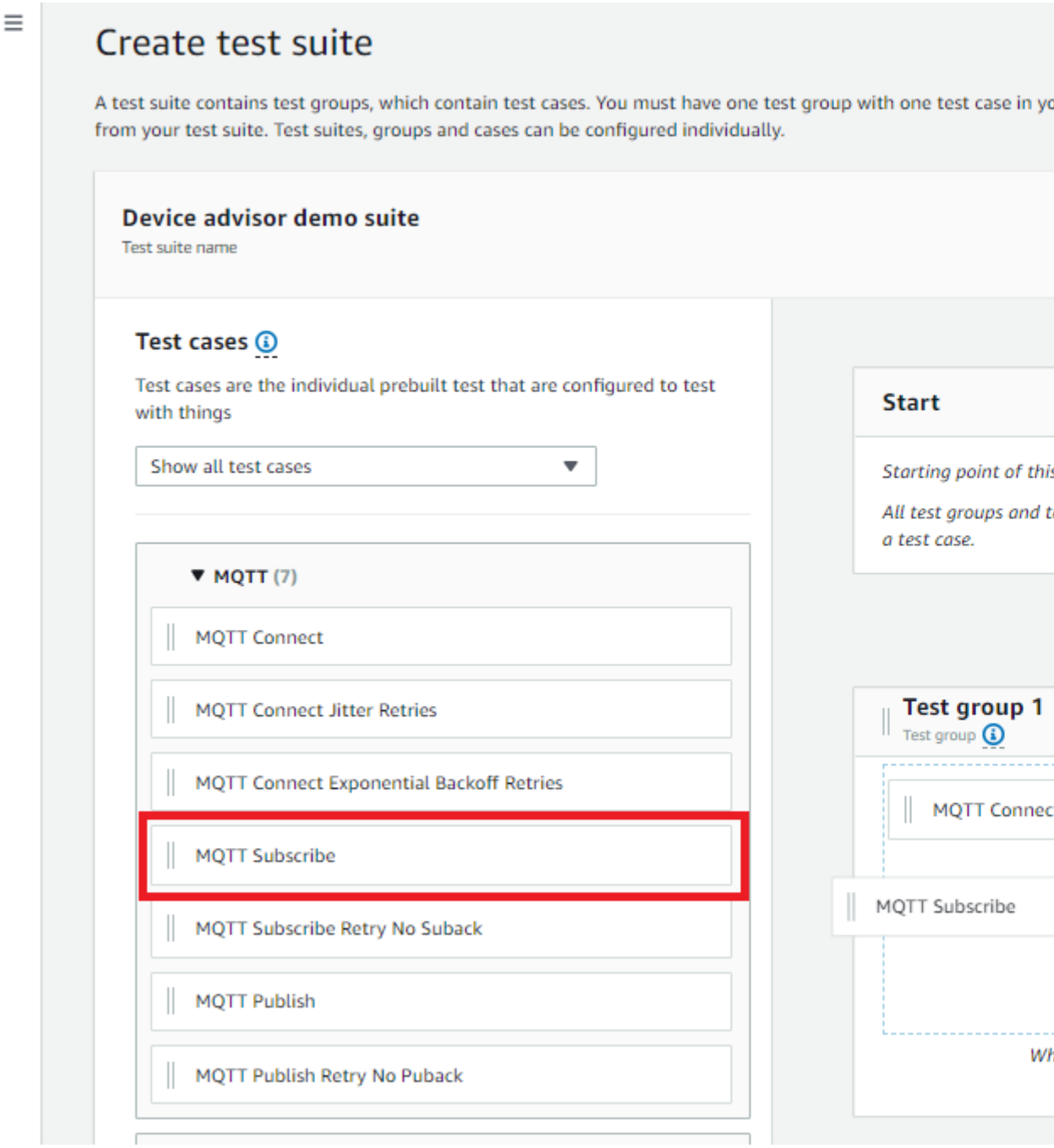
8. (Facultatif) Vous pouvez mettre à jour la configuration du cas de test en sélectionnantModifierEn regard du nom du cas de test.

   - Nom : Vous pouvez mettre à jour le cas de test avec un nom personnalisé.
   - Timeout (Expiration)(facultatif) : Délai d'expiration en secondes pour le cas de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.



Sélectionnez Done (Effectué).

9. (Facultatif) Pour ajouter d'autres groupes de tests à la suite de tests, choisissezAjouter un groupe testet configurez en conséquence.
10. (Facultatif) Pour ajouter d'autres cas de test, faites glisser les cas de test sousCas d'essaidans l'un des groupes de test que vous possédez.



11. Les groupes de test et les cas de test peuvent être réorganisés en faisant glisser. Device Advisor exécute les tests dans l'ordre où ils sont définis.

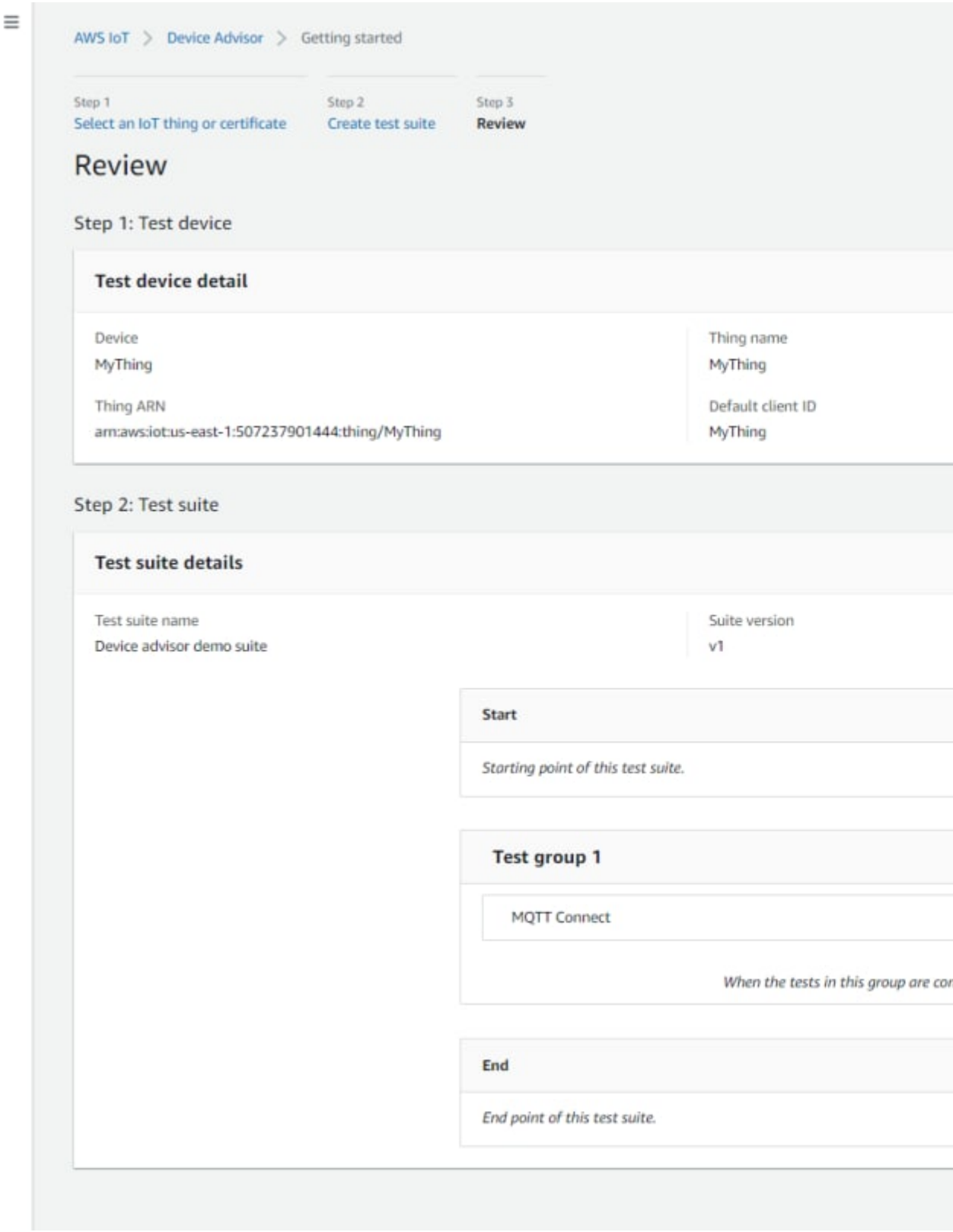
    Après avoir configuré votre suite de tests, choisissezSuivant.

12. Étape 4vous fournit une vue d'ensemble du rôle de périphérique de test, du périphérique de test et de la suite de tests sélectionnés que vous avez configurés. Si vous souhaitez apporter des modifications, choisissezModifier.

Note

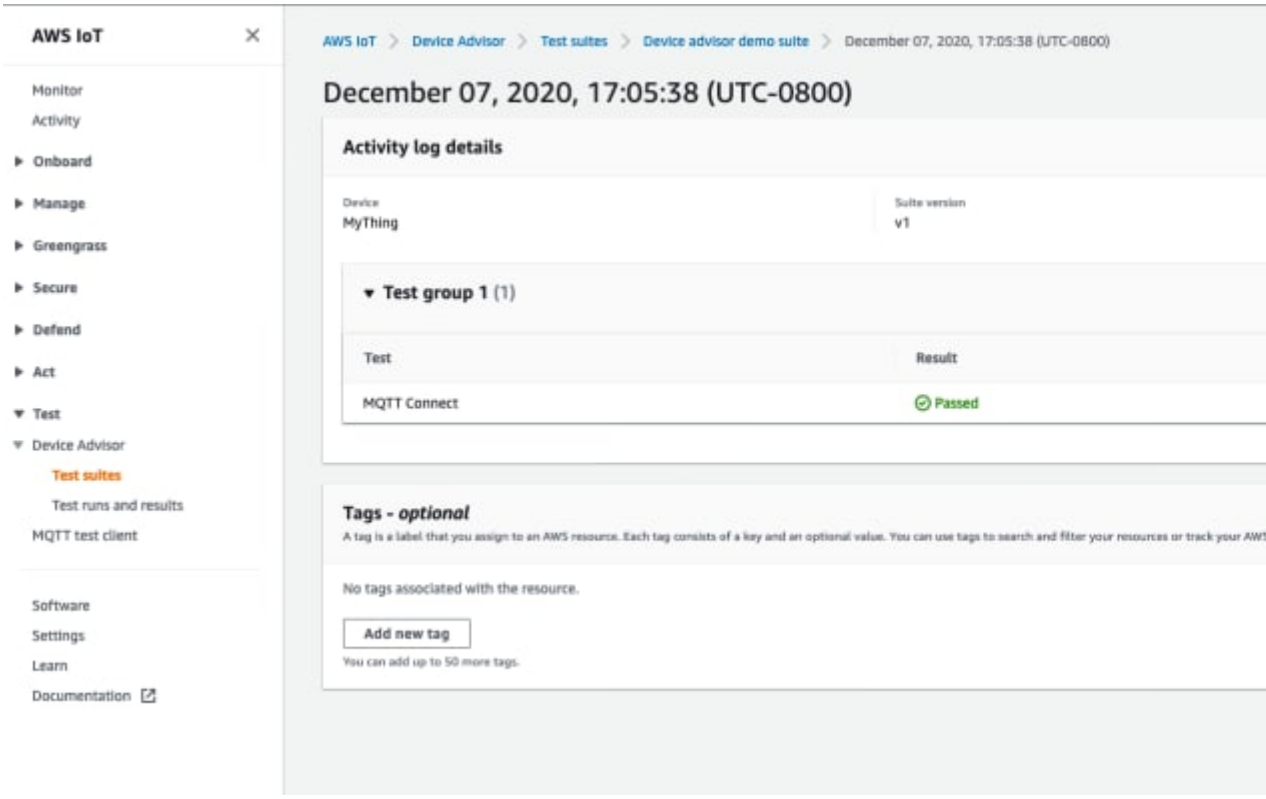
Pour une expérience plus fluide, vous pouvez connecter le périphérique de test sélectionné au point de terminaison de test Device Advisor avant de démarrer l'exécution de la suite. Nous vous recommandons de disposer d'un mécanisme conçu pour que votre appareil essaie de se connecter à notre terminal de test toutes les 5 secondes pendant une à deux minutes.

Pour créer la suite de tests et exécuter les tests sur votre appareil, choisissezRun (Exécuter Lambda).


AWS IoT > Device Advisor > Getting started
Step 1
Select an IoT thing or certificate
Step 2
Create test suite
Step 3
Review
Review
Step 1: Test device
Test device detail
Device
MyThing
Thing name
MyThing
Thing ARN
arn:aws:iot:us-east-1:507237901444:thing/MyThing
Default client ID
MyThing
Step 2: Test suite
Test suite details
Test suite name
Device advisor demo suite
Suite version
v1
Start
Starting point of this test suite.
Test group 1
MQTT Connect
When the tests in this group are co
End
End point of this test suite.

13. Dans le volet de navigation, développezTest,Device Advisor, puis.Tests et résultatsPour afficher les détails de l'exécution et les journaux.

    Sélectionnez l'exécution de la suite de tests qui a été démarrée pour afficher les détails de l'exécution et les journaux.



14. Pour accéder aux journaux CloudWatch de la suite, exécutez :

    - ChoisissezJournal de la suite de testspour afficher les journaux CloudWatch pour l'exécution de la suite de tests.
    - ChoisissezJournal des cas de testpour n'importe quel cas de test pour afficher les journaux CloudWatch spécifiques au cas de test.

15. Sur la base de vos résultats de test,DépannageVotre appareil jusqu'à ce que tous les tests soient passés.

Device Advisor Flux

Ce didacticiel fournit des instructions sur la création d'une suite de tests personnalisée et l'exécution de tests sur le périphérique que vous souhaitez tester dans la console. Une fois les tests terminés, vous pouvez afficher les résultats de test et les journaux détaillés.

Didacticiels

- Prerequisites (p. 1015)
- Créer une définition de suite de tests (p. 1015)
- Obtenir une suite de tests (p. 1017)
- Démarrer une exécution de la suite de tests (p. 1017)
- Obtenir une définition de suite de tests (p. 1017)

- Arrêter l'exécution d'une suite de tests (p. 1017)
- Obtenir un rapport de qualification pour une suite de tests de qualification réussie (p. 1018)

Prerequisites

Pour suivre ce didacticiel, suivez les étapes décrites dansConfiguration de (p. 998).

Créer une définition de suite de tests

Premièrement,Installer unAWSKIT DE DÉVELOPPEMENT LOGICIEL.

Syntaxe de `rootGroup`

Un groupe racine est une chaîne JSON qui spécifie quels cas de test sont inclus dans votre suite de tests et toutes les configurations nécessaires pour ces cas de test. Utilisez le groupe racine pour structurer et commander votre suite de tests comme vous le souhaitez. La hiérarchie d'une suite de tests est la suivante :

```
test suite # test group(s) # test case(s)
```

Une suite d'essais doit comporter au moins un groupe d'essais et chaque groupe d'essais doit comporter au moins un cas d'essai. Device Advisor exécute des tests dans l'ordre dans lequel vous définissez les groupes de test et les cas de test.

Chaque groupe racine a cette structure de base :

```
{
    "configuration": {  // for all tests in the test suite
        "": ""
    }
    "tests": [{
        "name": ""
        "configuration": {  // for all sub-groups in this test group
            "": ""
        },
        "tests": [{
            "name": ""
            "configuration": {  // for all test cases in this test group
                "": ""
            },
            "test": {
                "id": ""
                "version": ""
            }
        }]
    }]
}
```

Un bloc qui contient un`"name"`,`"configuration"`, et`"tests"`est appelée « définition de groupe ».
Un bloc qui contient un`"name"`,`"configuration"`, et`"test"`est appelée « définition de cas type ».
EACH`"test"`qui contient un`"id"`and`"version"`est appelé « cas type ».

Pour plus d'informations sur la façon de remplir le`"id"`and`"version"`pour chaque cas de test (`"test"`), voirDevice Advisor (p. 1036). Cette section contient également des informations sur les`"configuration"`Paramètres de .

Voici un exemple de configuration de groupe racine qui spécifie les cas de test « MQTT Connect Happy Case » et « MQTT Connect Exponential Backoff Retries » ainsi que les descriptions des champs de configuration.

```
{
    "configuration": {},  // Suite-level configuration
    "tests": [            // Group definitions should be provided here
      {
        "name": "My_MQTT_Connect_Group",  // Group definition name
        "configuration": {}               // Group definition-level configuration,
        "tests": [                        // Test case definitions should be provided here
        {
            "name": "My_MQTT_Connect_Happy_Case",  // Test case definition name
            "configuration": {
                "EXECUTION_TIMEOUT": 300        // Test case definition-level
 configuration, in seconds
            },
            "test": {
                "id": "MQTT_Connect",              // test case id
                "version": "0.0.0"                 // test case version
            }
        },
        {
            "name": "My_MQTT_Connect_Jitter_Backoff_Retries",  // Test case definition name
            "configuration": {
                "EXECUTION_TIMEOUT": 600                // Test case definition-level
 configuration,  in seconds
            },
            "test": {
                "id": "MQTT_Connect_Jitter_Backoff_Retries",  // test case id
                "version": "0.0.0"                                  // test case version
            }
        }]
    }]
}
```

Vous devez fournir la configuration du groupe racine lorsque vous créez la définition de la suite de tests. Enregistrer le`suiteDefinitionId`qui est retourné dans l'objet de réponse : il est utilisé pour récupérer les informations de définition de votre suite de tests et pour exécuter votre suite de tests.

Voici un exemple de SDK Java :

```
response = iotDeviceAdvisorClient.createSuiteDefinition(
        CreateSuiteDefinitionRequest.builder()
            .suiteDefinitionConfiguration(SuiteDefinitionConfiguration.builder()
                .suiteDefinitionName("your-suite-definition-name")
                .devices(
                    DeviceUnderTest.builder()
                        .thingArn("your-test-device-thing-arn")
                        .certificateArn("your-test-device-certificate-arn")
                        .build()
                )
                .rootGroup("your-root-group-configuration")
                .devicePermissionRoleArn("your-device-permission-role-arn")
                .build()
            )
            .build()
)
```

Obtenir une suite de tests

Après avoir créé votre définition de suite de tests, vous recevez le`suiteDefinitionId`dans l'objet de réponse de la`CreateSuiteDefinition`API.

Vous pouvez voir qu'il y a de nouveaux`id`dans chacune des définitions de groupe et de cas de test du groupe racine renvoyé. Ceci est attendu et vous pouvez utiliser ces ID pour exécuter un sous-ensemble de la définition de votre suite de tests.

Exemple SDK Java :

```
response = iotDeviceAdvisorClient.GetSuiteDefinition(
    GetSuiteDefinitionRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .build()
)
```

Démarrer une exécution de la suite de tests

Après avoir créé avec succès une définition de suite de tests et configuré votre périphérique de test pour se connecter à votre terminal de test Device Advisor, exécutez votre suite de tests avec l'outil`StartSuiteRun`API. Utilisez soit`certificateArn`ou`thingArn`pour exécuter la suite de tests. Si les deux sont configurés, le certificat sera utilisé s'il appartient à la chose.

Exemple de kit SDK :

```
response = iotDeviceAdvisorClient.startSuiteRun(StartSuiteRunRequest.builder()
.suiteDefinitionId("your-suite-definition-id")
.suiteRunConfiguration(SuiteRunConfiguration.builder()
    .primaryDevice(DeviceUnderTest.builder()
        .certificateArn("your-test-device-certificate-arn")
        .thingArn("your-test-device-thing-arn")
        .build())
    .build())
.build())
```

Enregistrer le`suiteRunId`qui est retourné dans la réponse : vous l'utiliserez pour récupérer les résultats de cette exécution de la suite de tests.

Obtenir une définition de suite de tests

Après avoir démarré une suite de tests, vous pouvez vérifier sa progression et ses résultats à l'aide de l'outil`GetSuiteRun`API.

Exemple de kit SDK :

```
// Using the SDK, call the GetSuiteRun API.

response = iotDeviceAdvisorClient.GetSuiteRun(
GetSuiteRunRequest.builder()
    .suiteDefinitionId("your-suite-definition-id")
    .suiteRunId("your-suite-run-id")
.build())
```

Arrêter l'exécution d'une suite de tests

AWS IoTDevice Advisor permet l'exécution d'une suite de tests active à la fois. Pour arrêter l'exécution d'une suite de tests toujours en cours, vous pouvez appeler la méthode`StopSuiteRun`API. Après

avoir appelé leStopSuiteRun, le service démarre le processus de nettoyage. Pendant que le service exécute le processus de nettoyage, l'état d'exécution de la suite de tests seraStopping. Le processus de nettoyage prend plusieurs minutes, et une fois le processus terminé, l'exécution de la suite de tests aura leStoppedÉtat. Une fois qu'une exécution de test s'est complètement arrêtée, vous pourrez démarrer une autre exécution de la suite de tests. Vous pouvez vérifier périodiquement l'état d'exécution de la suite à l'aide de l'outilGetSuiteRunComme indiqué dans la section précédente.

Exemple de kit SDK :

```
// Using the SDK, call the StopSuiteRun API.

response = iotDeviceAdvisorClient.StopSuiteRun(
StopSuiteRun.builder()
    .suiteDefinitionId("your-suite-definition-id")
    .suiteRunId("your-suite-run-id")
.build())
```

Obtenir un rapport de qualification pour une suite de tests de qualification réussie

Si vous exécutez une suite de tests de qualification qui se termine, vous pouvez récupérer un rapport de qualification à l'aide de l'API GetSuiteRunReport. Vous pouvez utiliser ce rapport de qualification pour qualifier votre appareil avec le AWS IoT Core Programme de qualification. Pour déterminer si votre suite de tests est une suite de tests de qualification, vérifiez si laintendedForQualificationest défini surtrue. Après avoir appelé l'API GetSuiteRunReport, l'URL de téléchargement renvoyée est disponible pendant 90 secondes. Si plus de 90 secondes se sont écoulées depuis la précédente fois que vous avez appelé l'API GetSuiteRunReport, appelez à nouveau l'API pour récupérer une URL valide.

Exemple de kit SDK :

```
// Using the SDK, call the getSuiteRunReport API.

response = iotDeviceAdvisorClient.getSuiteRunReport(
    GetSuiteRunReportRequest.builder()
        .suiteDefinitionId("your-suite-definition-id")
        .suiteRunId("your-suite-run-id")
        .build()
)
```

Workflow détaillé de la console Device

Dans ce didacticiel, vous allez créer une suite de tests personnalisée et exécuter des tests sur le périphérique que vous souhaitez tester dans la console. Une fois les tests terminés, vous pouvez afficher les résultats de test et les journaux détaillés.

Didacticiels

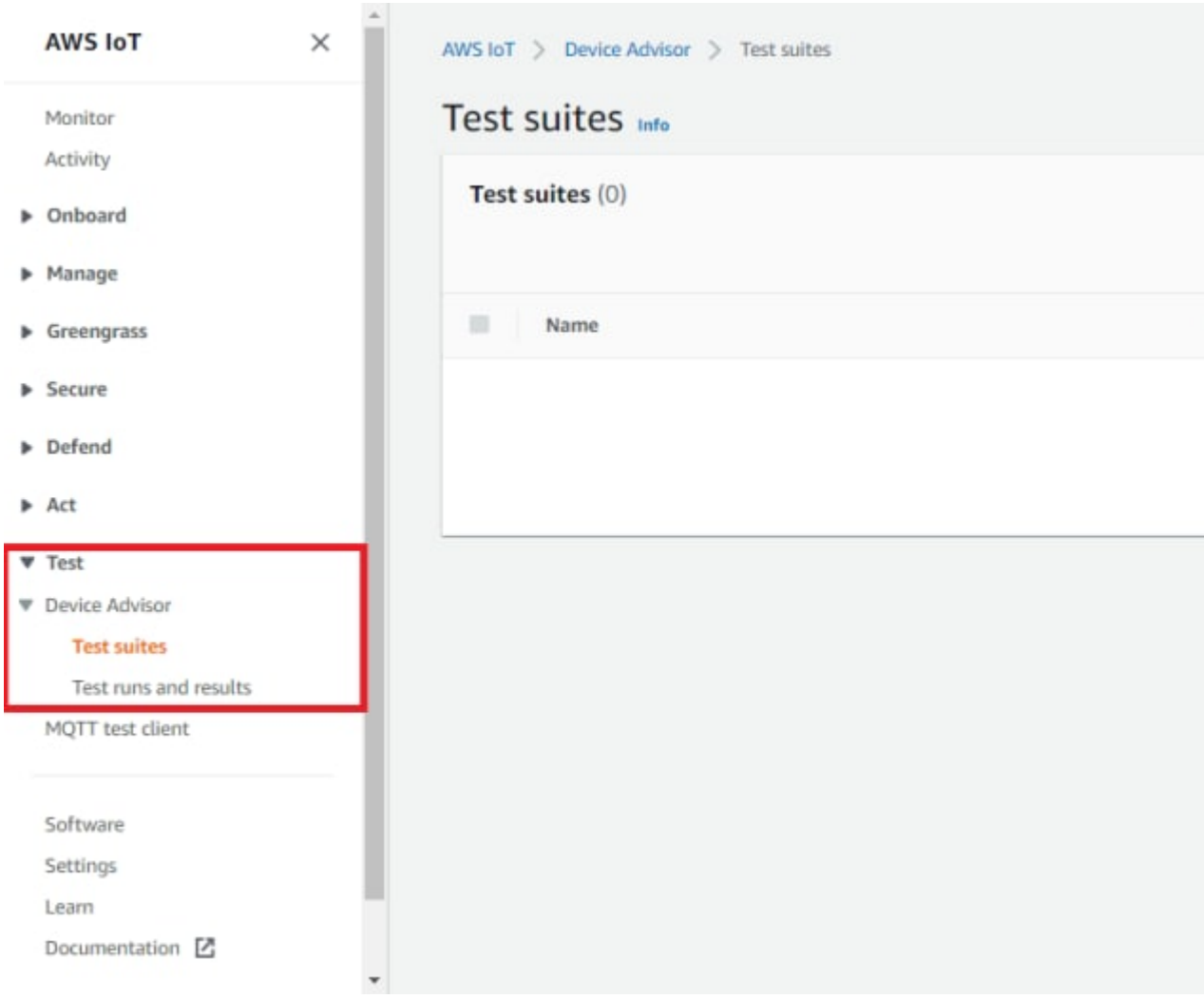
- Prerequisites (p. 1019)
- Créer une définition de suite de tests (p. 1019)
- Démarrer une exécution de la suite de tests (p. 1027)
- Arrêter l'exécution d'une suite de tests (facultatif) (p. 1030)
- Afficher les détails et les journaux de l'exécution de la suite (p. 1033)
- Télécharger unAWS IoTRapport de qualification (p. 1035)

Prerequisites

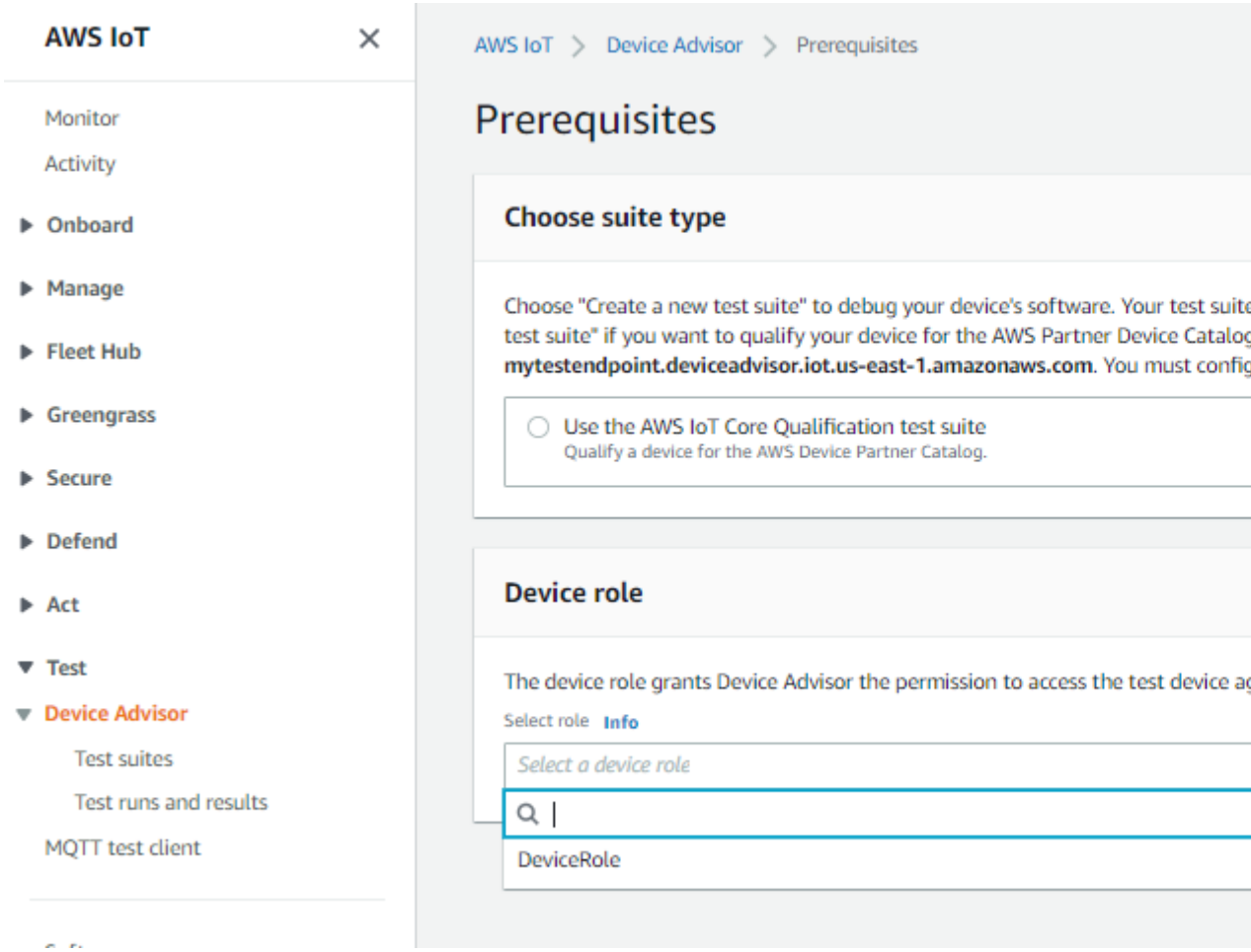
Pour suivre ce didacticiel, vous devez suivre les étapes décrites dansConfiguration de (p. 998).

Créer une définition de suite de tests

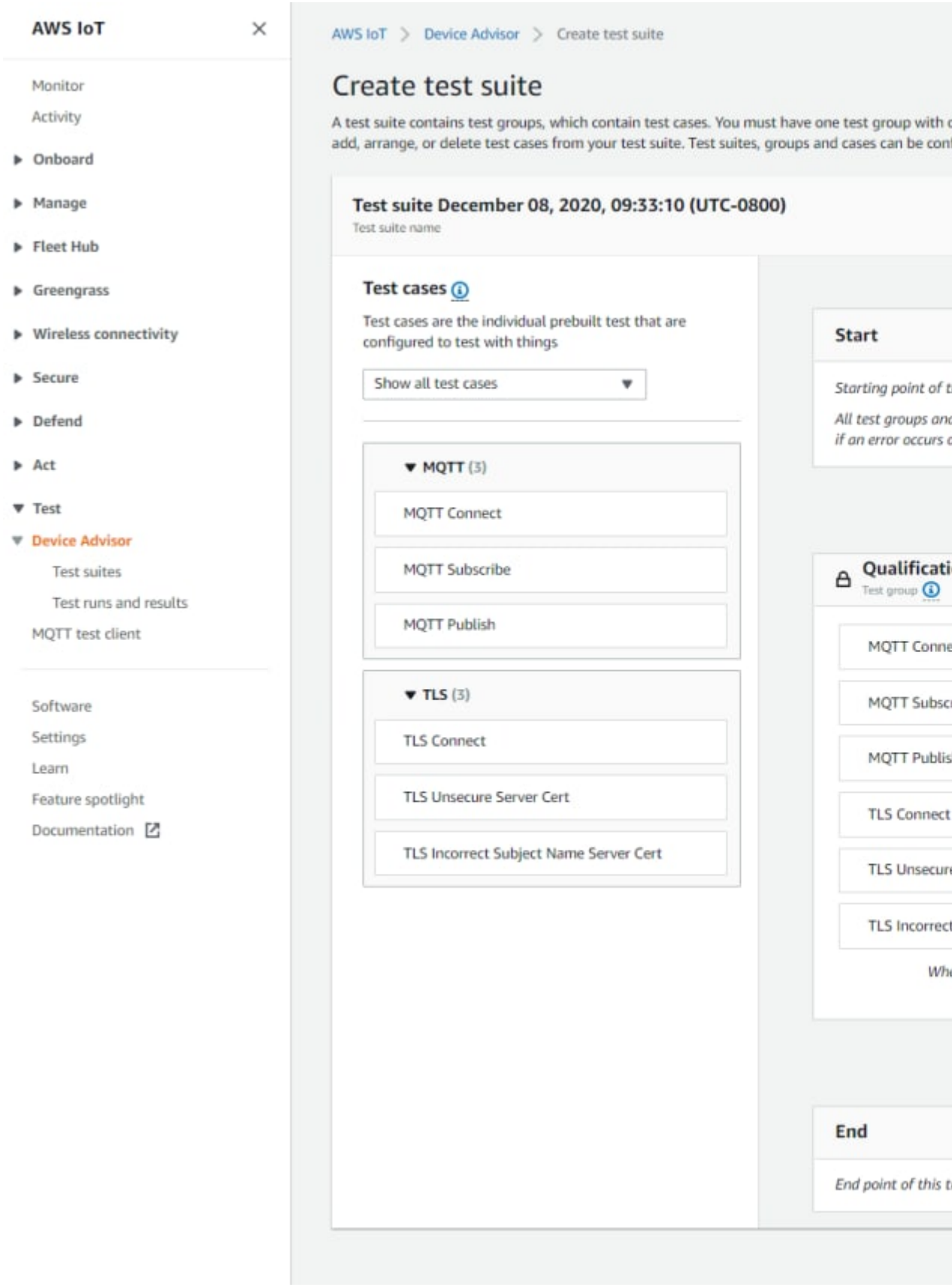
1. DansAWS IoTconsoleDans le volet de navigation, développezTest,Device Advisor, puis.Suites test.



2. Tâche de sélectionCréation d'une suite de tests. Choisissez entre`Use the AWS Qualification test suite`and`Create a new test suite`. Sélectionnez le rôle de périphérique qui a été créé dans le cadre deConfiguration de.

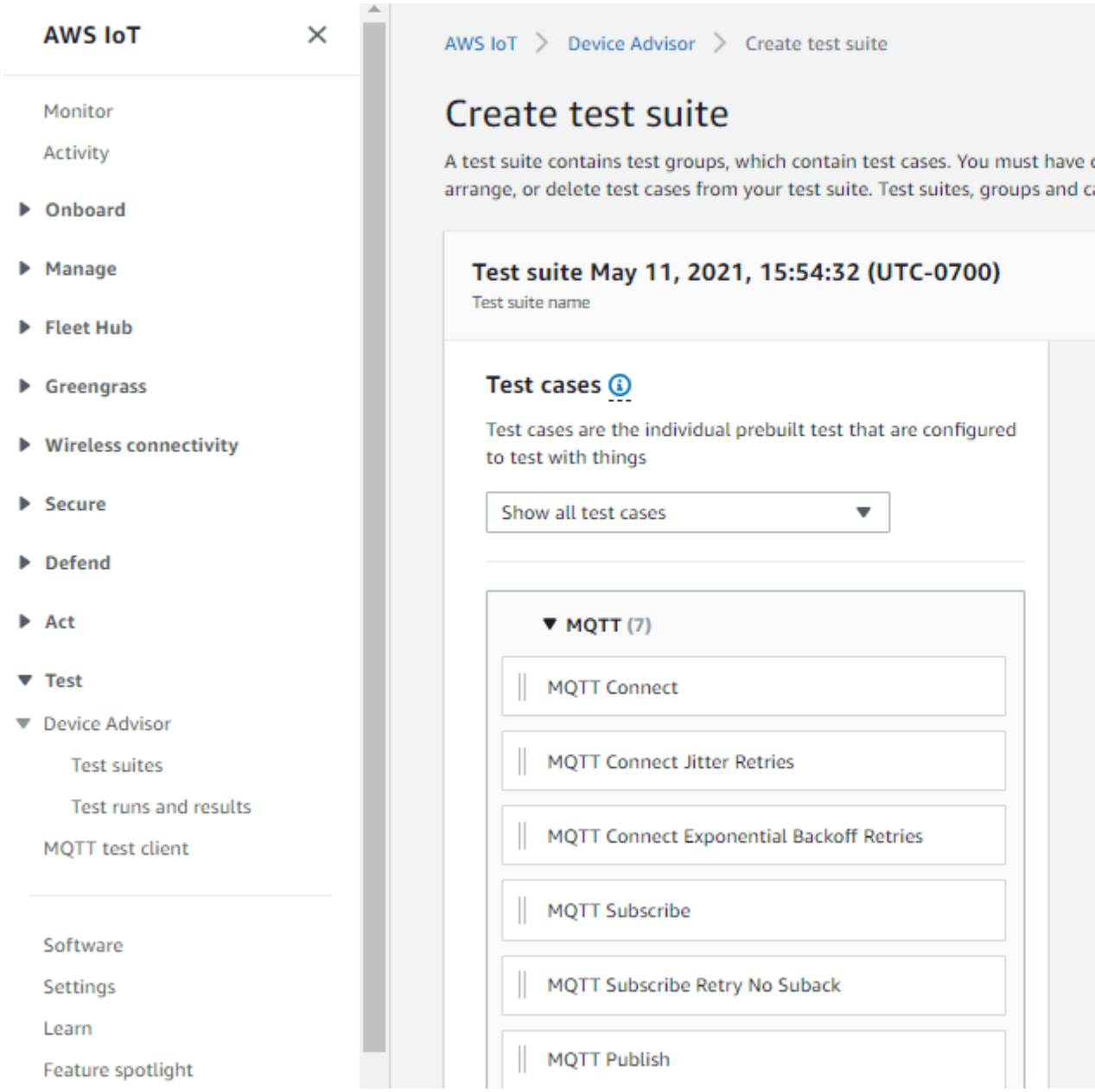


Choisissez`Use the AWS Qualification test suite`si vous souhaitez vous qualifier et répertorier votre appareil dans leAWSCatalogue d'appareils partenaires. En choisissant cette option, les cas de test requis pour la qualification de votre appareil au AWS IoT Core programme de qualification sont présélectionnés. Les groupes de test et les cas de test ne peuvent pas être ajoutés ou supprimés. Cependant, vous devrez toujours configurer les propriétés de la suite de tests.


AWS IoT
Monitor
Activity
Onboard
Manage
Fleet Hub
Greengrass
Wireless connectivity
Secure
Defend
Act
Test
Device Advisor
Test suites
Test runs and results
MQTT test client
Software
Settings
Learn
Feature spotlight
Documentation
AWS IoT > Device Advisor > Create test suite
Create test suite
A test suite contains test groups, which contain test cases. You must have one test group with
add, arrange, or delete test cases from your test suite. Test suites, groups and cases can be con
Test suite December 08, 2020, 09:33:10 (UTC-0800)
Test suite name
Test cases
Test cases are the individual prebuilt test that are configured to test with things
Show all test cases
MQTT (3)
MQTT Connect
MQTT Subscribe
MQTT Publish
TLS (3)
TLS Connect
TLS Unsecure Server Cert
TLS Incorrect Subject Name Server Cert
Start
Starting point of t
All test groups an
if an error occurs
Qualificati
Test group
MQTT Conne
MQTT Subsc
MQTT Publis
TLS Connect
TLS Unsecur
TLS Incorrect
Wh
End
End point of this t

ChoisissezCreate a new test suitepour créer et configurer une suite de tests personnalisée. Nous vous recommandons de commencer par cette option pour les tests initiaux et le dépannage. Une suite de tests personnalisée doit comporter au moins un groupe de tests et chaque groupe de tests doit comporter au moins un cas de test. Dans ce didacticiel, nous allons sélectionner cette option et choisirSuivant.

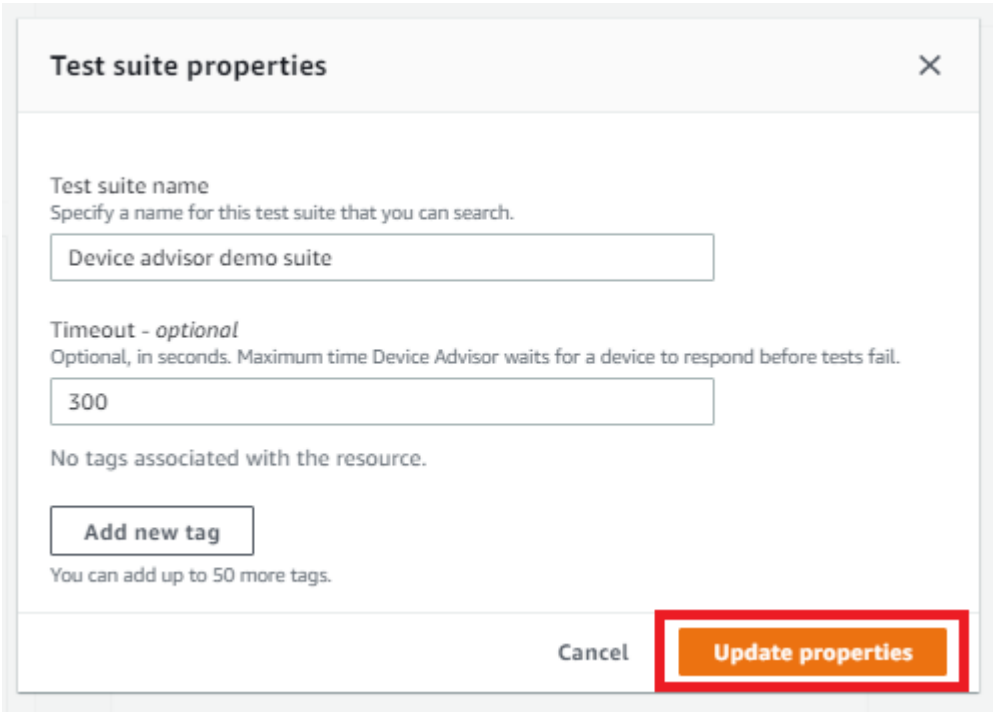
3. ChoisissezPropriétés de la suite de tests. Vous devez ajouter les propriétés de la suite de tests lorsque vous créez votre suite de tests.



UnderPropriétés de la suite de tests, remplissez ce qui suit.

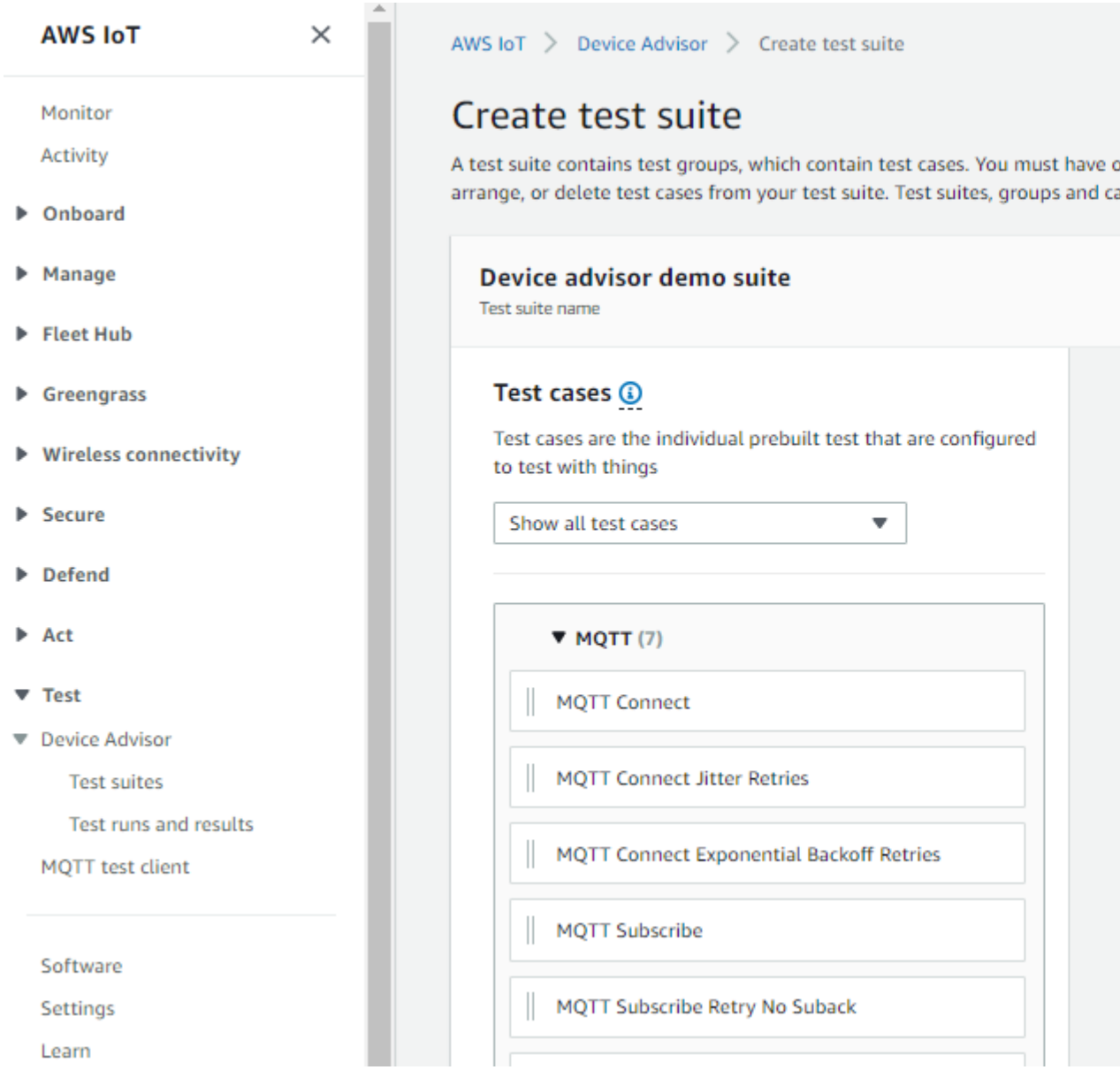
- Nom de la suite de tests : Vous pouvez créer la suite avec un nom personnalisé.

- Timeout (Expiration)(facultatif) : Délai d'expiration en secondes pour chaque cas de test dans la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.
- Tags (Balises)(facultatif) : Ajoutez des balises à la suite de tests que vous allez créer.



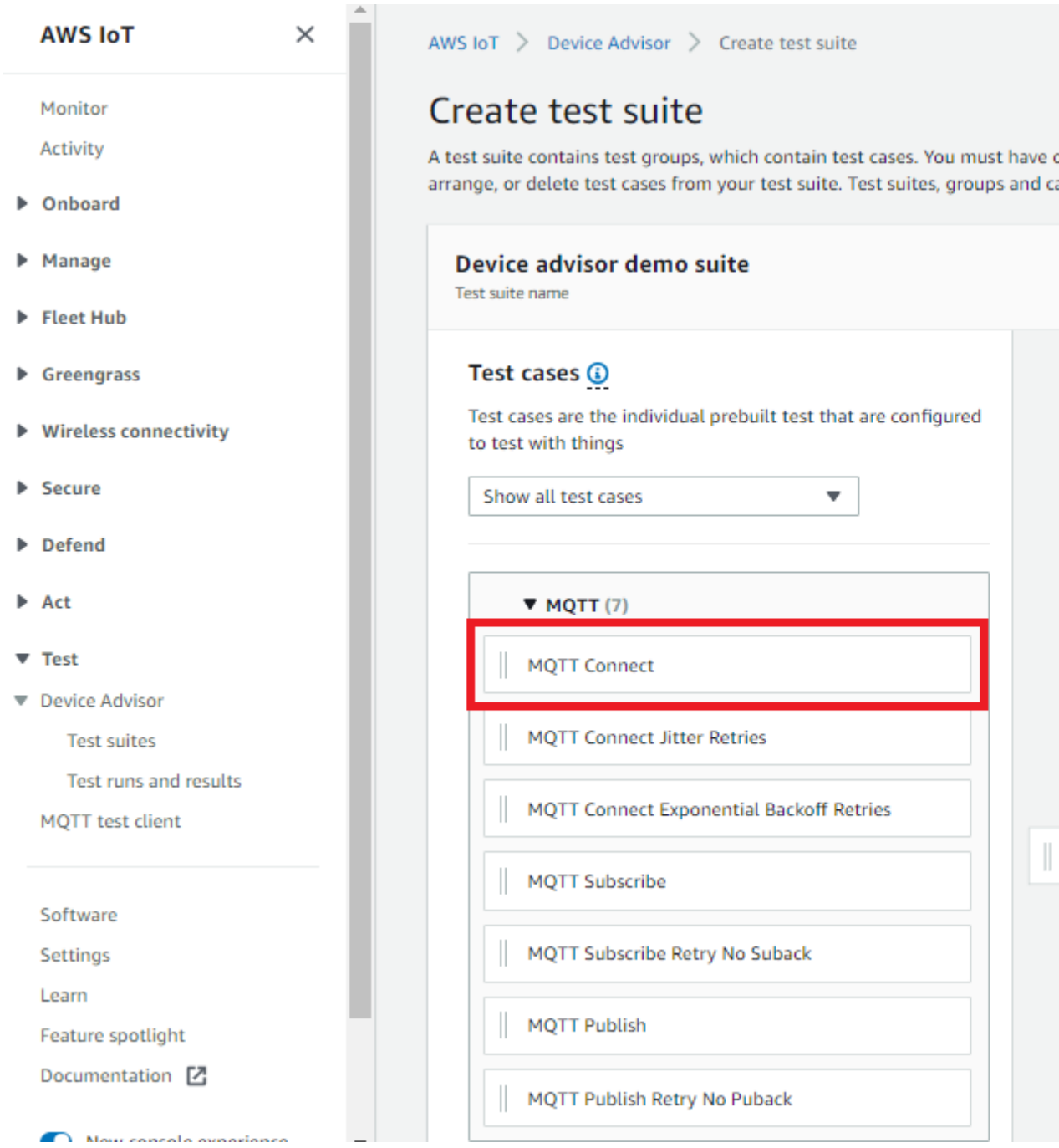
Lorsque vous avez terminé, choisissezMettre à jour les.

4. Pour modifier la configuration au niveau du groupe, sous`Test group 1`, choisissezModifier. Ensuite, saisissez un objetNompour donner au groupe un nom personnalisé. Le cas échéant, vous pouvez également entrer unTimeout (Expiration)en secondes sous le groupe de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.

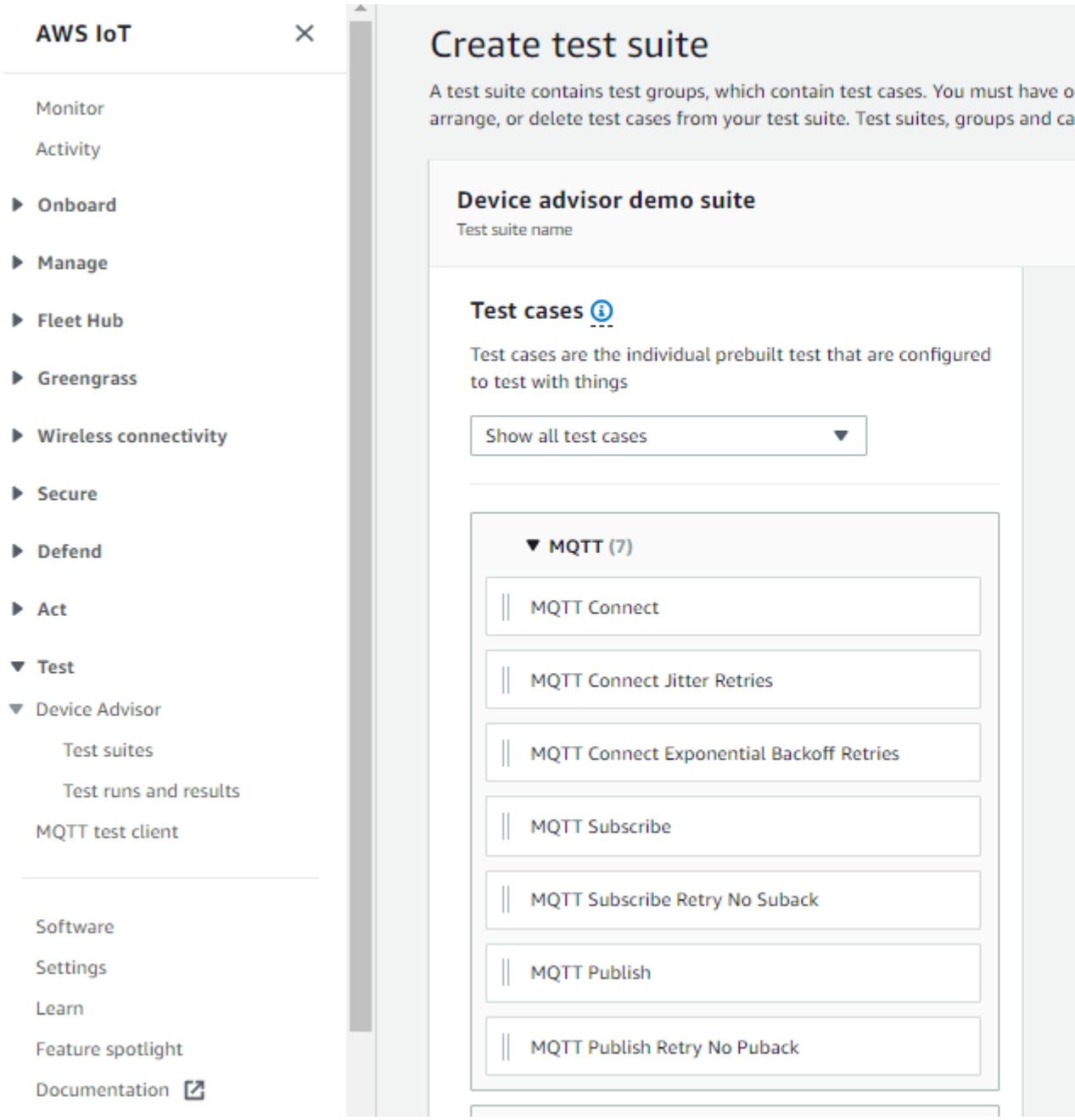


Sélectionnez Done (Effectué).

5. Faites glisser l'un des cas de test disponibles dansCas d'essaidans le groupe test.



6. Pour modifier la configuration de niveau de scénario de test sur le scénario de test sous votre groupe de test, sélectionnezModifier. Ensuite, saisissez un objetNompour donner au groupe un nom personnalisé. Le cas échéant, vous pouvez également entrer unTimeout (Expiration)en secondes sous le groupe de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'expiration, la valeur par défaut est utilisée.



Sélectionnez Done (Effectué).

> **Note**
>
> Pour ajouter d'autres groupes de tests à la suite de tests, choisissezAjouter un groupe test. Suivez les étapes précédentes pour créer et configurer d'autres groupes de test ou pour ajouter d'autres cas de test à un ou plusieurs groupes de test. Les groupes de test et les cas de test peuvent être réorganisés en faisant glisser. Device Advisor exécute des tests dans l'ordre dans lequel vous définissez les groupes de test et les cas de test.

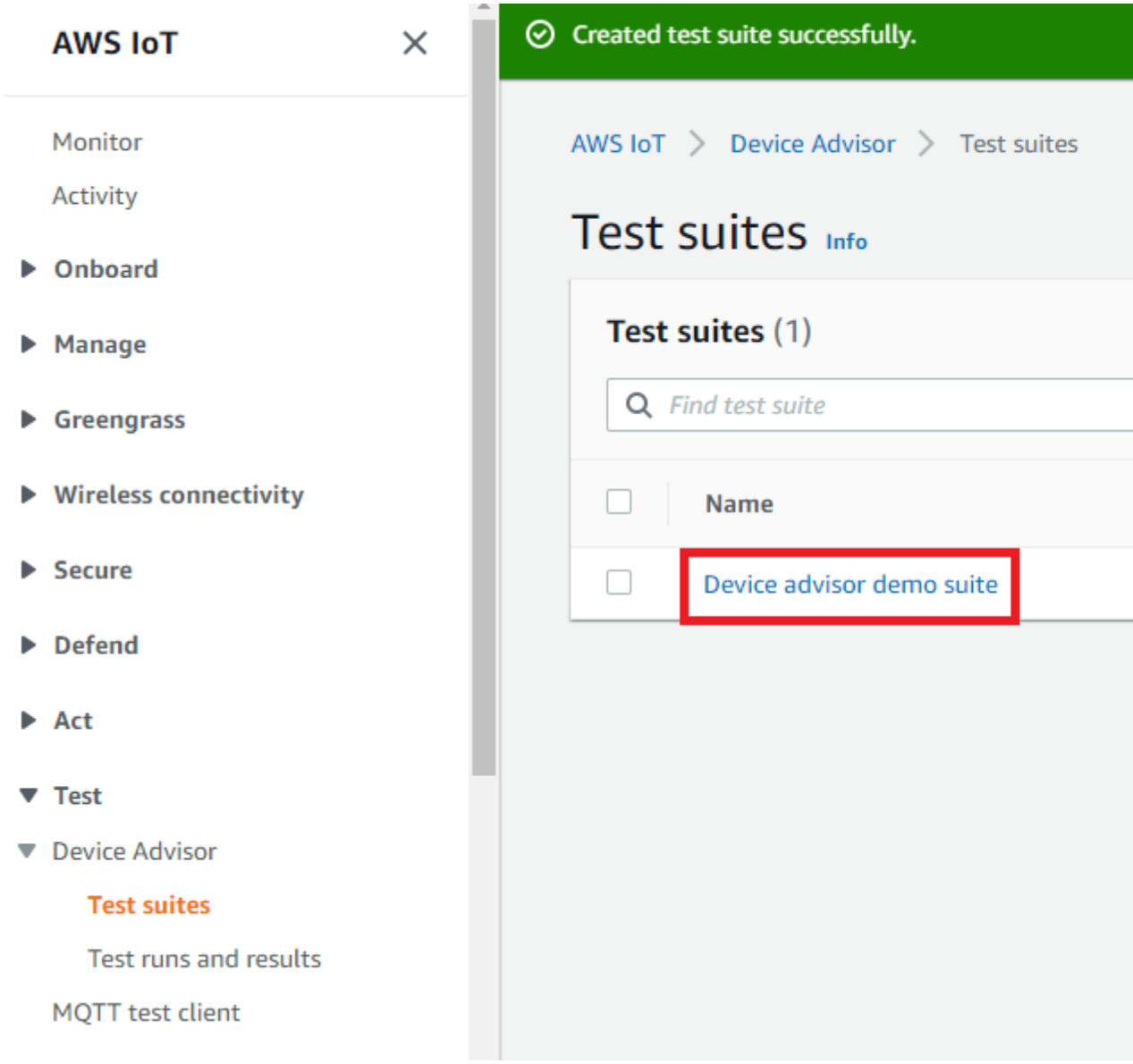
7. ChoisissezCréation d'une suite de tests.

La suite de tests doit être créée avec succès et vous serez redirigé vers leSuites testoù vous pouvez afficher toutes les suites de tests qui ont été créées.

Si la création de la suite de tests a échoué, assurez-vous que la suite de tests, les groupes de tests et les cas de test ont été configurés conformément aux instructions.

Démarrer une exécution de la suite de tests

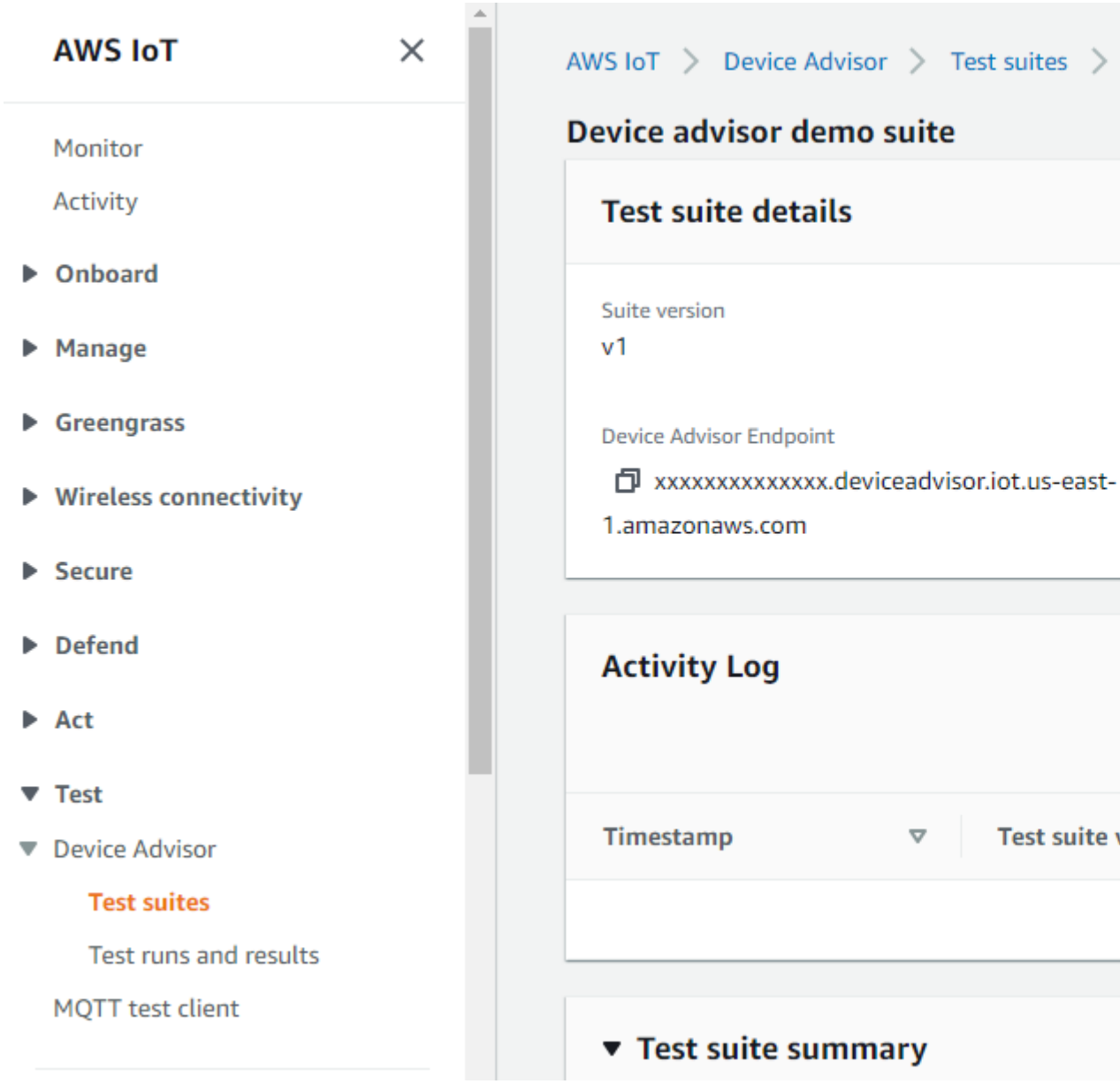
1. DansAWS IoTconsoleDans le volet de navigation, développezTest,Device Advisor, puis.Suites test.
2. Choisissez la suite de tests pour laquelle vous souhaitez afficher les détails de la suite de tests.



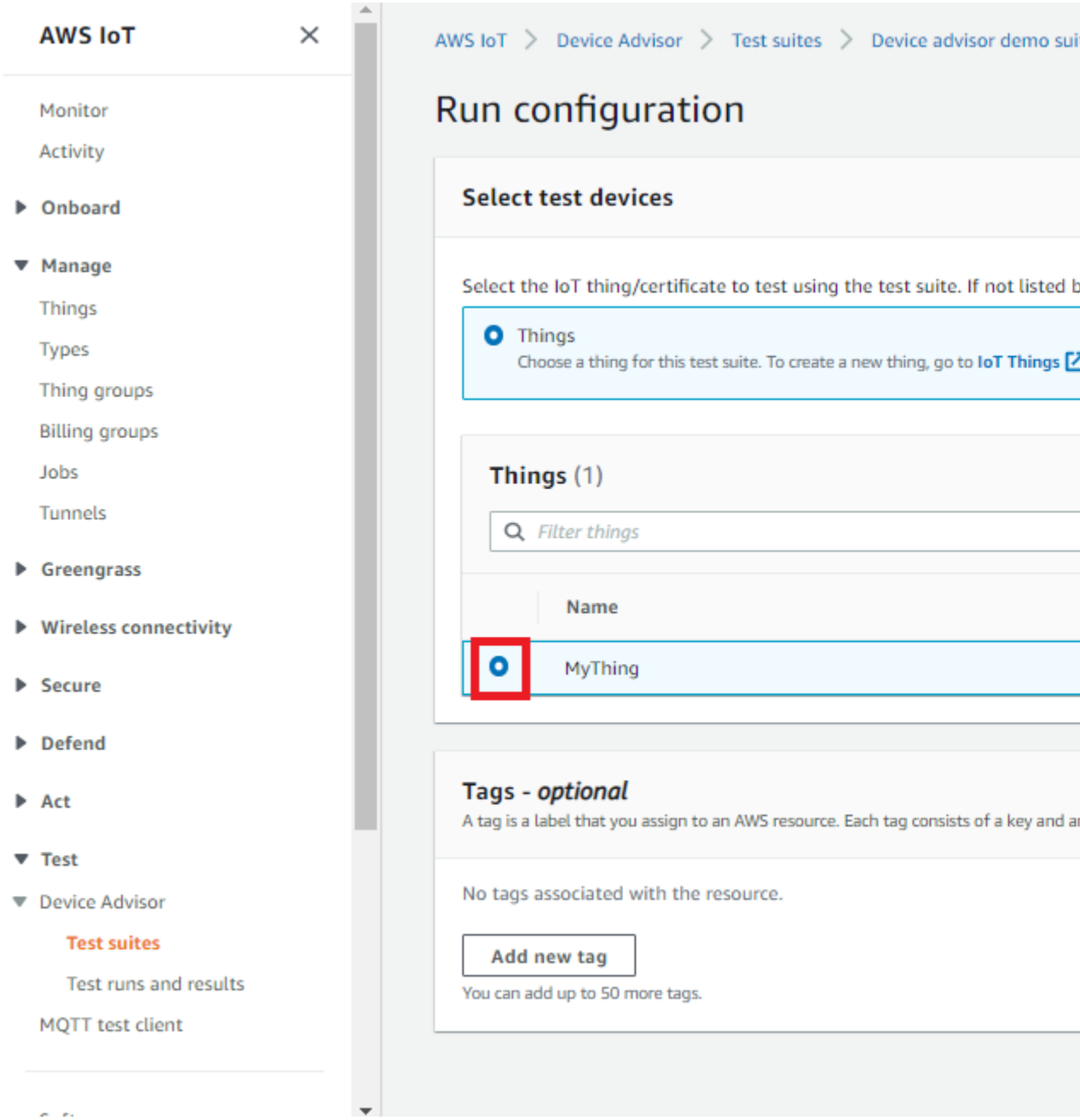
La page détaillée de la suite de tests affiche toutes les informations relatives à la suite de tests. La .Device Advisor Endpointaffiché sur cette page peut être utilisé pour configurer le micrologiciel/

logiciel sur l'appareil que vous utiliserez pour les tests afin de se connecter au point de terminaison de test Device Advisor pour votre compte.

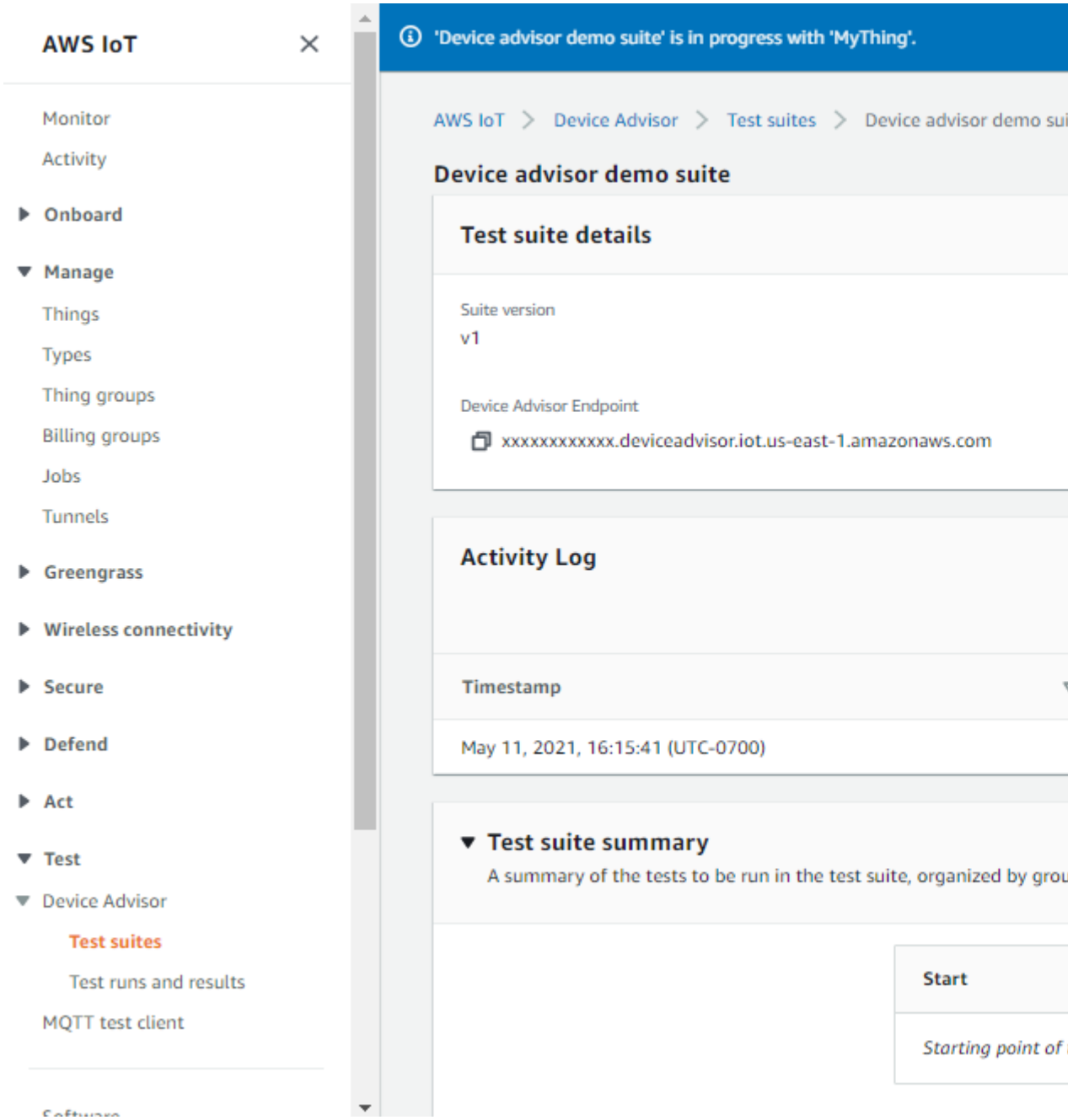
3. ChoisissezActions, puisExécuter la suite de tests.



4. UnderConfiguration d'exécution, vous devez sélectionner unAWS IoTchose ou certificat à tester à l'aide de Device Advisor. Si vous n'avez pas de choses ou de certificats existants, d'abordcreate AWS IoT Core resources (p. 998). Après avoir sélectionné un objet ou un certificat, choisissezExécution de tests.

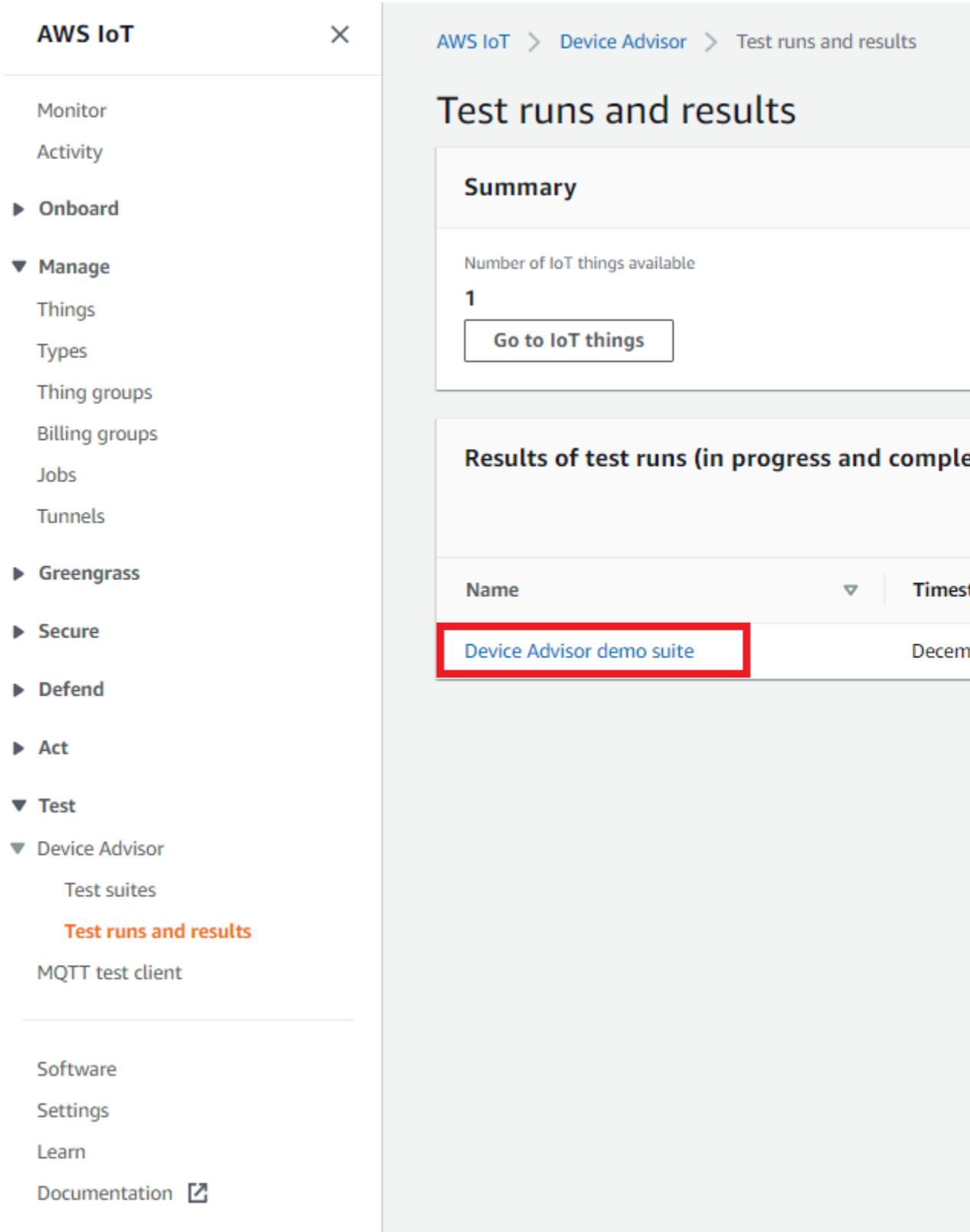


5. ChoisissezAccédez aux résultatssur la bannière supérieure pour afficher les détails de l'exécution de test.

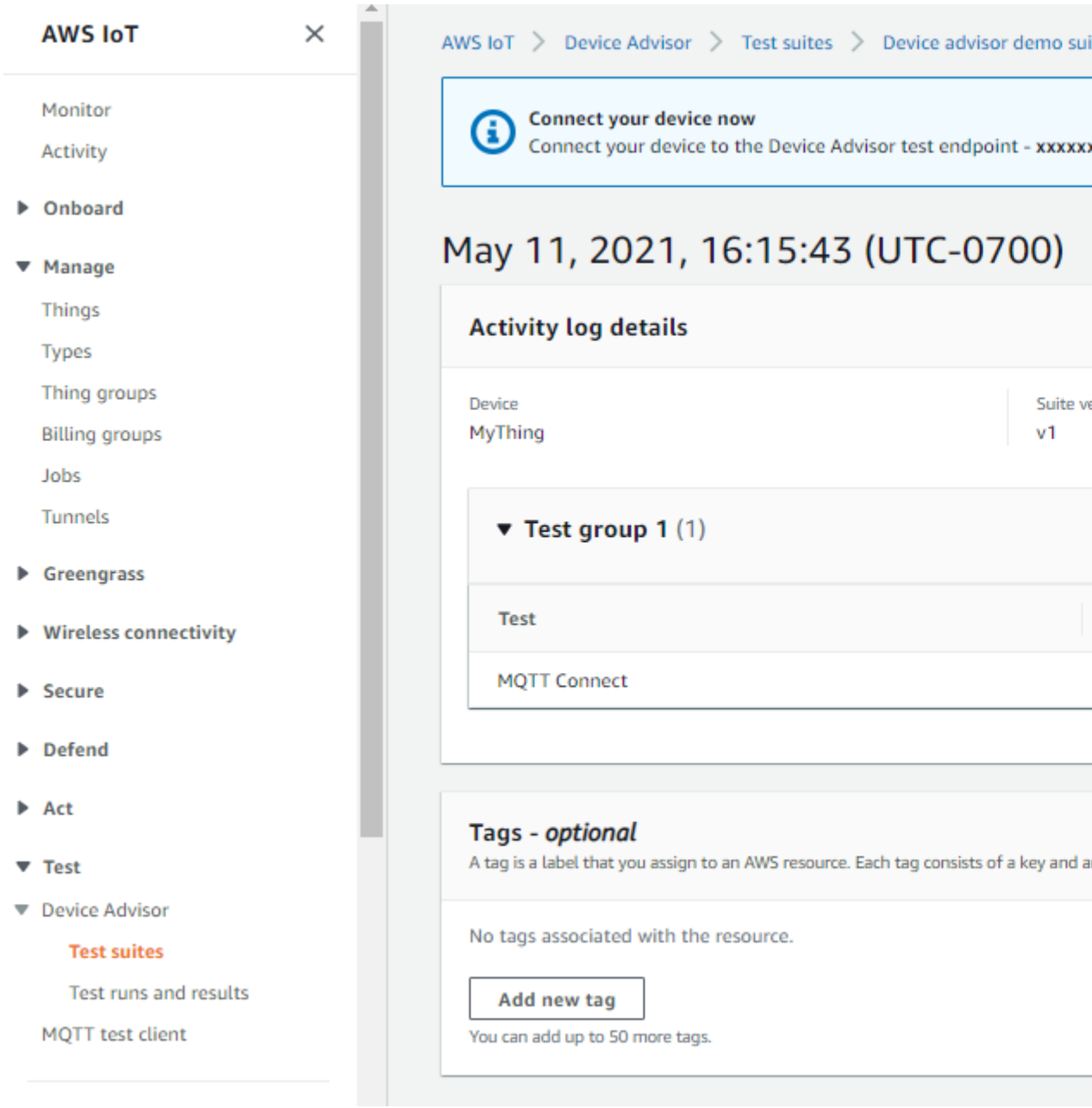


Arrêter l'exécution d'une suite de tests (facultatif)

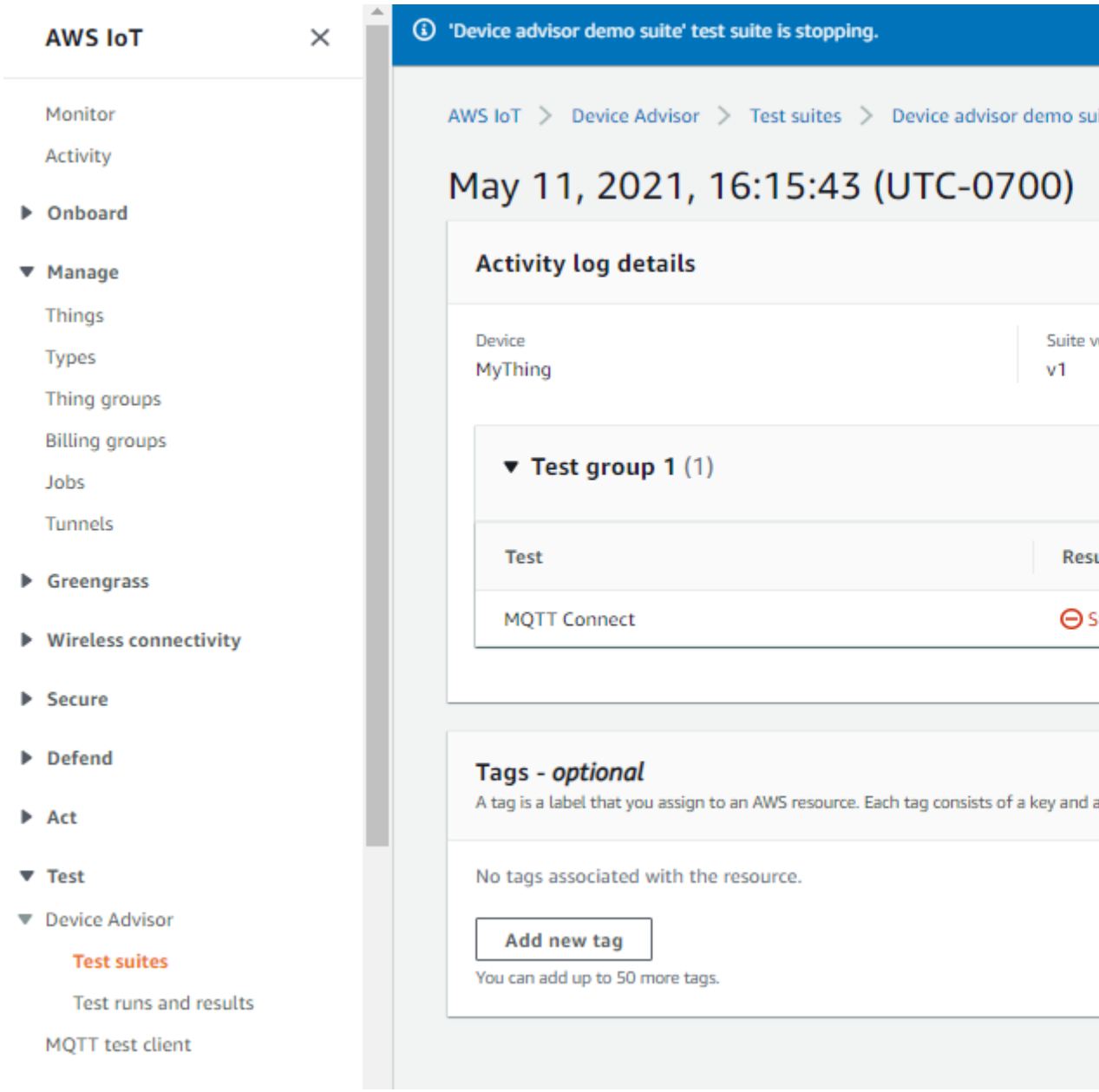
1. DansAWS IoTconsoleDans le volet de navigation, développezTest,Device Advisor, puis.Tests et résultats.
2. Choisissez la suite de tests en cours que vous souhaitez arrêter.



3. ChoisissezActions, puisArrêter Suite test.



4. Le processus de nettoyage prend plusieurs minutes. Pendant que le processus de nettoyage se déroule, l'état d'exécution du test sera`STOPPING`. Attendez que le processus de nettoyage se termine et que l'état de la suite de tests passe à`STOPPED`avant de commencer une nouvelle exécution de suite.



Afficher les détails et les journaux de l'exécution de la suite

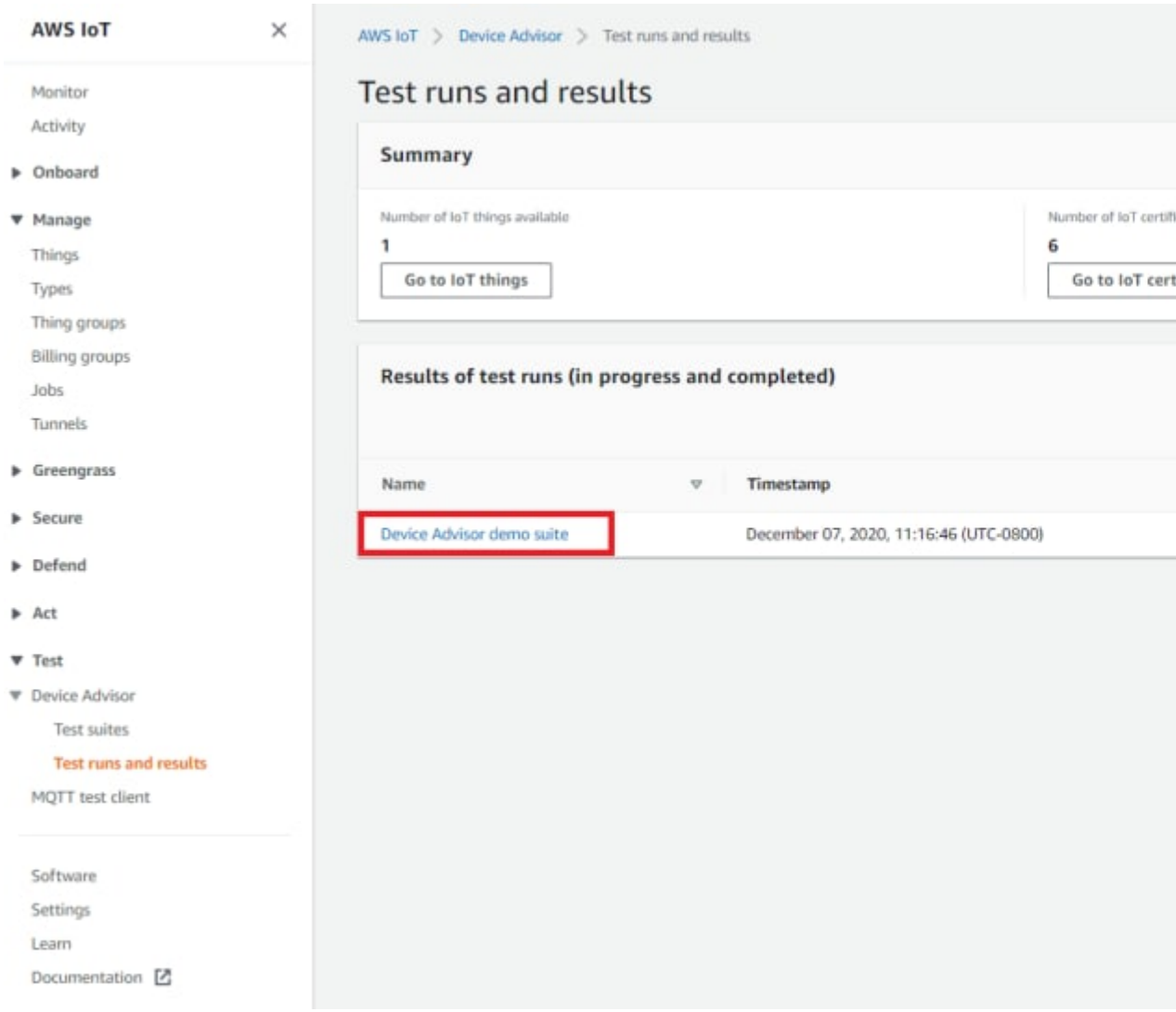
1. DansAWS IoTconsoleDans le volet de navigation, développezTest,Device Advisor, puis.Tests et résultats.

   Cette page affiche :

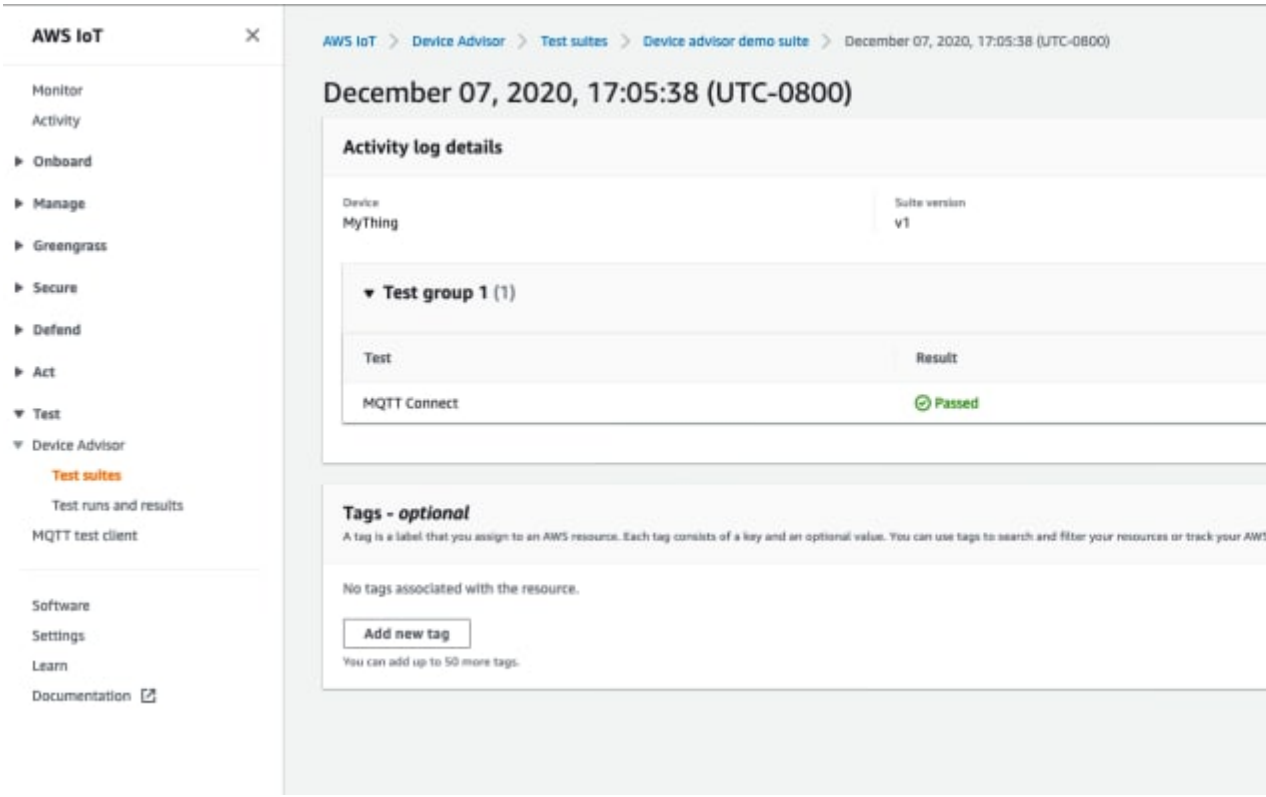
   - Nombre de choses IoT
   - Nombre de certificats IoT

- Nombre de suites de test en cours d'exécution
- Toutes les exécutions de la suite de tests qui ont été créées

2. Choisissez la suite de tests pour laquelle vous souhaitez afficher les détails et les journaux d'exécution.



La page récapitulatif de l'exécution affiche l'état de l'exécution de la suite de tests en cours. Cette page s'actualise automatiquement toutes les 10 secondes. Nous vous recommandons de disposer d'un mécanisme conçu pour que votre appareil essaie de se connecter à notre point de terminaison de test toutes les cinq secondes pendant une à deux minutes. Ensuite, vous pouvez exécuter plusieurs cas de test en séquence de manière automatisée.

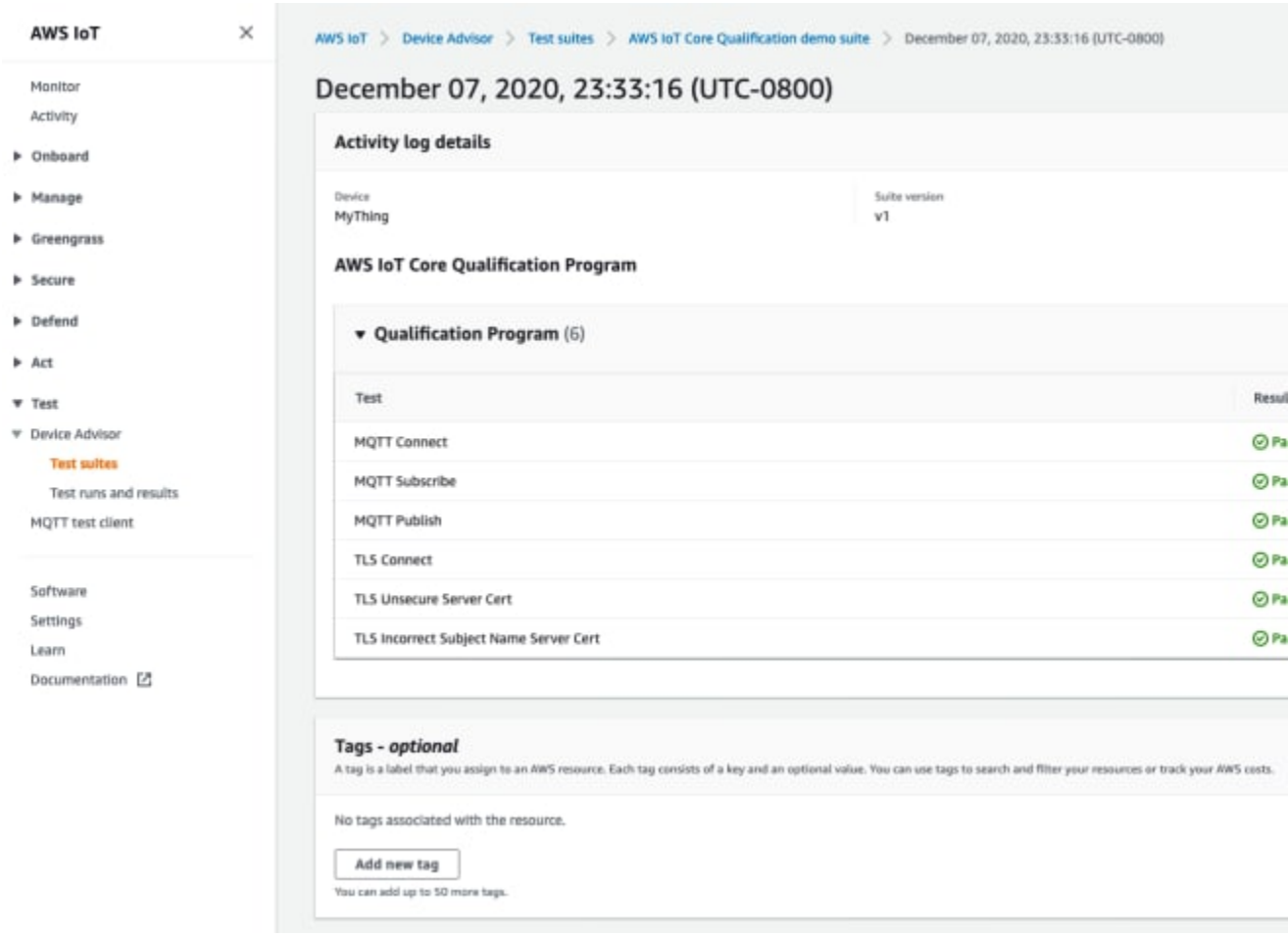


3. Pour accéder aux journaux CloudWatch pour l'exécution de la suite de tests, choisissezJournal de la suite de tests.

   Pour accéder aux journaux CloudWatch pour n'importe quel cas de test, choisissezJournal des cas de test.
4. Sur la base de vos résultats de test,DépannageVotre appareil jusqu'à ce que tous les tests soient passés.

Télécharger unAWS IoTRapport de qualification

Si vous avez choisiUtilisation de l'AWS IoTSuite test de qualificationlors de la création d'une suite de tests et ont pu exécuter une suite de tests de qualification, vous pouvez télécharger un rapport de qualification en sélectionnantTéléchargement du rapport de qualificationdans la page récapitulative de l'exécution du test.



Device Advisor

Device Advisor fournit des tests prédéfinis dans cinq catégories.

- TLS
- Autorisations et stratégies
- MQTT
- Shadow
- Exécution de Job

> **Note**
>
> Votre appareil doit réussir les tests de qualification suivants et être en mesure de répertorier votre appareil dans leAWSCatalogue d'appareils partenaires :
>
> - TLS Nom de sujet incorrect Cert serveur(« Nom commun du sujet incorrect (CN)/Nom alternatif du sujet (SAN) »)
> - Cert de serveur non sécurisé TLS(« Non signé par une autorité de certification reconnue »)
> - Connect TLS(« TLS Connect »)
> - Connect MQTT(« L'appareil envoie CONNECT à AWS IoT Core (Happy cas) »)

- S'abonner à MQTT(« Peut s'abonner (Happy Case) »)
- Publier MQTT(« QoS0 (Happy Case) »)

TLS

Vous utilisez ces tests pour déterminer si le protocole TLS (Transport Layer Security Protocol) entre vos appareils etAWS IoTest sécurisé.

Suites de chiffrement

« TLS Connect »

Valide si le périphérique testé peut terminer la poignée de main TLS àAWS IoT. Ce test ne valide pas l'implémentation MQTT de la machine cliente.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous avons recommandé une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_tls_connect_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  //in seconds
      },
      "test":{
         "id":"TLS_Connect",
         "version":"0.0.0"
      }
   }
]
```

Sorties du cas de test :

Pass : Le périphérique testé a terminé la poignée de main TLS avecAWS IoT.

Passer avec avertissements : Le périphérique testé a terminé la poignée de main TLS avecAWS IoT, mais il y avait des messages d'avertissement TLS à partir de l'appareil ouAWS IoT.

fail : Le périphérique testé n'a pas réussi à terminer la poignée de main TLS avecAWS IoTen raison d'une erreur de poignée de main.

« Prise en Support des périphériques TLS pourAWS IoTSuites de chiffrement recommandées »

Valide que les suites de chiffrement dans le message TLS Client Hello du périphérique sous test contiennentAWS IoTSuites de chiffrement recommandées (p. 313). Il fournit des informations supplémentaires sur les suites de chiffrement prises en charge par l'appareil.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
```

```
   {
      "name":"my_tls_support_aws_iot_cipher_suites_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
      },
      "test":{
         "id":"TLS_Support_AWS_IoT_Cipher_Suites",
         "version":"0.0.0"
      }
   }
]
```

Sorties du cas de test :

Pass : Les suites de chiffrement du périphérique sous test contiennent au moins unAWS IoTrecommandé et ne contiennent aucune suite de chiffrement non prise en charge.

Passer avec avertissements : Les suites de chiffrement de l'appareil contiennent au moins unAWS IoTmais 1) ne contiennent aucune des suites de chiffrement recommandées, ou 2) contiennent des suites de chiffrement non supportées parAWS IoT. Nous suggérons de vérifier que les suites de chiffrement non prises en charge sont sûres.

fail : Le périphérique sous les suites de chiffrement de test ne contient aucune desAWS IoTSuites de chiffrement prises en charge.

Mauvais certificat de serveur

« Non signé par une autorité de certification reconnue »

Valide que le périphérique testé ferme la connexion s'il est présenté avec un certificat de serveur qui n'a pas de signature valide de l'autorité de certification ATS. Un périphérique doit se connecter uniquement à un point de terminaison qui présente un certificat valide.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_tls_unsecure_server_cert_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",  //in seconds
      },
      "test":{
         "id":"TLS_Unsecure_Server_Cert",
         "version":"0.0.0"
      }
   }
]
```

Sorties du cas de test :

Pass : L'appareil testé a fermé la connexion.

fail : Le périphérique testé a terminé la poignée de main TLS avecAWS IoT.

« Nom commun du sujet incorrect (CN)/Nom alternatif du sujet (SAN) »

Valide que le périphérique testé ferme la connexion s'il est présenté avec un certificat de serveur pour un nom de domaine différent de celui demandé.

Définition du cas de test de l'API :

> Note
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_tls_incorrect_subject_name_cert_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",   // in seconds
      },
      "test":{
         "id":"TLS_Incorrect_Subject_Name_Server_Cert",
         "version":"0.0.0"
      }
   }
]
```

Sorties du cas de test :

Pass : L'appareil testé a fermé la connexion.

fail : Le périphérique testé a terminé la poignée de main TLS avecAWS IoT.

Autorisations et stratégies

Vous utilisez ces tests pour déterminer si les stratégies associées aux certificats de vos appareils suivent les meilleures pratiques.

« Les stratégies attachées au certificat de périphérique ne contiennent pas de caractères génériques »

Valide si les stratégies d'autorisation associées à un appareil suivent les meilleures pratiques et n'accordent pas au périphérique plus d'autorisations que nécessaire.

Définition du cas de test de l'API :

> Note
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 1 minute. Il est recommandé de définir le délai d'attente pendant au moins 30 secondes.

```
"tests":[
   {
        "name":"my_security_device_policies"
        "configuration": {
            // optional:
            "EXECUTION_TIMEOUT":"60"    // in seconds
        }
        "test": {
            "id": "Security_Device_Policies",
            "version": "0.0.0"
```

```
        }
    }
]
```

MQTT

CONNECTER, DÉCONNECTER et RECONN

« L'appareil envoie CONNECT à AWS IoT Core (Happy cas) »

Valide que le périphérique testé envoie une requête CONNECT.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_mqtt_connect_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300",   // in seconds
      },
      "test":{
         "id":"MQTT_Connect",
         "version":"0.0.0"
      }
   }
]
```

« Reconnexion de l'appareil avec backoff de gigue - Pas de réponse CONNACK »

Valide que l'appareil testé utilise le backoff de gigue approprié lors de la reconnexion avec le courtier pendant au moins cinq fois. Le broker enregistre l'horodatage du périphérique sous la demande CONNECT du test, effectue la validation des paquets, interrompt sans envoyer de CONNACK au périphérique testé et attend que le périphérique testé renvoie la demande. La sixième tentative de connexion est autorisée à passer et CONNACK est autorisé à revenir à l'appareil soumis à l'essai.

Le processus ci-dessus est effectué à nouveau. Au total, ce cas de test nécessite que l'appareil se connecte au moins 12 fois au total. Les horodatages collectés sont utilisés pour valider que la jitter backoff est utilisée par le périphérique testé. Si l'appareil soumis à l'essai a un délai de retour strictement exponentiel, ce cas de test passera avec des avertissements.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 10 minutes. Nous recommandons une valeur de délai d'expiration de 4 minutes.

```
"tests":[
   {
      "name":"my_mqtt_jitter_backoff_retries_test"
      "configuration": {
         // optional:
```

```
         "EXECUTION_TIMEOUT":"300",    // in seconds
      },
      "test":{
         "id":"MQTT_Connect_Jitter_Backoff_Retries",
         "version":"0.0.0"
      }
   }
]
```

« L'appareil peut renvoyer PUBACK à un sujet arbitraire pour QOS1"

Ce cas de test vérifiera si le périphérique (client) peut renvoyer le message PUBACK s'il a reçu un message de publication du broker après s'être abonné à une rubrique avec QoS1.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
        "name":"my_mqtt_client_puback_qos1"
        "configuration": {
            // optional:
            "TRIGGER_TOPIC": "myTopic",
            "EXECUTION_TIMEOUT":"300", // in seconds
            "PAYLOAD_FOR_PUBLISH_VALIDATION":"custom payload"
        }
        "test": {
            "id": "MQTT_Client_Puback_Qos1",
            "version": "0.0.0"
        }
    }
]
```

« Reconnexion de l'appareil avec backoff exponentielle - Pas de réponse CONNACK »

Valide que le périphérique testé utilise le backoff exponentiel approprié lors de la reconnexion avec le broker pendant au moins cinq fois. Le broker enregistre l'horodatage du périphérique sous la demande CONNECT du test, effectue la validation des paquets, interrompt sans envoyer de CONNACK à la machine cliente et attend que le périphérique testé renvoie la demande. Les horodatages collectés sont utilisés pour valider que le périphérique testé utilise le backoff exponentiel.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 10 minutes. Nous recommandons une valeur de délai d'expiration de 4 minutes.

```
"tests":[
   {
      "name":"my_mqtt_exponential_backoff_retries_test"
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"600",  // in seconds
      },
      "test":{
         "id":"MQTT_Connect_Exponential_Backoff_Retries",
         "version":"0.0.0"
```

```
      }
   }
]
```

Publish

« QoS0 (Happy Case) »

Valide que le périphérique testé publie un message avec QoS0. Vous pouvez également valider la rubrique du message en spécifiant cette valeur dans les paramètres de test.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_mqtt_publish_test":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_FOR_PUBLISH_VALIDATION": "my_TOPIC_FOR_PUBLISH_VALIDATION",
         "PAYLOAD_FOR_PUBLISH_VALIDATION": "my_PAYLOAD_FOR_PUBLISH_VALIDATION",
      },
      "test":{
         "id":"MQTT_Publish",
         "version":"0.0.0"
      }
   }
]
```

« QOS1 publier une nouvelle tentative - Pas de PUBACK »

Valide que le périphérique testé republie un message envoyé avec QOS1, si le broker n'envoie pas PUBACK. Vous pouvez également valider la rubrique du message en spécifiant cette rubrique dans les paramètres de test. Le périphérique client ne doit pas se déconnecter avant de republier le message. Ce test valide également que le message republié a le même identifiant de paquet que l'original.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Il est recommandé pendant au moins 4 minutes.

```
"tests":[
   {
      "name":"my_mqtt_publish_retry_test":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_FOR_PUBLISH_VALIDATION": "my_TOPIC_FOR_PUBLISH_VALIDATION",
         "PAYLOAD_FOR_PUBLISH_VALIDATION": "my_PAYLOAD_FOR_PUBLISH_VALIDATION",
      },
      "test":{
         "id":"MQTT_Publish_Retry_No_Puback",
         "version":"0.0.0"
      }
   }
```

```
]
```

Subscribe

« Peut s'abonner (Happy Case) »

Valide que le périphérique sous test est abonné aux rubriques MQTT. Vous pouvez également valider la rubrique à laquelle le périphérique sous test est abonné en spécifiant cette rubrique dans les paramètres de test.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_mqtt_subscribe_test":{
         // optional:
         "EXECUTION_TIMEOUT":"300",  // in seconds
         "TOPIC_LIST_FOR_SUBSCRIPTION_VALIDATION_ID":
["my_TOPIC_FOR_PUBLISH_VALIDATION_a","my_TOPIC_FOR_PUBLISH_VALIDATION_b"]
      },
      "test":{
         "id":"MQTT_Subscribe",
         "version":"0.0.0"
      }
   }
]
```

« Réessayer de s'abonner - Pas de SUBACK »

Valide que le périphérique testé tente de nouveau un échec d'abonnement aux rubriques MQTT. Le serveur attend alors et n'envoie pas de SUBACK. Si la machine cliente ne réessaie pas l'abonnement, le test échoue. Vous pouvez également valider la rubrique à laquelle le périphérique sous test est abonné en spécifiant cette rubrique dans les paramètres de test.

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 4 minutes.

```
"tests":[
   {
      "name":"my_mqtt_subscribe_retry_test":{
         "EXECUTION_TIMEOUT":"300",  // in seconds
         // optional:
         "TOPIC_LIST_FOR_SUBSCRIPTION_VALIDATION_ID":
["myTOPIC_FOR_PUBLISH_VALIDATION_a","my_TOPIC_FOR_PUBLISH_VALIDATION_b"]
      },
      "test":{
         "id":"MQTT_Subscribe_Retry_No_Suback",
         "version":"0.0.0"
      }
   }
]
```

Shadow

Utilisez ces tests pour vérifier l'utilisation de vos appareils testésAWS IoTLe service Device Shadow correctement. Pour plus d'informations, consultez Service AWS IoT Device Shadow (p. 547). Si ces cas de test sont configurés dans votre suite de tests, vous devez fournir une chose lors du démarrage de l'exécution de la suite.

Publish

« L'appareil publie l'état après qu'il se connecte (cas heureux) »

Valide si un périphérique peut publier son état après sa connexion à AWS IoT Core

Définition du cas de test de l'API :

> **Note**
>
> `EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_shadow_publish_reported_state",
      "configuration": {
         // optional:
         "EXECUTION_TIMEOUT":"300", // in seconds
         "SHADOW_NAME": "SHADOW_NAME",
         "REPORTED_STATE": {
            "STATE_ATTRIBUTE": "STATE_VALUE"
         }
      },
      "test":{
         "id":"Shadow_Publish_Reported_State",
         "version":"0.0.0"
      }
   }
]
```

La .`REPORTED_STATE`peut être fourni pour une validation supplémentaire sur l'état exact de l'ombre de votre appareil, après qu'il se connecte. Par défaut, ce cas de test valide l'état de publication de votre appareil.

Si*`SHADOW_NAME`*n'est pas fourni, le cas de test recherche les messages publiés dans les préfixes de rubrique du type d'ombre Unnamed (classique) par défaut. Indiquez un nom d'ombre si votre appareil utilise le type d'ombre nommé. VoirUtilisation des shadows sur les appareilspour en savoir plus.

Update

« Les mises à jour de l'appareil ont signalé l'état souhaité (cas heureux) »

Valide si votre appareil lit tous les messages de mise à jour reçus et synchronise l'état de l'appareil pour qu'il corresponde aux propriétés d'état souhaitées. Votre appareil doit publier son dernier état signalé après la synchronisation. Si votre appareil dispose déjà d'une ombre avant d'exécuter le test, assurez-vous que l'état souhaité configuré pour le cas de test et l'état signalé existant ne correspondent pas déjà. Vous pouvez identifier les messages de mise à jour Shadow envoyés par Device Advisor en regardant le champ ClientToken dans le document Shadow tel qu'il sera`DeviceAdvisorShadowTestCaseSetup`.

Définition du cas de test de l'API :

Note

`EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
      "name":"my_shadow_update_reported_state",
      "configuration": {
         "DESIRED_STATE": {
            "STATE_ATTRIBUTE": "STATE_VALUE"
         },
         // optional:
         "EXECUTION_TIMEOUT":"300", // in seconds
         "SHADOW_NAME": "SHADOW_NAME"
      },
      "test":{
         "id":"Shadow_Update_Reported_State",
         "version":"0.0.0"
      }
   }
]
```

La .`DESIRED_STATE`doit avoir au moins un attribut et une valeur associée.

Si`SHADOW_NAME`n'est pas fourni, le cas de test recherche les messages publiés dans les préfixes de rubrique du type d'ombre Unnamed (classique) par défaut. Indiquez un nom d'ombre si votre appareil utilise le type d'ombre nommé. VoirUtilisation des shadows sur les appareilspour en savoir plus.

Exécution de Job

« L'appareil peut terminer l'exécution d'une tâche »

Ce cas de test vous aide à valider si votre appareil est capable de recevoir des mises à jour à l'aide deAWS IoTTâches et publiez l'état de la mise à jour réussie. Pour plus d'informations surAWS IoTEmplois, voirTâches.

Définition du cas de test de l'API :

Note

`EXECUTION_TIMEOUT`a pour valeur par défaut 5 minutes. Nous recommandons une valeur de délai d'expiration de 2 minutes.

```
"tests":[
   {
        "name":"my_job_execution",
        "configuration": {
            // Document is a JSON formatted string
            "JOB_DOCUMENT": "{
                "operation":"reboot",
                "files" : {
                   "fileName" : "install.py",
                   "url" : "${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/
key}"
            }",

            // Document_SOURCE is an S3 link to the job document
            // Document and document are optional but can't be both null.
```

```
            "JOB_DOCUMENT_SOURCE": "https://s3.amazonaws.com/bucket/key",
            // optional:
            "EXECUTION_TIMEOUT": "300"  // in seconds
        },
        "test": {
            "id": "Job_Execution",
            "version": "0.0.0"
        }
    }
]
```

Pour plus d'informations sur la création et l'utilisation de documents de tâche, consultezdocument de tâche.

Messages d'événements

Cette section contient des informations sur les messages publiés par AWS IoT lorsque des objets ou des tâches sont mis à jour ou modifiés. Pour obtenir des informations sur leAWS IoT Eventsqui vous permet de créer des détecteurs capables de surveiller vos appareils en cas de défaillances ou de modifications dans le fonctionnement, ainsi que de déclencher des actions lorsqu'elles surviennent, consultez.AWS IoT Events.

AWS IoT publie des messages d'événements lorsque certains événements se produisent. Par exemple, le registre génère des événements quand des objets sont ajoutés, mis à jour ou supprimés. Chaque événement génère l'envoi d'un seul message. Les messages d'événements sont publiés sur MQTT avec une charge utile JSON. Le contenu de la charge utile dépend du type d'événement.

> **Note**
>
> Il est garanti que les messages d'événements sont publiés une fois. Ils peuvent être publiés plusieurs fois. L'ordre des messages d'événement n'est pas garanti.

Afin de recevoir des messages d'événement, votre appareil doit utiliser une stratégie appropriée qui lui permet de se connecter à la passerelle de l'appareil AWS IoT et de s'abonner aux rubriques d'événements MQTT. Vous devez aussi vous abonner aux filtres de rubriques appropriés.

Voici un exemple de stratégie requise pour recevoir des événements de cycle de vie :

```
{
    "Version":"2012-10-17",
    "Statement":[{
        "Effect":"Allow",
        "Action":[
            "iot:Subscribe",
            "iot:Receive"
        ],
        "Resource":[
            "arn:aws:iot:region:account:/$aws/events/*"
        ]
    }]
}
```

Vous contrôlez les types d'événements publiés en appelant l'API UpdateEventConfigurations ou en utilisant la commande update-event-configurations de l'interface de ligne de commande (CLI). Exemples :

aws iot update-event-configurations --event-configurations "{\"THING\":{\"Enabled\": true}}"

> **Note**
>
> L'échappement de tous les guillemets (") est effectué avec des barres obliques inverses (\).

Vous pouvez obtenir la configuration de l'événement en cours en appelant l'API DescribeEventConfigurations ou en utilisant la commande describe-event-configurations de l'interface de ligne de commande (CLI). Par exemple :

aws iot describe-event-configurations

La sortie de la commande describe-event-configurations ressemble à ce qui suit :

```
{
    "lastModifiedDate": 1552671347.841,
    "eventConfigurations": {
```

```
        "THING_TYPE": {
            "Enabled": false
        },
        "JOB_EXECUTION": {
            "Enabled": false
        },
        "THING_GROUP_HIERARCHY": {
            "Enabled": false
        },
        "THING_TYPE_ASSOCIATION": {
            "Enabled": false
        },
        "THING_GROUP_MEMBERSHIP": {
            "Enabled": false
        },
        "THING": {
            "Enabled": true
        },
        "JOB": {
            "Enabled": false
        },
        "THING_GROUP": {
            "Enabled": false
        }
    },
    "creationDate": 1552671347.84
}
```

Événements de registre

Le registre publie des messages d'événement lorsque des objets, types d'objets et groupes d'objets sont créés, mis à jour ou supprimés. Le registre prend actuellement en charge les types d'événements suivants :

Objet créé/mis à jour/supprimé

Le registre publie les messages d'événement suivants lorsque des objets sont créés, mis à jour ou supprimés :

- `$aws/events/thing/`*`thingName`*`/created`
- `$aws/events/thing/`*`thingName`*`/updated`
- `$aws/events/thing/`*`thingName`*`/deleted`

Les messages contiennent l'exemple de charge utile suivant :

```
{
    "eventType" : "THING_EVENT",
    "eventId" : "f5ae9b94-8b8e-4d8e-8c8f-b3266dd89853",
    "timestamp" : 1234567890123,
    "operation" : "CREATED|UPDATED|DELETED",
    "accountId" : "123456789012",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "thingName" : "MyThing",
    "versionNumber" : 1,
    "thingTypeName" : null,
    "attributes": {
                "attribute3": "value3",
                "attribute1": "value1",
                "attribute2": "value2"
    }
}
```

Les charges utiles contiennent les attributs suivants :

eventType

Défini sur « THING_EVENT ».

eventId

Un ID d'événement unique (chaîne).

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

fonctionnement

L'opération qui a déclenché l'événement. Les valeurs valides sont :

- CRÉÉ
- MIS À JOUR
- SUPPRIMEE

accountId

Vos produits Compte AWS ID.

thingId

L'ID de l'objet en cours de création, de mise à jour ou de suppression.

thingName

Le nom de l'objet en cours de création, de mise à jour ou de suppression.

versionNumber

La version de l'objet en cours de création, de mise à jour ou de suppression. Cette valeur est définie sur 1 lors de la création d'un objet. Elle augmente de 1 à chaque mise à jour de l'objet.

thingTypeName

Le type d'objet associé à l'objet, le cas échéant. Sinon la valeur `null` est renvoyée.

attributs

Un ensemble de paires nom-valeur associées à l'objet.

Type d'objet créé/obsolète/dont l'obsolescence est annulée/supprimé

Le registre publie les messages d'événement suivants lorsque des types d'objets sont créés, obsolètes, supprimés ou que leur obsolescence est annulée :

- `$aws/events/thingType/`*`thingTypeName`*`/created`
- `$aws/events/thingType/`*`thingTypeName`*`/updated`
- `$aws/events/thingType/`*`thingTypeName`*`/deleted`

Le message contient l'exemple de charge utile suivant :

```
{
    "eventType" : "THING_TYPE_EVENT",
    "eventId" : "8827376c-4b05-49a3-9b3b-733729df7ed5",
    "timestamp" : 1234567890123,
    "operation" : "CREATED|UPDATED|DELETED",
    "accountId" : "123456789012",
    "thingTypeId" : "c530ae83-32aa-4592-94d3-da29879d1aac",
    "thingTypeName" : "MyThingType",
    "isDeprecated" : false|true,
    "deprecationDate" : null,
    "searchableAttributes" : [ "attribute1", "attribute2", "attribute3" ],
```

```
    "description" : "My thing type"
}
```

Les charges utiles contiennent les attributs suivants :

eventType

Défini sur « THING_TYPE_EVENT ».

eventId

Un ID d'événement unique (chaîne).

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

fonctionnement

L'opération qui a déclenché l'événement. Les valeurs valides sont :

- CRÉÉ
- MIS À JOUR
- SUPPRIMEE

accountId

Vos produits Compte AWS ID.

thingTypeId

L'ID du type d'objet en cours de création, de suppression ou de déclaration d'obsolescence.

thingTypeName

Le nom du type d'objet en cours de création, de suppression ou de déclaration d'obsolescence.

isDeprecated

`true` si le type d'objet est obsolète. Sinon la valeur `false` est renvoyée.

deprecationDate

L'horodatage UNIX associé au moment où ce type d'objet est devenu obsolète.

searchableAttributes

Un ensemble de paires nom-valeur associées au type d'objet qui peut être utilisé pour la recherche.

description

Une description du type d'objet.

Type d'objet associé à un objet/dissocié d'un objet

Le registre publie les messages d'événement suivants lorsqu'un type d'objet est associé à un objet ou dissocié d'un objet.

- `$aws/events/thingTypeAssociation/thing/`*`thingName`*`/`*`typeName`*

Les messages contiennent l'exemple de charge utile suivant :

```
{
    "eventId" : "87f8e095-531c-47b3-aab5-5171364d138d",
    "eventType" : "THING_TYPE_ASSOCIATION_EVENT",
    "operation" : "CREATED|DELETED",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "thingName": "myThing",
    "thingTypeName" : "MyThingType",
```

```
    "timestamp" : 1234567890123,
}
```

Les charges utiles contiennent les attributs suivants :

eventId

Un ID d'événement unique (chaîne).

eventType

Défini sur « THING_TYPE_ASSOCIATION_EVENT ».

fonctionnement

L'opération qui a déclenché l'événement. Les valeurs valides sont :

- CRÉÉ
- SUPPRIMEE

thingId

ID de l'objet dont l'association du type a été modifiée.

thingName

Nom de l'objet dont l'association du type a été modifiée.

thingTypeName

Type d'objet associé à l'objet ou qui n'est plus associé à l'objet.

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

Type d'objet créé/mis à jour/supprimé

Le registre publie les messages d'événement suivants lorsqu'un groupe d'objets est créé, mis à jour ou supprimé.

- `$aws/events/thingGroup/`*`groupName`*`/created`
- `$aws/events/thingGroup/`*`groupName`*`/updated`
- `$aws/events/thingGroup/`*`groupName`*`/deleted`

Voici un exemple de`updated`charge utile. Charges utiles pour`created`and`deleted`Les messages sont similaires.

```
{
  "eventType": "THING_GROUP_EVENT",
  "eventId": "8b9ea8626aeaa1e42100f3f32b975899",
  "timestamp": 1603995417409,
  "operation": "UPDATED",
  "accountId": "571EXAMPLE833",
  "thingGroupId": "8757eec8-bb37-4cca-a6fa-403b003d139f",
  "thingGroupName": "Tg_level5",
  "versionNumber": 3,
  "parentGroupName": "Tg_level4",
  "parentGroupId": "5fce366a-7875-4c0e-870b-79d8d1dce119",
  "description": "New description for Tg_level5",
  "rootToParentThingGroups": [
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/TgTopLevel",
      "groupId": "36aa0482-f80d-4e13-9bff-1c0a75c055f6"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level1",
```

```
      "groupId": "bc1643e1-5a85-4eac-b45a-92509cbe2a77"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level2",
      "groupId": "0476f3d2-9beb-48bb-ae2c-ea8bd6458158"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level3",
      "groupId": "1d9d4ffe-a6b0-48d6-9de6-2e54d1eae78f"
    },
    {
      "groupArn": "arn:aws:iot:us-west-2:571EXAMPLE833:thinggroup/Tg_level4",
      "groupId": "5fce366a-7875-4c0e-870b-79d8d1dce119"
    }
  ],
  "attributes": {
    "attribute1": "value1",
    "attribute3": "value3",
    "attribute2": "value2"
  },
  "dynamicGroupMappingId": null
}
```

Les charges utiles contiennent les attributs suivants :

eventType

Défini sur « THING_GROUP_EVENT ».

eventId

Un ID d'événement unique (chaîne).

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

fonctionnement

L'opération qui a déclenché l'événement. Les valeurs valides sont :

- CRÉÉ
- MIS À JOUR
- SUPPRIMEE

accountId

Vos produits Compte AWS ID.

thingGroupId

L'ID du groupe d'objets en cours de création, de mise à jour ou de suppression.

thingGroupName

Le nom du groupe d'objets en cours de création, de mise à jour ou de suppression.

versionNumber

Version du groupe d'objets. Cette valeur est définie sur 1 lors de la création d'un groupe d'objets. Elle augmente de 1 à chaque mise à jour du groupe d'objets.

parentGroupName

Le nom du groupe d'objets parent (le cas échéant).

parentGroupId

L'ID du groupe d'objets parent (le cas échéant).

description

La description du groupe d'objets.

rootToParentThingGroups

Tableau d'informations sur le groupe d'objets parent. Un élément correspond à chaque groupe d'objets parent, à commencer par le groupe d'objets racine, jusqu'au parent du groupe d'objets. Chaque entrée contient la propriété`groupArn`and`groupId`.

attributs

Un ensemble de paires nom-valeur associées au groupe d'objets.

Objet ajouté à un groupe d'objets/retiré d'un groupe d'objets

Le registre publie les messages d'événement suivants lorsqu'un objet est ajouté à un groupe d'objets ou retiré d'un groupe d'objets.

- `$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/added`
- `$aws/events/thingGroupMembership/thingGroup/`*`thingGroupName`*`/thing/`*`thingName`*`/removed`

Les messages contiennent l'exemple de charge utile suivant :

```
{
    "eventType" : "THING_GROUP_MEMBERSHIP_EVENT",
    "eventId" : "d684bd5f-6f6e-48e1-950c-766ac7f02fd1",
    "timestamp" : 1234567890123,
    "operation" : "ADDED|REMOVED",
    "accountId" : "123456789012",
    "groupArn" : "arn:aws:iot:ap-northeast-2:123456789012:thinggroup/
MyChildThingGroup",
    "groupId" : "06838589-373f-4312-b1f2-53f2192291c4",
    "thingArn" : "arn:aws:iot:ap-northeast-2:123456789012:thing/MyThing",
    "thingId" : "b604f69c-aa9a-4d4a-829e-c480e958a0b5",
    "membershipId" : "8505ebf8-4d32-4286-80e9-c23a4a16bbd8"
}
```

Les charges utiles contiennent les attributs suivants :

eventType

Défini sur « THING_GROUP_MEMBERSHIP_EVENT ».

eventId

L'ID d'événement.

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

fonctionnement

`ADDED` lorsqu' un objet est ajouté à un groupe d'objets. `REMOVED` lorsqu'un objet est supprimé d'un groupe d'objets.

accountId

Vos produits Compte AWS ID.

groupArn

L'ARN du groupe d'objets.

groupId

L'ID du groupe.

thingArn

L'ARN de l'objet qui a été ajouté au groupe d'objets ou supprimé de ce groupe.

thingId

L'ID de l'objet qui a été ajouté au groupe d'objets ou supprimé de ce groupe.

membershipId

Un ID qui représente la relation entre l'objet et le groupe d'objets. Cette valeur est générée lorsque vous ajoutez un objet à un groupe d'objets.

Groupe d'objets ajouté à un groupe d'objets/retiré d'un groupe d'objets

Le registre publie les messages d'événement suivants lorsqu'un groupe d'objets est ajouté à un autre groupe d'objets ou retiré d'un autre groupe d'objets.

- `$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/added`
- `$aws/events/thingGroupHierarchy/thingGroup/`*`parentThingGroupName`*`/childThingGroup/`*`childThingGroupName`*`/removed`

Le message contient l'exemple de charge utile suivant :

```
{
    "eventType" : "THING_GROUP_HIERARCHY_EVENT",
    "eventId" : "264192c7-b573-46ef-ab7b-489fcd47da41",
    "timestamp" : 1234567890123,
    "operation" : "ADDED|REMOVED",
    "accountId" : "123456789012",
    "thingGroupId" : "8f82a106-6b1d-4331-8984-a84db5f6f8cb",
    "thingGroupName" : "MyRootThingGroup",
    "childGroupId" : "06838589-373f-4312-b1f2-53f2192291c4",
    "childGroupName" : "MyChildThingGroup"
}
```

Les charges utiles contiennent les attributs suivants :

eventType

Défini sur « THING_GROUP_HIERARCHY_EVENT ».

eventId

L'ID d'événement.

timestamp

L'horodatage UNIX du moment où l'événement s'est produit.

fonctionnement

`ADDED` lorsqu' un objet est ajouté à un groupe d'objets. `REMOVED` lorsqu'un objet est supprimé d'un groupe d'objets.

accountId

Vos produits Compte AWS ID.

thingGroupId

L'ID du groupe d'objets parent.

thingGroupName

Le nom du groupe d'objets parent.

childGroupId

L'ID du groupe d'objets enfant.

childGroupName

Le nom du groupe d'objets enfant.

Événements Jobs

Le service AWS IoT Jobs publie dans des rubriques réservées du protocole MQTT lorsque des tâches sont en cours, terminées ou annulées, et lorsqu'un appareil indique la réussite ou l'échec de l'exécution d'une tâche. Les appareils ou les applications de gestion et de surveillance peuvent effectuer le suivi de l'état des tâches en s'abonnant à ces rubriques. Utilisez l'API UpdateEventConfigurations pour contrôler les types d'événement de tâche que vous recevez.

Comme l'annulation ou la suppression de tâches peut prendre un certain temps, deux messages sont envoyés pour indiquer le début et la fin d'une demande. Par exemple, lorsqu'une demande d'annulation démarre, un message est envoyé à la rubrique `$aws/events/job/jobID/cancellation_in_progress`. Lorsque la demande d'annulation est terminée, un message est envoyé à la rubrique `$aws/events/job/jobID/canceled`. Le processus est le même pour une requête de suppression de tâche. Les applications de gestion et de surveillance peuvent effectuer le suivi de l'état des tâches en s'abonnant à ces rubriques.

Pour plus d'informations sur la publication et l'abonnement aux rubriques MQTT, consultez the section called “Protocoles de communication de périphérique” (p. 79).

Tâche terminée/annulée/supprimée

Le service AWS IoT Jobs publie un message dans une rubrique MQTT lorsqu'une tâche est terminée, annulée ou supprimée, ou lorsqu'une annulation ou une suppression est en cours :

- `$aws/events/job/jobID/completed`
- `$aws/events/job/jobID/canceled`
- `$aws/events/job/jobID/deleted`
- `$aws/events/job/jobID/cancellation_in_progress`
- `$aws/events/job/jobID/deletion_in_progress`

Le message `completed`contient l'exemple de charge utile suivant :

```
{
  "eventType": "JOB",
  "eventId": "7364ffd1-8b65-4824-85d5-6c14686c97c6",
  "timestamp": 1234567890,
  "operation": "completed",
  "jobId": "27450507-bf6f-4012-92af-bb8a1c8c4484",
  "status": "COMPLETED",
  "targetSelection": "SNAPSHOT|CONTINUOUS",
  "targets": [
    "arn:aws:iot:us-east-1:123456789012:thing/a39f6f91-70cf-4bd2-a381-9c66df1a80d0",
    "arn:aws:iot:us-east-1:123456789012:thinggroup/2fc4c0a4-6e45-4525-
a238-0fe8d3dd21bb"
  ],
  "description": "My Job Description",
  "completedAt": 1234567890123,
```

```
  "createdAt": 1234567890123,
  "lastUpdatedAt": 1234567890123,
  "jobProcessDetails": {
    "numberOfCanceledThings": 0,
    "numberOfRejectedThings": 0,
    "numberOfFailedThings": 0,
    "numberOfRemovedThings": 0,
    "numberOfSucceededThings": 3
  }
}
```

Le message `canceled`contient l'exemple de charge utile suivant :

```
{
  "eventType": "JOB",
  "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
  "timestamp": 1234567890,
  "operation": "canceled",
  "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
  "status": "CANCELED",
  "targetSelection": "SNAPSHOT|CONTINUOUS",
  "targets": [
    "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
    "arn:aws:iot:us-east-1:123456789012:thinggroup/ThingGroup1-95c644d5-1621-41a6-9aa5-
ad2de581d18f"
  ],
  "description": "My job description",
  "createdAt": 1234567890123,
  "lastUpdatedAt": 1234567890123
}
```

Le message `deleted`contient l'exemple de charge utile suivant :

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "deleted",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
      "status": "DELETED",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

Le message `cancellation_in_progress`contient l'exemple de charge utile suivant :

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "cancellation_in_progress",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
```

```
      "status": "CANCELLATION_IN_PROGRESS",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

Le message `deletion_in_progress`contient l'exemple de charge utile suivant :

```
{
      "eventType": "JOB",
      "eventId": "568d2ade-2e9c-46e6-a115-18afa1286b06",
      "timestamp": 1234567890,
      "operation": "deletion_in_progress",
      "jobId": "4d2a531a-da2e-47bb-8b9e-ff5adcd53ef0",
      "status": "DELETION_IN_PROGRESS",
      "targetSelection": "SNAPSHOT|CONTINUOUS",
      "targets": [
        "arn:aws:iot:us-east-1:123456789012:thing/Thing0-947b9c0c-ff10-4a80-b4b3-
cd33d0145a0f",
        "arn:aws:iot:us-east-1:123456789012:thinggroup/
ThingGroup1-95c644d5-1621-41a6-9aa5-ad2de581d18f"
      ],
      "description": "My job description",
      "createdAt": 1234567890123,
      "lastUpdatedAt": 1234567890123,
      "comment": "Comment for this operation"
    }
```

Statut terminal de l'exécution d'une tâche

Le service AWS IoT Jobs publie un message lorsqu'un appareil indique le statut final pour l'exécution d'une tâche :

- `$aws/events/jobExecution/`*`jobID`*`/succeeded`
- `$aws/events/jobExecution/`*`jobID`*`/failed`
- `$aws/events/jobExecution/`*`jobID`*`/rejected`
- `$aws/events/jobExecution/`*`jobID`*`/canceled`
- `$aws/events/jobExecution/`*`jobID`*`/timed_out`
- `$aws/events/jobExecution/`*`jobID`*`/removed`
- `$aws/events/jobExecution/`*`jobID`*`/deleted`

Le message contient l'exemple de charge utile suivant :

```
{
  "eventType": "JOB_EXECUTION",
  "eventId": "cca89fa5-8a7f-4ced-8c20-5e653afb3572",
  "timestamp": 1234567890,
  "operation": "succeeded|failed|rejected|canceled|removed|timed_out",
  "jobId": "154b39e5-60b0-48a4-9b73-f6f8dd032d27",
  "thingArn": "arn:aws:iot:us-east-1:123456789012:myThing/6d639fbc-8f85-4a90-924d-
a2867f8366a7",
  "status": "SUCCEEDED|FAILED|REJECTED|CANCELED|REMOVED|TIMED_OUT",
  "statusDetails": {
```

```
    "key": "value"
  }
}
```

Événements du cycle de vie

AWS IoT publie des événements de cycle de vie dans les rubriques MQTT présentées dans les sections suivantes. Ces messages vous permettent d'être informé des événements de cycle de vie de l'agent de messages.

> **Note**
>
> Les messages de cycle de vie peuvent être envoyés dans le désordre. Vous pouvez recevoir des messages en double.

Événements de connexion/déconnexion

AWS IoT publie un message dans les rubriques MQTT suivantes lorsqu'un client se connecte ou se déconnecte :

- `$aws/events/presence/connected/`*`clientId`*— Un client connecté au courtier de messages.
- `$aws/events/presence/disconnected/`*`clientId`*— Un client déconnecté du courtier de messages.

Voici une liste d'éléments JSON qui sont contenus dans les messages de connexion/déconnexion publiés dans la rubrique `$aws/events/presence/connected/`*`clientId`*.

clientId

ID du client qui se connecte ou se déconnecte.

> **Note**
>
> Les ID des clients qui contiennent des # ou + ne reçoivent pas les événements de cycle de vie.

clientInitiatedDisconnect

True si le client est à l'origine de la déconnexion. Sinon, la valeur renvoyée est Faux. Figurant seulement dans les messages de déconnexion.

disconnectReason

La raison pour laquelle le client se déconnecte. Figurant uniquement dans les messages de déconnexion. Le tableau suivant contient des valeurs valides.

Raison de la déconnexion	Description
`AUTH_ERROR`	Le client n'a pas pu s'authentifier ou l'autorisation a échoué.
`CLIENT_INITIATED_DISCONNECT`	Le client indique qu'il va se déconnecter. Le client peut le faire en envoyant un paquet de contrôle `DISCONNECT` MQTT ou un `Close frame` si le client utilise une connexion WebSocket.

Raison de la déconnexion	Description
`CLIENT_ERROR`	Le client a réalisé une erreur qui a provoqué sa déconnexion. Par exemple, un client sera déconnecté pour avoir envoyé plus d'un paquet `CONNECT` MQTT sur la même connexion ou pour avoir tenté de publier avec une charge utile supérieure à la limite de cette dernière.
`CONNECTION_LOST`	La connexion client-serveur est coupée. Cela peut se produire pendant une période de latence réseau élevée ou lorsque la connexion Internet est perdue.
`DUPLICATE_CLIENTID`	Le client emploie un ID client déjà utilisé. Dans ce cas, le client déjà connecté sera déconnecté avec cette raison de déconnexion.
`FORBIDDEN_ACCESS`	Le client n'est pas autorisé à être connecté. Par exemple, un client avec une adresse IP refusée échouera à se connecter.
`MQTT_KEEP_ALIVE_TIMEOUT`	S'il n'y a pas de communication client-serveur pour 1.5x le temps de maintien de la connexion du client, le client est déconnecté.
`SERVER_ERROR`	Déconnecté en raison de problèmes de serveur inattendus.
`SERVER_INITIATED_DISCONNECT`	Le serveur déconnecte intentionnellement un client pour des raisons opérationnelles.
`THROTTLED`	Le client est déconnecté en raison du dépassement d'une limite de limitation.
`WEBSOCKET_TTL_EXPIRATION`	Le client est déconnecté car un WebSocket a été connecté plus longtemps que sa valeur de durée de vie.

eventType

Type d'événement. Les valeurs valides sont `connected` ou `disconnected`.

ipAddress

Adresse IP du client de connexion. Elle peut être au format IPv4 ou IPv6. Figurant uniquement dans les messages de connexion.

principalIdentifier

Informations d'identification utilisées pour l'authentification. Pour les certificats d'authentification mutuelle TLS, il s'agit de l'ID du certificat. Pour les autres connexions, ce sont les informations d'identification IAM.

sessionIdentifier

Identifiant globalement unique dans AWS IoT qui existe pendant la durée de vie de la session.

timestamp

Approximation du moment où l'événement s'est produit, exprimée en millisecondes en heure Unix. La précision de l'horodatage est de +/-2 minutes.

versionNumber

Numéro de version de l'événement de cycle de vie. Il s'agit d'un entier qui augmente de façon monotone pour chaque connexion d'ID client. Le numéro de version peut être utilisé par un abonné afin de déduire l'ordre des événements de cycle de vie.

Note

Les messages de connexion et de déconnexion d'une connexion de client ont le même numéro de version.
Le numéro de version peut ignorer des valeurs ; il n'est pas garanti qu'il augmentera uniformément de 1 à chaque événement.
Si un client n'est pas connecté pendant une heure environ, le numéro de version est réinitialisé à 0. En ce qui concerne les sessions permanentes, le numéro de version est réinitialisé à 0 quand un client a été déconnecté pendant plus longtemps que la durée de vie (TTL) configurée pour la session permanente.

Un message de connexion présente la structure suivante.

```
 {
    "clientId": "186b5",
    "timestamp": 1573002230757,
    "eventType": "connected",
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",
    "principalIdentifier": "123456789012345678901234567890123456",
    "ipAddress": "192.0.2.0",
    "versionNumber": 0
}
```

Un message de déconnexion présente la structure suivante.

```
{
    "clientId": "186b5",
    "timestamp": 1573002340451,
    "eventType": "disconnected",
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",
    "principalIdentifier": "123456789012345678901234567890123456",
    "clientInitiatedDisconnect": true,
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT",
    "versionNumber": 0
}
```

Gestion des déconnexions du client

Les bonnes pratiques consistent à toujours implémenter un état d'attente pour les événements du cycle de vie, notamment avec des messages LWT (Last Will and Testament). À réception d'un message de déconnexion, votre code doit déclencher un délai d'attente et vérifier si un appareil est toujours hors ligne avant de prendre des mesures. Pour ce faire, vous pouvez utiliser des files d'attente à retardement SQS. Lorsqu'un client reçoit un message LWT ou un événement de cycle de vie, vous pouvez mettre ce message en file d'attente (pendant 5 secondes, par exemple). Lorsque ce message est disponible et traité (par Lambda ou un autre service), vous pouvez commencer par vérifier si l'appareil est toujours hors ligne avant de prendre d'autres mesures.

Événements d'abonnement/désabonnement

AWS IoT publie un message dans la rubrique MQTT suivante lorsqu'un client s'abonne ou se désabonne à une rubrique MQTT :

```
$aws/events/subscriptions/subscribed/clientId
```

ou

```
$aws/events/subscriptions/unsubscribed/clientId
```

Où `clientId` est l'ID du client MQTT qui se connecte à l'agent de messages AWS IoT.

Le message publié sur cette rubrique a la structure suivante :

```
{
    "clientId": "186b5",
    "timestamp": 1460065214626,
    "eventType": "subscribed" | "unsubscribed",
    "sessionIdentifier": "00000000-0000-0000-0000-000000000000",
    "principalIdentifier": "000000000000/ABCDEFGHIJKLMNOPQRSTU:some-user/
ABCDEFGHIJKLMNOPQRSTU:some-user",
    "topics" : ["foo/bar","device/data","dog/cat"]
}
```

Voici une liste d'éléments JSON qui sont contenus dans les messages d'abonnement/désabonnement publiés dans les rubriques `$aws/events/subscriptions/subscribed/clientId` et `$aws/events/subscriptions/unsubscribed/clientId`.

clientId

ID du client qui s'abonne ou se désabonne.

> **Note**
>
> Les ID des clients qui contiennent des # ou + ne reçoivent pas les événements de cycle de vie.

eventType

Type d'événement. Les valeurs valides sont `subscribed` ou `unsubscribed`.

principalIdentifier

Informations d'identification utilisées pour l'authentification. Pour les certificats d'authentification mutuelle TLS, il s'agit de l'ID du certificat. Pour les autres connexions, ce sont les informations d'identification IAM.

sessionIdentifier

Identifiant globalement unique dans AWS IoT qui existe pendant la durée de vie de la session.

timestamp

Approximation du moment où l'événement s'est produit, exprimée en millisecondes en heure Unix. La précision de l'horodatage est de +/-2 minutes.

topics

Tableau des sujets MQTT auquel le client s'est abonné.

> **Note**
>
> Les messages de cycle de vie peuvent être envoyés dans le désordre. Vous pouvez recevoir des messages en double.

AWS IoT Core pour LoRaWAN

La .Alliance LoRadécrit LoraWan comme« un protocole réseau LPWA (Low Power, Wide Area) conçu pour connecter sans fil des objets fonctionnant sur batterie à Internet dans les réseaux régionaux, nationaux ou mondiaux, et cible les principales exigences de l'Internet des objets (IoT) telles que la communication bidirectionnelle, la sécurité de bout en bout, la mobilité et les services de localisation. ".

Qu'est-ce que LoRaWAN ?

Le protocole LoRaWan est un protocole LPWAN (Low Power Wide Area Networking) qui fonctionne sur LoRa, une technologie de fréquence audio sans fil qui fonctionne dans un spectre de fréquences radio sans licence. La spécification LoRaWan est ouverte afin que tout le monde puisse configurer et exploiter un réseau LoRa. La technologie LoRa prend en charge la communication longue portée au prix d'une bande passante étroite. Il utilise une forme d'onde à bande étroite avec une fréquence centrale pour envoyer des données, ce qui la rend robuste aux interférences. Voici les caractéristiques de la technologie LoRaWan.

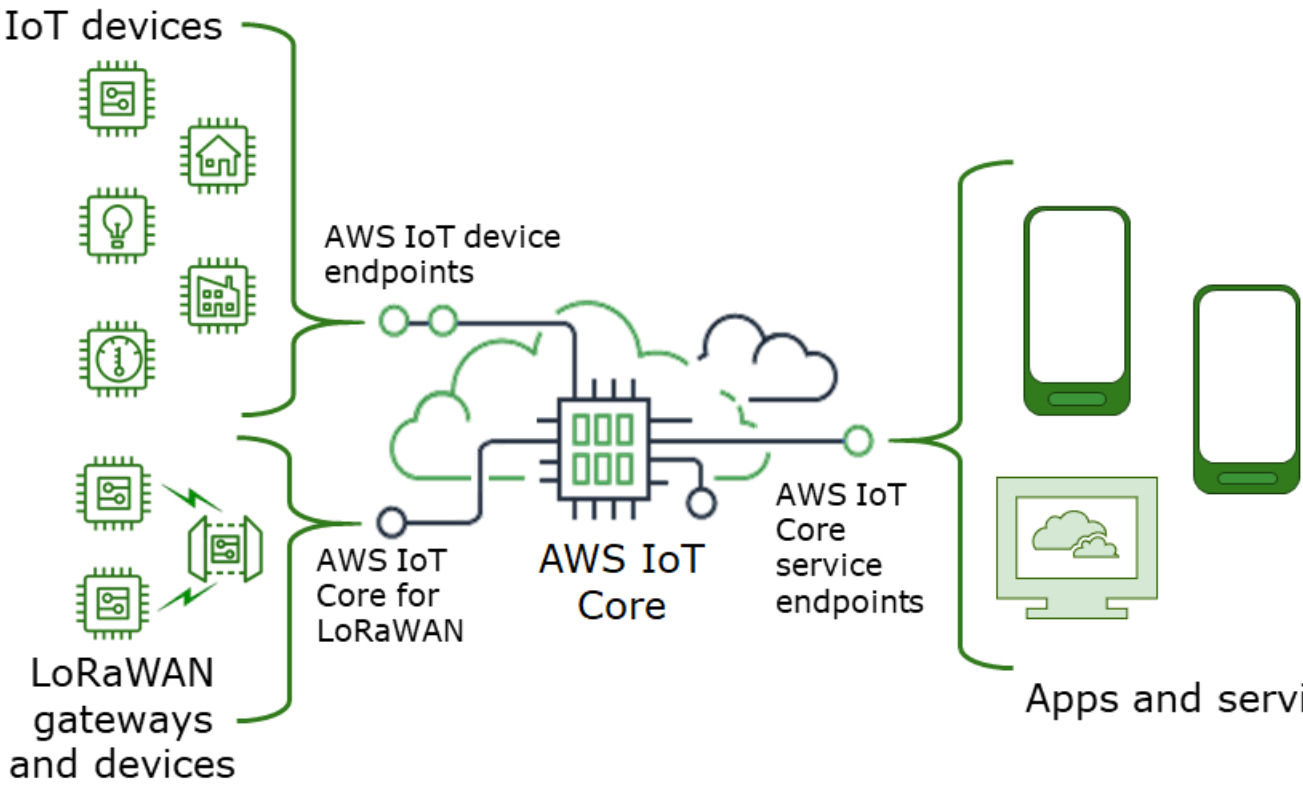
Caractéristiques de la technologie LoraWan

- Communication longue portée jusqu'à 10 miles en ligne de mire.
- Longue durée de la batterie jusqu'à 10 ans. Pour améliorer l'autonomie de la batterie, vous pouvez utiliser vos appareils en mode Classe A ou Classe B, ce qui nécessite une latence accrue de liaison descendante.
- Faible coût pour les appareils et la maintenance.
- Spectre radioélectrique sans licence, mais des règlements propres à la région s'appliquent.
- Faible puissance, mais a une taille de charge utile limitée de 51 octets à 241 octets en fonction du débit de données. Le débit de données peut être de 0,3 kbit/s à 27 kbit/s avec une taille de charge utile maximale de 222.

Quoi AWS IoT Core pour LoRaWAN peut faire

L'architecture réseau LoRaWan est déployée dans une topologie étoile dans laquelle les passerelles relayent les informations entre les périphériques finaux et le serveur réseau LoRaWan (LNS).

AWS IoT Core pour LoRaWan vous aide à connecter et à gérer des appareils sans fil LoRaWan (réseau étendu à longue portée de faible consommation) et remplace la nécessité pour vous de développer et d'exploiter un LNS. Les périphériques et passerelles longue portée WAN (LoRaWan) peuvent se connecter à AWS IoT Core à l'aide d' AWS IoT Core pour LoRaWAN.



Les appareils LoRaWan communiquent avec AWS IoT Core via les passerelles LoRaWan.

- Appareils : Les terminaux LoRaWan se connectent aux passerelles LoRaWan à l'aide du protocole de communication LoRa.
- Passerelles : Les passerelles LoRaWan se connectent à AWS IoT Core à l'aide du protocole LoRa BasicStation sur des WebSockets sécurisés.
- AWS IoT Core pour LoRaWAN : AWS IoT Core pour LoRaWan gère les stratégies de service et d'appareil AWS IoT Core nécessite de gérer et de communiquer avec les passerelles et les appareils LoRaWan. AWS IoT Core pour LoRaWan gère également les destinations qui décrivent lesAWS IoTqui envoient des données de périphérique à d'autres services.
- Applications et services :AWS IoTrègles envoient des messages de périphérique LoRaWan à d'autresAWSet peut traiter les messages de l'appareil pour formater les données des services.

Ressources d'apprentissage pour AWS IoT Core pour LoRaWAN

Les ressources suivantes vous aideront à vous familiariser avec la technologie LoRaWan et AWS IoT Core pour LoRaWAN.

En savoir plus sur LoRaWAN

Les liens suivants contiennent des informations utiles sur la technologie LoRaWan et sur LoRa Basics Station, qui est le logiciel qui s'exécute sur vos passerelles LoRaWan pour connecter des périphériques terminaux à AWS IoT Core pour LoRaWAN.

- Les fondements des choses sur LoRaWan

The Things Fundamentals on LoraWan contient une vidéo d'introduction qui couvre les principes fondamentaux de LoRaWan et une série de chapitres qui vous aideront à en apprendre davantage sur LoRa et LoRaWan.

- Présentation de

  LoRa Alliance fournit une vue d'ensemble technique de LoRa et LoRaWan, y compris un résumé des spécifications LoRaWan dans différentes régions.

- Principes de base LoRaM

  Semtech Corporation fournit des concepts utiles sur les bases LoRa pour les passerelles et les nœuds d'extrémité. LoRa Basics Station, un logiciel open source qui s'exécute sur votre passerelle LoRaWan, est maintenu et distribué via leGitHubrepository. Vous pouvez également en savoir plus sur les protocoles LNS et CUPS qui décrivent comment échanger des données LoRaWan et effectuer des mises à jour de configuration.

AWS IoT Core Ressources LoRaWAN

Les liens suivants contiennent des informations utiles sur la mise en route avec AWS IoT Core pour LoRaWan et comment il peut vous aider à créer des solutions IoT.

- Démarrer avec AWS IoT Core pour LoRaWAN

  La vidéo suivante décrit comment AWS IoT Core Pour LoRaWan travaille et vous guide à travers le processus d'ajout de passerelles LoRaWan à partir duAWS Management Console.

- AWS IoT Core pour LoRaWAN

  L'atelier couvre les fondamentaux de la technologie LoRaWan et leur mise en œuvre avec AWS IoT Core pour LoRaWAN. Vous pouvez également utiliser l'atelier pour parcourir les laboratoires qui montrent comment connecter votre passerelle et votre appareil à AWS IoT Core pour LoRaWan afin de créer un exemple de solution IoT.

Connexion de passerelles et d'appareils à AWS IoT Core pour LoRaWAN

AWS IoT Core pour LoRaWan vous aide à connecter et à gérer des appareils sans fil LoRaWan (réseau étendu à longue portée de faible consommation) et remplace la nécessité pour vous de développer et d'exploiter un LNS. Les périphériques et passerelles longue portée WAN (LoRaWan) peuvent se connecter à AWS IoT Core à l'aide d' AWS IoT Core pour LoRaWAN.

Pour commencer à utiliser AWS IoT Core pour LoRaWAN

- Sélectionnez les périphériques sans fil et les passerelles LoRaWan dont vous avez besoin

  La .AWSCatalogue des appareils partenairescontient des passerelles et des kits de développement qualifiés pour une utilisation avec AWS IoT Core pour LoRaWAN. Pour plus d'informations, consultez Utilisation de passerelles qualifiées à partir de laAWSCatalogue des appareils partenaires (p. 1085).

- Ajoutez vos appareils sans fil et vos passerelles LoRaWan à AWS IoT Core pour LoRaWAN

  Connexion de passerelles et d'appareils à AWS IoT Core pour LoRaWAN (p. 1064)vous donne des informations sur la façon de décrire vos ressources et d'ajouter vos appareils sans fil et vos passerelles

LoRaWan à AWS IoT Core pour LoRaWAN. Vous allez également apprendre à configurer l'autre AWS IoT Core pour les ressources LoRaWan dont vous aurez besoin pour gérer ces appareils et envoyer leurs données àAWSServices .

- Terminer votre AWS IoT Core pour LoRaWAN

  Commencez parNotre échantillon AWS IoT Core pour LoRaWANet faites-le à vous.

Conventions de dénomination pour vos appareils, passerelles, profils et destinations

Avant de commencer à utiliser AWS IoT Core pour LoRaWan et en créant les ressources, tenez compte de la convention de dénomination de vos appareils, passerelles et destination.

AWS IoT Core pour LoRaWan attribue des ID uniques aux ressources que vous créez pour les périphériques sans fil, les passerelles et les profils. Toutefois, vous pouvez également donner à vos ressources des noms plus descriptifs pour faciliter leur identification. Avant d'ajouter des périphériques, des passerelles, des profils et des destinations à AWS IoT Core Pour LoRawan, réfléchissez à la façon dont vous allez les nommer pour simplifier leur gestion.

Vous pouvez également ajouter des balises aux ressources que vous créez. Avant d'ajouter vos appareils LoRaWan, réfléchissez à la façon dont vous pouvez utiliser les balises pour identifier et gérer vos AWS IoT Core pour les ressources LoRaWAN. Les balises peuvent être modifiées après les avoir ajoutées.

Pour de plus amples informations sur l'attribution de noms et de balisage, veuillez consulter.Décrire vos AWS IoT Core Ressources LoRaWAN (p. 1066).

Mappage des données du périphérique aux données du service

Les données des appareils sans fil LoRaWan sont souvent codées pour optimiser la bande passante. Ces messages encodés arrivent à AWS IoT Core pour LoRaWan dans un format qui pourrait ne pas être facilement utilisé par d'autresAWSServices . AWS IoT Core pour LoRaWAN utiliseAWS IoTrègles qui peuvent utiliserAWS Lambdapour traiter et décoder les messages de l'appareil dans un format que d'autresAWSpeuvent utiliser.

Pour transformer les données du périphérique et les envoyer à d'autresAWSservices, vous devez savoir :

- Format et contenu des données envoyées par les périphériques sans fil.
- Le service auquel vous souhaitez envoyer les données.
- Format requis par le service.

À l'aide de ces informations, vous pouvez créer l'AWS IoTqui effectue la conversion et envoie les données converties à la méthodeAWSqui l'utiliseront.

Utilisation de la console pour embarquer votre appareil et la passerelle vers AWS IoT Core pour LoRaWAN

Vous pouvez utiliser l'interface de la console ou l'API pour ajouter votre passerelle et vos périphériques LoRaWan. Si vous utilisez AWS IoT Core Pour LoRawan pour la première fois, nous vous recommandons d'utiliser la console. L'interface de la console est plus pratique lors de la gestion de quelques AWS IoT Core pour les ressources LoRaWan à la fois. Lors de la gestion d'un grand nombre de AWS IoT Core pour les ressources LoRaWan, envisagez de créer des solutions plus automatisées à l'aide duAWS IoTAPI sans fil.

Une grande partie des données que vous entrez lors de la configuration AWS IoT Core pour les ressources LoRaWan sont fournies par les fournisseurs des appareils et sont spécifiques aux spécifications LoRaWan qu'ils prennent en charge. Les rubriques suivantes décrivent comment décrire votre AWS IoT Core pour les ressources LoRaWan et utilisez la console ou l'API pour ajouter vos passerelles et périphériques.

Rubriques

Décrire vos AWS IoT Core Ressources LoRaWAN

Si vous utilisez AWS IoT Core Pour LoRaWan pour la première fois, vous pouvez ajouter votre première passerelle et votre premier appareil LoRaWan à l'aide de l' AWS IoT Core pour LoRaWANPage d'introduction de laAWS IoTconsole

Avant de commencer à créer les ressources, tenez compte de la convention de dénomination de vos appareils, passerelles et destination. AWS IoT Core pour LoRaWan propose plusieurs options pour identifier les ressources que vous créez. HILE AWS IoT Core pour les ressources LoRaWan reçoivent un ID unique lors de leur création, cet ID n'est pas descriptif ni ne peut être modifié après la création de la ressource. Vous pouvez également attribuer un nom, ajouter une description et joindre des balises et des valeurs de balise à la plupart des AWS IoT Core pour les ressources LoRaWan afin de faciliter la sélection, l'identification et la gestion de votre AWS IoT Core pour les ressources LoRaWAN.

- Noms des ressources (p. 1066)

  Pour les passerelles, les périphériques et les profils, le nom de la ressource est un champ facultatif que vous pouvez modifier après la création de la ressource. Le nom apparaît dans les listes affichées sur les pages du hub de ressources.

  Pour les destinations, vous fournissez un nom unique à votreAWSCompte et Région AWS . Vous ne pouvez pas modifier le nom de destination une fois que vous créez la ressource de destination.

  Alors qu'un nom peut contenir jusqu'à 256 caractères, l'espace d'affichage dans le hub de ressources est limité. Assurez-vous que la partie distinctive du nom apparaît dans les 20 à 30 premiers caractères, si possible.
- Indicateurs de ressource (p. 1067)

  Les balises sont des paires clé-valeur de métadonnées qui peuvent être attachées àAWSAWS. Vous choisissez les deux clés de balise et leurs valeurs correspondantes.

  Les passerelles, destinations et profils peuvent comporter 50 balises maximum. Les appareils ne prennent pas en charge les balises.

Noms des ressources

AWS IoT Core pour la prise en charge des ressources LoRaWan pour le nom

Ressource	Prise en charge du champ Nom	
Destination	Le nom est un ID unique de la ressource et ne peut pas être modifié.	

Ressource	Prise en charge du champ Nom	
Device	Nom est un descripteur facultatif de ressource et peut être modifié.	
Passerelle	Nom est un descripteur facultatif de ressource et peut être modifié.	
Profil	Nom est un descripteur facultatif de ressource et peut être modifié.	

Le champ nom apparaît dans les listes de ressources du hub de ressources ; toutefois, l'espace est limité et seuls les 15 à 30 premiers caractères du nom peuvent être visibles.

Lorsque vous sélectionnez des noms pour vos ressources, considérez comment vous souhaitez qu'elles identifient les ressources et comment elles seront affichées dans la console.

Description

Les ressources de destination, de périphérique et de passerelle prennent également en charge un champ de description, qui peut accepter jusqu'à 2 048 caractères. Le champ de description s'affiche uniquement dans la page détaillée de la ressource individuelle. Bien que le champ de description puisse contenir beaucoup d'informations, car il n'apparaît que dans la page détaillée de la ressource, il n'est pas pratique pour l'analyse dans le contexte de plusieurs ressources.

Indicateurs de ressource

AWS IoT Core pour la prise en charge des ressources LoRaWan pourAWSbalises.

Ressource	AWSSupport des balises	
Destination	jusqu'à 50AWSBalises peuvent être ajoutées à la ressource.	
Device	Cette ressource ne prend pas en chargeAWSBalises.	
Passerelle	jusqu'à 50AWSBalises peuvent être ajoutées à la ressource.	
Profil	jusqu'à 50AWSBalises peuvent être ajoutées à la ressource.	

Les balises sont des mots ou des expressions qui agissent comme des métadonnées que vous pouvez utiliser pour identifier et organiser vosAWSAWS. Vous pouvez considérer la clé de balise comme une catégorie d'informations et la valeur de balise comme une valeur spécifique dans cette catégorie.

Par exemple, vous pouvez avoir une valeur de balisecolor, puis donnez à certaines ressources une valeur debluepour cette balise et d'autres une valeur dered. Avec cela, vous pouvez utiliserTag Editordans leAWSpour trouver les ressources avec uncolorValeur de baliseblue.

Pour de plus amples informations sur le balisage et les stratégies de balisage, veuillez consulter.Tag Editor.

À bord de vos passerelles vers AWS IoT Core pour LoRaWAN

Si vous utilisez AWS IoT Core Pour LoRaWan pour la première fois, vous pouvez ajouter votre première passerelle et votre premier appareil LoRaWan à l'aide de la console.

Avant d'intégrer votre passerelle

Avant d'embarquer sur votre passerelle vers AWS IoT Core Pour LoraWan, nous vous recommandons de procéder comme suit :

- Utilisez des passerelles qualifiées pour une utilisation avec AWS IoT Core pour LoRaWAN. Ces passerelles se connectent à AWS IoT Core sans aucun paramètre de configuration supplémentaire et disposer d'une version compatible duPrincipes de base LoRaMles logiciels qui s'exécutent sur eux. Pour plus d'informations, consultez Gestion des passerelles avec AWS IoT Core pour LoRaWAN (p. 1085).
- Considérez la convention d'attribution de noms des ressources que vous créez afin que vous puissiez les gérer plus facilement. Pour plus d'informations, consultez Décrire vos AWS IoT Core Ressources LoRaWAN (p. 1066).
- Faites en sorte que les paramètres de configuration propres à chaque passerelle soient prêts à entrer à l'avance, ce qui facilite la saisie des données dans la console. Les paramètres de configuration de la passerelle sans filAWS IoTnécessite de communiquer avec et de gérer la passerelle, y compris l'EUI de la passerelle et sa bande de fréquences LoRa.

Pour l'intégration de vos passerelles vers AWS IoT Core pour LoRaWAN :

- Envisager la sélection de la bande de fréquences et ajouter le rôle IAM nécessaire (p. 1068)
- Ajouter une passerelle à AWS IoT Core pour LoRaWAN (p. 1070)
- Connect votre passerelle LoRaWan et vérifiez son état de connexion (p. 1072)

Envisager la sélection de la bande de fréquences et ajouter le rôle IAM nécessaire

Avant d'ajouter votre passerelle à AWS IoT Core pour LoRaWan, nous vous recommandons de prendre en compte la bande de fréquences dans laquelle votre passerelle fonctionnera et d'ajouter le rôle IAM nécessaire pour connecter votre passerelle à AWS IoT Core pour LoRaWAN.

> Note
>
> Si vous ajoutez votre passerelle à l'aide de la console, cliquez surCréation d'un rôledans la console pour créer le rôle IAM nécessaire afin que vous puissiez ensuite ignorer ces étapes. Vous devez effectuer ces étapes uniquement si vous utilisez l'interface de ligne de commande pour créer la passerelle.

Considérez la sélection de bandes de fréquences LoRa pour vos passerelles et votre connexion de périphérique

AWS IoT Core pour LoRaWan prend en charge les bandes de fréquences EU863-870, US902-928, AU915 et AS923-1, que vous pouvez utiliser pour connecter vos passerelles et périphériques physiquement présents dans les pays qui prennent en charge les plages de fréquences et les caractéristiques de ces bandes. Les bandes EU863-870 et US902-928 sont couramment utilisées en Europe et en Amérique du Nord, respectivement. La bande AS923-1 est couramment utilisée en Australie, en Nouvelle-Zélande, au Japon et à Singapour, entre autres pays. L'AU915 est utilisé en Australie et en Argentine entre

autres pays. Pour plus d'informations sur la bande de fréquences à utiliser dans votre région ou pays, consultezParamètres régionaux LoraWan®.

LoRa Alliance publie des spécifications LoRaWan et des documents de paramètres régionaux qui peuvent être téléchargés sur le site Web de LoRa Alliance. Les paramètres régionaux de LoRa Alliance aident les entreprises à décider quelle bande de fréquences utiliser dans leur région ou leur pays. AWS IoT Core pour la mise en œuvre de la bande de fréquences de LoRaWan suit la recommandation du document de spécification des paramètres régionaux. Ces paramètres régionaux sont regroupés en un ensemble de paramètres radio, ainsi qu'une allocation de fréquence adaptée à la bande industrielle, scientifique et médicale (ISM). Nous vous recommandons de travailler avec les équipes de conformité pour vous assurer que vous respectez toutes les exigences réglementaires applicables.

Ajouter un rôle IAM pour permettre au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de la passerelle

Cette procédure décrit comment ajouter un rôle IAM qui permettra au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de passerelle. Assurez-vous d'effectuer cette procédure avant qu'une passerelle LoRaWan ne tente de se connecter à AWS IoT Core Pour LoraWan ; cependant, vous n'avez besoin de le faire qu'une seule fois.

Ajouter le rôle IAM pour permettre au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de la passerelle

1. Ouverture d'Hub de rôles de la console IAMet choisissezCréation d'un rôle.
2. Si vous pensez que vous avez peut-être déjà ajouté leIOTWIrelessGatewayCertManagerRoleRôle, dans la barre de recherche, entrez**`IoTWirelessGatewayCertManagerRole`**.

   Si unIOTWIrelessGatewayCertManagerRoledans les résultats de recherche, vous disposez du rôle IAM nécessaire. Vous pouvez quitter la procédure maintenant.

   Si les résultats de la recherche sont vides, vous n'avez pas le rôle IAM nécessaire. Continuez la procédure pour l'ajouter.
3. DansZone Select type of trusted entity, choisissezAutre Compte AWS .
4. DansID de compte, entrez votre Compte AWS ID, puis choisissezSuivant: Permissions (Autorisations).
5. Dans la zone de recherche, saisissez **`AWSIoTWirelessGatewayCertManager`**.
6. Dans la liste des résultats de recherche, sélectionnez la stratégie nomméeAWSIOTWirelessGatewayCertManager.
7. Choisissez Next (Suivant) Tags (Balises), puisSuivant: Review (Examiner).
8. DansNom de rôle, saisissez**`IoTWirelessGatewayCertManagerRole`**, puisCréation d'un rôle.
9. Pour modifier le nouveau rôle, dans le message de confirmation, sélectionnezIOTWIrelessGatewayCertManagerRole.
10. DansRécapitulatif, choisissez leRelations d'approbation, puis.Modification de la relation d'approbation.
11. DansDocument de stratégie, modifiez`Principal`se présente ainsi.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

Une fois que vous avez modifié le`Principal`, le document de stratégie complet doit se présenter comme cet exemple.

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

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

12. Pour enregistrer vos modifications et quitter, choisissezStratégie d'approbation.

Vous avez maintenant créé leIOTWIrelessGatewayCertManagerRole. Tu n'auras pas besoin de recommencer.

Si vous avez effectué cette procédure lors de l'ajout d'une passerelle, vous pouvez fermer cette fenêtre et la console IAM et revenir à laAWS IoTpour terminer l'ajout de la passerelle.

Ajouter une passerelle à AWS IoT Core pour LoRaWAN

Vous pouvez ajouter votre passerelle à AWS IoT Core Pour LoraWan à l'aide de la console ou de l'interface de ligne de commande.

Avant d'ajouter votre passerelle, nous vous recommandons de prendre en compte les facteurs mentionnés dans laAvant d'intégrer votre passerelleSection deÀ bord de vos passerelles vers AWS IoT Core pour LoRaWAN (p. 1068).

Si vous ajoutez votre passerelle pour la première fois, nous vous recommandons d'utiliser la console. Si vous souhaitez ajouter votre passerelle à l'aide de l'interface de ligne de commande, vous devez avoir déjà créé le rôle IAM nécessaire pour que la passerelle puisse se connecter à AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la création du rôle, consultez la rubriqueAjouter un rôle IAM pour permettre au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de la passerelle (p. 1069).

Ajouter une passerelle à l'aide de la console

Accédez à . AWS IoT Core pour LoRaWAN IntroductionPage de laAWS IoTet choisissezMise en route, puisAjout d'un. Si vous avez déjà ajouté une passerelle, choisissezAfficher la passerellepour afficher la passerelle que vous avez ajoutée. Si vous souhaitez ajouter d'autres passerelles, choisissezAjout d'un.

1. Fournir des détails sur la passerelle et des informations sur la bande

   Utilisation de l'Détails de la passerellepour fournir des informations sur les données de configuration du périphérique telles que l'IUE de la passerelle et la configuration de la bande de fréquences.

   - EUI de Gateway

     L'EUI (Extended Unique Identifier) du périphérique de passerelle individuel. L'IUE est un code alphanumérique à 16 chiffres, tel que`c0ee40ffff29df10`, qui identifie de manière unique une passerelle dans votre réseau LoRaWan. Ces informations sont spécifiques à votre modèle de passerelle et vous pouvez les trouver sur votre périphérique de passerelle ou dans son manuel d'utilisation.

     > **Note**
     >
     > L'interface utilisateur de la passerelle est différente de l'adresse MAC Wi-Fi que vous pouvez voir imprimée sur votre périphérique passerelle. L'IUE respecte une norme EUI-64 qui identifie de manière unique votre passerelle et ne peut donc pas être reprise dans d'autres Compte AWS s et régions.

- Bande de fréquence (RFRegion)

  La bande de fréquences de la passerelle. Vous pouvez choisir parmi`US915`,`EU868`,`AU915`, ou`AS923-1`, en fonction de ce que votre passerelle prend en charge et du pays ou de la région à partir duquel la passerelle se connecte physiquement. Pour de plus amples informations sur les bandes, veuillez consulterConsidérez la sélection de bandes de fréquences LoRa pour vos passerelles et votre connexion de périphérique (p. 1068).

2. Spécifiez les données de configuration de votre passerelle sans fil (facultatif)

   Ces champs sont facultatifs et vous pouvez les utiliser pour fournir des informations supplémentaires sur la passerelle et sa configuration.

   - Nom, description et balises de votre passerelle

     Les informations contenues dans ces champs facultatifs proviennent de la façon dont vous organisez et décrivez les éléments de votre système sans fil. Vous pouvez affecter unNomà la passerelle, utilisez l'outilDescriptionpour fournir des informations sur la passerelle et utiliserTags (Balises)pour ajouter des paires clé-valeur de métadonnées sur la passerelle. Pour de plus amples informations sur l'attribution de noms et la description à vos ressources, veuillez consulter.Décrire vos AWS IoT Core Ressources LoRaWAN (p. 1066).

   - Configuration de LoRaWan à l'aide de sous-bandes et de filtres

     Vous pouvez également spécifier des données de configuration LoRaWan telles que les sous-canaux que vous souhaitez utiliser et les filtres permettant de contrôler le flux de trafic. Pour ce didacticiel, vous pouvez ignorer ces champs. Pour plus d'informations, consultez Configurer les sous-bandes et les capacités de filtrage de votre passerelle (p. 1086).

3. Associer unAWS IoTchose avec la passerelle

   Indiquez s'il faut créer unAWS IoTchose et l'associer à la passerelle. Choses dansAWS IoTcontribuent à rechercher et à gérer vos appareils. Associer une chose à votre passerelle permet à la passerelle d'accéder à d'autres AWS IoT Core Fonctionnalités

4. Créer et télécharger le certificat de passerelle

   Pour authentifier votre passerelle afin qu'elle puisse communiquer en toute sécurité avecAWS IoT, votre passerelle LoRaWan doit présenter une clé privée et un certificat à AWS IoT Core pour LoRaWAN. Création d'unCertificat de passerellede sorte queAWS IoTpeut vérifier l'identité de votre passerelle à l'aide de la norme X.509.

   Cliquez surCréer un certificatet téléchargez les fichiers de certificat. Vous les utiliserez ultérieurement pour configurer votre passerelle.

5. Copiez les points de terminaison CUPS et LNS et téléchargez les certificats

   Votre passerelle LoRaWan doit se connecter à un point de terminaison CUPS ou LNS lors de l'établissement d'une connexion à AWS IoT Core pour LoRaWAN. Nous vous recommandons d'utiliser le point de terminaison CUPS car il fournit également la gestion de la configuration. Pour vérifier l'authenticité de AWS IoT Core pour les points de terminaison LoRaWan, votre passerelle utilisera un certificat d'approbation pour chacun des points de terminaison CUPS et LNS,

   Cliquez surCopierpour copier les points de terminaison CUPS et LNS. Vous aurez besoin de ces informations ultérieurement afin de configurer votre passerelle. Ensuite, cliquez surTélécharger les certificats d'approbation du serveurpour télécharger les certificats d'approbation des points de terminaison CUPS et LNS.

6. Créer le rôle IAM pour les autorisations de passerelle

   Vous devez ajouter un rôle IAM qui permet au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de passerelle. Vous devez le faire avant qu'une passerelle LoRaWan tente de se connecter avec AWS IoT Core pour LoRaWan ; cependant, vous ne devez le faire qu'une seule fois.

Pour créer leIOTWIrelessGatewayCertManagerIAM pour votre compte, cliquez sur le boutonCréation d'un rôle. Si le rôle existe déjà, sélectionnez-le dans la liste déroulante.

Cliquez surEnvoyerpour terminer la création de la passerelle.

Ajouter une passerelle à l'aide de l'API

Si vous ajoutez une passerelle pour la première fois à l'aide de l'API ou de l'interface de ligne de commande, vous devez ajouter la propriétéIOTWIrelessGatewayCertManagerRôle IAM afin que la passerelle puisse se connecter à AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la création d'un rôle, consultez la section suivante.Ajouter un rôle IAM pour permettre au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de la passerelle (p. 1069).

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'une passerelle LoRaWan.

AWS IoTActions d'API sans fil pour AWS IoT Core pour les passerelles LoRaWAN

- CreateWirelessGateway
- GetWirelessGateway
- ListWirelessGateways
- UpdateWirelessGateway
- DeleteWirelessGateway

Pour obtenir la liste complète des actions et types de données disponibles pour la création et la gestion AWS IoT Core Pour les ressources LoRaWAN, consultezAWS IoTRéférence d'API sans fil.

Comment utiliserAWS CLIpour ajouter une passerelle

Vous pouvez utiliser la stratégieAWS CLIpour créer une passerelle sans fil à l'aide de l'outilcreate-wireless gatewayCommande l'. L'exemple suivant crée une passerelle de périphérique LoRawan sans fil. Vous avez également la possibilité de fournir un`input.json`qui contiendra des détails supplémentaires tels que le certificat de passerelle et les informations d'identification de provisionnement.

> **Note**
>
> Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

```
aws iotwireless create-wireless-gateway \
    --lorawan GatewayEui="a1b2c3d4567890ab",RfRegion="US915" \
    --name "myFirstLoRaWANGateway" \
    --description "Using my first LoRaWAN gateway"
    --cli-input-json input.json
```

Pour de plus amples informations sur les CLI que vous pouvez utiliser, veuillez consulter.AWS CLIRéférence

Connect votre passerelle LoRaWan et vérifiez son état de connexion

Avant de pouvoir vérifier l'état de connexion de la passerelle, vous devez avoir déjà ajouté votre passerelle et l'avoir connectée à AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la façon d'ajouter votre passerelle, consultez la pageAjouter une passerelle à AWS IoT Core pour LoRaWAN (p. 1070).

Connect votre passerelle à AWS IoT Core pour LoRaWAN

Après avoir ajouté votre passerelle, connectez-vous à l'interface de configuration de votre passerelle pour entrer les informations de configuration et les certificats d'approbation.

Après avoir ajouté les informations de la passerelle à AWS IoT Core pour LoraWan, ajoutez quelques AWS IoT Core Pour obtenir des informations LoRaWan sur le périphérique de passerelle. La documentation fournie par le fournisseur de la passerelle doit décrire le processus de téléchargement des fichiers de certificats vers la passerelle et de configuration du périphérique de passerelle pour qu'il communique avec AWS IoT Core pour LoRaWAN.

Passerelles qualifiées pour une utilisation avec AWS IoT Core pour LoRaWAN

Pour obtenir des instructions sur la façon de configurer votre passerelle LoRawan, consultez le documentConfiguration du périphérique de passerelleSection du AWS IoT Core pour LoRaWAN Vous trouverez ici des informations sur les instructions de connexion des passerelles pouvant être utilisées avec AWS IoT Core pour LoRaWAN.

Passerelles qui prennent en charge le protocole CUPS

Les instructions suivantes expliquent comment connecter vos passerelles qui prennent en charge le protocole CUPS.

1. Téléchargez les fichiers suivants que vous avez obtenus lors de l'ajout de votre passerelle.
   - Les fichiers de certificat et de clé privée des périphériques de passerelle.
   - Fichier de certificat d'approbation pour le point de terminaison CUPS,`cups.trust`.
2. Spécifiez l'URL du point de terminaison CUPS que vous avez obtenue précédemment. Le point de terminaison sera au format*`prefix`*`.cups.lorawan.`*`region`*`.amazonaws.com:443`.

Pour plus d'informations sur la façon d'obtenir ces informations, consultez la pageAjouter une passerelle à AWS IoT Core pour LoRaWAN (p. 1070).

Passerelles prenant en charge le protocole LNS

Les instructions suivantes expliquent comment connecter vos passerelles qui prennent en charge le protocole LNS.

1. Téléchargez les fichiers suivants que vous avez obtenus lors de l'ajout de votre passerelle.
   - Les fichiers de certificat et de clé privée des périphériques de passerelle.
   - Fichier de certificat d'approbation pour le point de terminaison LNS,`lns.trust`.
2. Spécifiez l'URL du point de terminaison LNS que vous avez obtenue précédemment. Le point de terminaison sera au format*`prefix`*`.lns.lorawan.`*`region`*`.amazonaws.com:443`.

Pour plus d'informations sur la façon d'obtenir ces informations, consultez la pageAjouter une passerelle à AWS IoT Core pour LoRaWAN (p. 1070).

Après cela, vous avez connecté votre passerelle à AWS IoT Core Pour LoRAWAN, vous pouvez vérifier l'état de votre connexion et obtenir des informations sur la date à laquelle la dernière liaison montante a été reçue à l'aide de la console ou de l'API.

Vérifier l'état de connexion de la passerelle à l'aide de

Pour vérifier l'état de la connexion à l'aide de la console, accédez à l'ongletPasserellesPage de laAWS IoTet choisissez la passerelle que vous avez ajoutée. DansDétails spécifiques LoRaWANde la page de détails de la passerelle, vous verrez l'état de la connexion ainsi que la date et l'heure de réception de la dernière liaison montante.

Vérifier l'état de connexion de la passerelle à l'aide de

Pour vérifier l'état de la connexion à l'aide de l'API, utilisez le`GetWirelessGatewayStatistics`API. Cette API n'a pas de corps de requête et contient uniquement un corps de réponse qui indique si la passerelle est connectée et quand la dernière liaison montante a été reçue.

```
HTTP/1.1 200
Content-type: application/json

{
  “ConnectionStatus”: “Connected”,
  “LastUplinkReceivedAt”: “2021-03-24T23:13:08.476015749Z”,
  “WirelessGatewayId”: “30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

À bord de vos appareils AWS IoT Core pour LoRaWAN

Une fois que vous avez intégré votre passerelle vers AWS IoT Core pour LoRaWan et vérifié son état de connexion, vous pouvez embarquer vos appareils sans fil. Pour plus d'informations sur la façon d'intégrer vos passerelles, consultez la pageÀ bord de vos passerelles vers AWS IoT Core pour LoRaWAN (p. 1068).

Les appareils LoRaWan utilisent un protocole LoRaWan pour échanger des données avec des applications hébergées dans le cloud. AWS IoT Core pour LoRaWan prend en charge les appareils conformes aux spécifications 1.0.x ou 1.1 LoRaWan standardisées par LoRa Alliance.

Un appareil LoRaWan contient généralement un ou plusieurs capteursandacteurs. Les appareils envoient des données de télémétrie de liaison montante via des passerelles LoRaWan à AWS IoT Core pour LoRaWAN. Les applications hébergées dans le cloud peuvent contrôler les capteurs en envoyant des commandes de liaison descendante aux appareils LoRaWan via des passerelles LoRaWan.

Avant d'intégrer votre appareil sans fil

Avant d'embarquer sur votre appareil sans fil pour AWS IoT Core Pour LoRawan, vous devez disposer des informations suivantes à l'avance :

- Spécifications LoRaWan et configuration des périphériques sans fil

  Le fait que les paramètres de configuration propres à chaque appareil soient prêts à entrer à l'avance facilite la saisie des données dans la console. Les paramètres spécifiques que vous devez entrer dépendent de la spécification LoRaWan utilisée par l'appareil. Pour obtenir la liste complète de ses spécifications et paramètres de configuration, consultez la documentation de chaque périphérique.
- Nom et description de l'appareil (facultatif)

  Les informations contenues dans ces champs facultatifs proviennent de la façon dont vous organisez et décrivez les éléments de votre système sans fil. Pour de plus amples informations sur l'attribution de noms et la description de vos ressources, veuillez consulter.Décrire vos AWS IoT Core Ressources LoRaWAN (p. 1066).
- Profils d'appareil et de service

  HAVERcertains paramètres de configuration de périphériques sans fil prêts qui sont partagés par de nombreux périphériques et peuvent être stockés dans AWS IoT Core pour LoRaWan en tant que profils d'appareil et de service. Les paramètres de configuration se trouvent dans la documentation du périphérique ou sur l'appareil lui-même. Vous devez identifier un profil de périphérique correspondant aux paramètres de configuration du périphérique, ou en créer un si nécessaire, avant d'ajouter

le périphérique. Pour plus d'informations, consultez Ajouter des profils à AWS IoT Core pour LoRaWAN (p. 1077).

- AWS IoT Core pour LoRaWAN

  Chaque périphérique doit être affecté à une destination qui traitera ses messages à envoyer àAWS IoTet autres services. La .AWS IoTqui traitent et envoient les messages de l'appareil sont spécifiques au format de message du périphérique. Pour traiter les messages de l'appareil et les envoyer au bon service, identifiez la destination que vous allez créer à utiliser avec les messages de l'appareil et affectez-la au périphérique.

Pour intégrer votre appareil sans fil à AWS IoT Core pour LoRaWAN

- Ajoutez votre appareil sans fil à AWS IoT Core pour LoRaWAN (p. 1075)
- Ajouter des profils à AWS IoT Core pour LoRaWAN (p. 1077)
- Ajouter des destinations à AWS IoT Core pour LoRaWAN (p. 1079)
- Créer des règles pour traiter les messages de l'appareil LoRaWan (p. 1082)
- Connect votre appareil LoRaWan et vérifiez son état de connexion (p. 1084)

Ajoutez votre appareil sans fil à AWS IoT Core pour LoRaWAN

Si vous ajoutez votre appareil sans fil pour la première fois, nous vous recommandons d'utiliser la console. Accédez à . AWS IoT Core pour LoRaWAN IntroductionPage de laAWS IoT, choisissezMise en route, puisAjouter un appareil. Si vous avez déjà ajouté un appareil, choisissezAfficher l'appareilpour afficher la passerelle que vous avez ajoutée. Si vous souhaitez ajouter d'autres appareils, choisissezAjouter un appareil.

Vous pouvez également ajouter des périphériques sans fil à partir duAppareilsPage de laAWS IoTconsole

Ajoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de la console

Choisissez unSpécification du périphérique sans filen fonction de votre méthode d'activation et de la version LoRaWan. Une fois sélectionnées, vos données sont chiffrées avec une cléAWSpossède et gère pour vous.

Modes d'activation OTAA et ABP

Avant que votre appareil LoRaWan puisse envoyer des données de liaison montante, vous devez effectuer un processus appeléactivationouprocédure de jointure. Pour activer votre appareil, vous pouvez soit utiliser OTAA (Over the Air Activation), soit ABP (Activation par personnalisation).

ABP ne nécessite pas de procédure de jointure et utilise des clés statiques. Lorsque vous utilisez OTAA, votre appareil LoRaWan envoie une demande de jointure et le serveur réseau peut l'autoriser. Nous vous recommandons d'utiliser OTAA pour activer votre appareil car de nouvelles clés de session sont générées pour chaque activation, ce qui le rend plus sécurisé.

Version LoRaWAN

Lorsque vous utilisez OTAA, votre appareil LoRaWan et les applications hébergées dans le cloud partagent les clés racine. Ces clés racine dépendent de l'utilisation de la version v1.0.x ou v1.1. v1.0.x n'a qu'une seule clé racine,AppKey(Clé d'application) alors que v1.1 a deux clés racine,AppKey(Clé d'application) etNwkKey(clé réseau). Les clés de session sont dérivées en fonction des clés racine pour chaque activation. Les deuxNwkKeyandAppKeysont des valeurs hexadécimales de 32 chiffres fournies par votre fournisseur de services sans fil.

IUE des périphériques sans fil

Une fois que vous avez sélectionné leSpécification du périphérique sans fil, les paramètres EUI (Extended Unique Identifier) du périphérique sans fil s'affichent sur la console. Vous pouvez trouver ces informations dans la documentation de l'appareil ou du fournisseur de services sans fil.

- DevEui : valeur hexadémique à 16 chiffres qui est unique à votre appareil et qui figure sur l'étiquette de l'appareil ou sa documentation.
- AppEui : valeur hexadémique à 16 chiffres unique au serveur de jointure et trouvée dans la documentation du périphérique. Dans LoRaWAN version v1.1,AppEuiest appelée commeJoinEui.

Pour plus d'informations sur les identificateurs uniques, les clés de session et les clés racine, reportez-vous à la sectionAlliance LoRa.

Ajoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de l'API

Si vous ajoutez un périphérique sans fil à l'aide de l'API, vous devez d'abord créer votre profil de périphérique et votre profil de service avant de créer le périphérique sans fil. Vous utiliserez le profil de l'appareil et l'ID de profil de service lors de la création du périphérique sans fil. Pour plus d'informations sur la création de ces profils avec l'API, consultez.Ajouter un profil de périphérique à l'aide de l'API (p. 1077).

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoTActions API sans fil pour les profils de service

- CreateWirelessDevice
- GetWirelessDevice
- ListWirelessDevices
- UpdateWirelessDevice
- SupprimerWirelessDevice

Pour obtenir la liste complète des actions et types de données disponibles pour la création et la gestion AWS IoT Core Pour les ressources LoRaWAN, consultezAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIPour créer un dispositif sans fil

Vous pouvez utiliser la stratégieAWS CLIpour créer un périphérique sans fil à l'aide de l'outilcréation-périphérique sans filCommande l'. L'exemple suivant crée un périphérique sans fil à l'aide d'un fichier input.json pour entrer les paramètres.

> **Note**
>
> Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

Contenu de input.json

```
{
    "Description": "My LoRaWAN wireless device"
    "DestinationName": "IoTWirelessDestination"
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
```

```
        "DevEui": "ac12efc654d23fc2"
    },
    "Name": "SampleIoTWirelessThing"
    "Type": LoRaWAN
}
```

Vous pouvez fournir ce fichier en entrée à la`create-wireless-device`Commande l'.

```
aws iotwireless create-wireless-device \
    --cli-input-json file://input.json
```

Pour de plus amples informations sur les CLI que vous pouvez utiliser, veuillez consulter.AWS CLIRéférence

Ajouter des profils à AWS IoT Core pour LoRaWAN

Les profils de périphérique et de service peuvent être définis pour décrire les configurations de périphériques courantes. Ces profils décrivent les paramètres de configuration partagés par les périphériques pour faciliter l'ajout de ces périphériques. AWS IoT Core pour LoRaWan prend en charge les profils de périphériques et les profils de service.

Les paramètres de configuration et les valeurs à entrer dans ces profils sont fournis par le fabricant de l'appareil.

Ajouter des profils d'appareils

Les profils de périphérique définissent les capacités du périphérique et les paramètres de démarrage utilisés par le serveur réseau pour définir le service d'accès radio LoRaWan. Il comprend la sélection de paramètres tels que la bande de fréquences LoRa, la version des paramètres régionaux LoRa et la version MAC de l'appareil. Pour en savoir plus sur les différentes bandes de fréquences, consultezConsidérez la sélection de bandes de fréquences LoRa pour vos passerelles et votre connexion de périphérique (p. 1068).

Ajouter un profil de périphérique à l'aide de la console

Si vous ajoutez un périphérique sans fil à l'aide de la console, comme décrit dans la sectionAjoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de la console (p. 1075), après avoir ajouté la spécification du périphérique sans fil, vous pouvez ajouter votre profil de périphérique. Vous pouvez également ajouter des périphériques sans fil à partir duProfilsPage de laAWS IoTsur la consoleLoRaWANOnglet.

Vous pouvez choisir parmi les profils de périphérique par défaut ou créer un nouveau profil de périphérique. Nous vous recommandons d'utiliser les profils d'appareil par défaut. Si votre application vous demande de créer un profil de périphérique, fournissez unNom du profil de périphérique, sélectionnezBande de fréquence (RFRegion)que vous utilisez pour le périphérique et la passerelle, et conservez les autres paramètres aux valeurs par défaut, sauf indication contraire dans la documentation du périphérique.

Ajouter un profil de périphérique à l'aide de l'API

Si vous ajoutez un périphérique sans fil à l'aide de l'API, vous devez créer votre profil de périphérique avant de créer le périphérique sans fil.

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoTActions API sans fil pour les profils de service

- CreateDeviceProfile
- GetDeviceProfile

- ListDeviceProfiles
- UpdateDeviceProfile
- SupprimerDeviceProfile

Pour obtenir la liste complète des actions et types de données disponibles pour la création et la gestion AWS IoT Core Pour les ressources LoRaWAN, consultezAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIPour créer un profil d'appareil

Vous pouvez utiliser la stratégieAWS CLIPour créer un profil d'appareil à l'aide de l'créer un profil de périphériqueCommande l'. L'exemple suivant crée un profil de périphérique.

```
aws iotwireless create-device-profile
```

L'exécution de cette commande crée automatiquement un profil de périphérique avec un ID que vous pouvez utiliser lors de la création du périphérique sans fil. Vous pouvez maintenant créer le profil de service à l'aide de l'API suivante, puis créer le périphérique sans fil à l'aide des profils de périphérique et de service.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Pour de plus amples informations sur les CLI que vous pouvez utiliser, veuillez consulter.AWS CLIRéférence

Ajouter des profils de service

Les profils de service décrivent les paramètres de communication dont le périphérique a besoin pour communiquer avec le serveur d'applications.

Ajouter un profil de service à l'aide de la console

Si vous ajoutez un périphérique sans fil à l'aide de la console, comme décrit dans la sectionAjoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de la console (p. 1075), après avoir ajouté le profil de l'appareil, vous pouvez ajouter votre profil de service. Vous pouvez également ajouter des périphériques sans fil à partir duProfilsPage de laAWS IoTsur la consoleLoRaWANOnglet.

Nous vous recommandons de laisser le paramètreAddGwMetadataactivé afin que vous receviez des métadonnées de passerelle supplémentaires pour chaque charge utile, telles que RSSI et SNR pour la transmission de données.

Ajouter un profil de service à l'aide de l'API

Si vous ajoutez un périphérique sans fil à l'aide de l'API, vous devez d'abord créer votre profil de service avant de créer le périphérique sans fil.

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoTActions API sans fil pour les profils de service

- CreateServiceProfile
- GetServiceProfile

- ListServiceProfiles
- UpdateServiceProfile
- DeleteServiceProfile

Pour obtenir la liste complète des actions et types de données disponibles pour la création et la gestion AWS IoT Core Pour les ressources LoRaWAN, consultezAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIpour créer un profil de service

Vous pouvez utiliser la stratégieAWS CLIPour créer un service à l'aide de l'create-service-profileCommande l'. L'exemple suivant crée un profil de service.

```
aws iotwireless create-service-profile
```

L'exécution de cette commande crée automatiquement un profil de service avec un ID que vous pouvez utiliser lors de la création du périphérique sans fil. Vous pouvez désormais créer le périphérique sans fil à l'aide des profils de périphérique et de service.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

Ajouter des destinations à AWS IoT Core pour LoRaWAN

AWS IoT Core pour les destinations LoRaWan décrivent lesAWS IoTqui traite les données d'un périphérique pour les utiliser parAWSServices .

Parce que la plupart des appareils LoRaWan n'envoient pas de données à AWS IoT Core Pour LoraWan dans un format que peut utiliser parAWSServices, unAWS IoTdoit d'abord la traiter. La .AWS IoTcontient l'instruction SQL qui interprète les données du périphérique et les actions de règle de rubrique qui envoient le résultat de l'instruction SQL aux services qui l'utiliseront.

Ajouter une destination à l'aide de la console

Si vous ajoutez un périphérique sans fil à l'aide de la console, comme décrit dans la sectionAjoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de la console (p. 1075), une fois que vous avez déjà ajouté la spécification et les profils du périphérique sans fil à AWS IoT Core pour LoRaWan comme décrit précédemment, vous pouvez aller de l'avant et ajouter une destination.

Vous pouvez également ajouter un kit AWS IoT Core pour la destination LoRaWan à partir duDestinationsPage de laAWS IoTconsole

Pour traiter les données d'un périphérique, spécifiez les champs suivants lors de la création d'un AWS IoT Core pour la destination LoRaWan, puis choisissezAjouter une destination.

- Détails de la destination

  Saisissez unNom de destinationUne description facultative de votre destination.
- Nom de la règle

  La .AWS IoTqui est configurée pour traiter les données du périphérique. Votre destination aura besoin d'une règle pour traiter les messages qu'elle reçoit. Entrez un nom de règle, puis choisissezCopierpour copier le nom de la règle que vous allez entrer lors de la création de laAWS IoTRègle. Vous avez le choix entre les optionsCréer une règlePour créer la règle maintenant ou accédez à l'RèglesHub de laAWS IoTet créez une règle portant ce nom.

Pour plus d'informations surAWS IoTpour les destinations, consultezCréer des règles pour traiter les messages de l'appareil LoRaWan (p. 1082).

- Nom de rôle

  Rôle IAM qui donne aux données de l'appareil l'autorisation d'accéder à la règle nommée dansRule name (Nom de la règle). Pour de plus amples informations sur les détails qu'une définition requiert dans le rôle, veuillez consulter.Créer un rôle IAM pour vos destinations (p. 1080)

  Pour de plus amples informations sur les rôles IAM, veuillez consulterUtilisation de rôles IAM.

Ajouter une destination à l'aide de l'API

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'une destination.

AWS IoTActions API sans fil pour les profils de service

- CreateDestination
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

Pour obtenir la liste complète des actions et types de données disponibles pour créer et gérer AWS IoT Core Pour les ressources LoRaWAN, consultezAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIpour ajouter une destination

Vous pouvez utiliser la stratégieAWS CLIPour ajouter une destination à l'aide de l'Créer la destinationCommande l'. L'exemple suivant crée une destination.

```
aws iotwireless create-destination \
    --name IoTWirelessDestination \
    --expression-type RuleName \
    --expression IoTWirelessRule \
    --role-arn arn:aws:iam::123456789012:role/IoTWirelessDestinationRole
```

L'exécution de cette commande crée une destination avec le nom de destination, le nom de règle et le nom de rôle spécifiés. Pour plus d'informations sur les noms de règles et de rôles pour les destinations, voirCréer des règles pour traiter les messages de l'appareil LoRaWan (p. 1082)andCréer un rôle IAM pour vos destinations (p. 1080).

Pour de plus amples informations sur les CLI que vous pouvez utiliser, veuillez consulter.AWS CLIRéférence.

Créer un rôle IAM pour vos destinations

AWS IoT Core pour les destinations LoRaWan nécessitent des rôles IAM qui donnent AWS IoT Core pour LoRaWan les autorisations nécessaires pour envoyer des données à laAWS IoTRègle. Si un tel rôle n'est pas déjà défini, vous devrez le définir afin qu'il apparaisse dans la liste des rôles.

Pour créer une stratégie IAM pour votre AWS IoT Core pour LoRaWAN

1. Ouverture d'Hub de stratégies de la console IAM.
2. ChoisissezCréer une stratégie, puis choisissez l'optionJSONOnglet.

3. Dans l'éditeur, supprimez tout contenu de l'éditeur et collez ce document de stratégie.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeEndpoint",
                "iot:Publish"
            ],
            "Resource": "*"
        }
    ]
}
```

4. ChoisissezExaminer une stratégie, et dansNom, attribuez un nom à cette stratégie. Vous aurez besoin de ce nom à utiliser dans la procédure suivante.

   Vous pouvez également décrire cette stratégie dansDescription, si vous le souhaitez.
5. Choisissez Créer une stratégie.

Pour créer un rôle IAM pour un AWS IoT Core pour LoRaWAN

1. Ouverture d'Hub de rôles de la console IAMet choisissezCréation d'un rôle.
2. DansZone Select type of trusted entity, choisissezAutre Compte AWS .
3. DansID de compte, entrez votre Compte AWS ID, puis choisissezSuivant: Permissions (Autorisations).
4. Dans la zone de recherche, saisissez le nom de la stratégie IAM que vous avez créée lors de la procédure précédente.
5. Dans les résultats de la recherche, vérifiez la stratégie IAM que vous avez créée lors de la procédure précédente.
6. Choisissez Next (Suivant) Tags (Balises), puisSuivant: Review (Examiner).
7. DansNom de rôle, saisissez le nom de ce rôle, puis cliquez surCréation d'un rôle.
8. Dans le message de confirmation, choisissez le nom du rôle que vous avez créé pour modifier le nouveau rôle.
9. DansRécapitulatif, choisissez leRelations d'approbation, puis.Modification de la relation d'approbation.
10. DansDocument de stratégie, modifiez`Principal`se présente ainsi.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

    Une fois que vous avez modifié le`Principal`, le document de stratégie complet doit se présenter comme cet exemple.

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

```
  ]
}
```

11. Pour enregistrer vos modifications et quitter, choisissezStratégie d'approbation.

Lorsque ce rôle est défini, vous pouvez le trouver dans la liste des rôles lorsque vous configurez votre AWS IoT Core pour LoRaWAN

Créer des règles pour traiter les messages de l'appareil LoRaWan

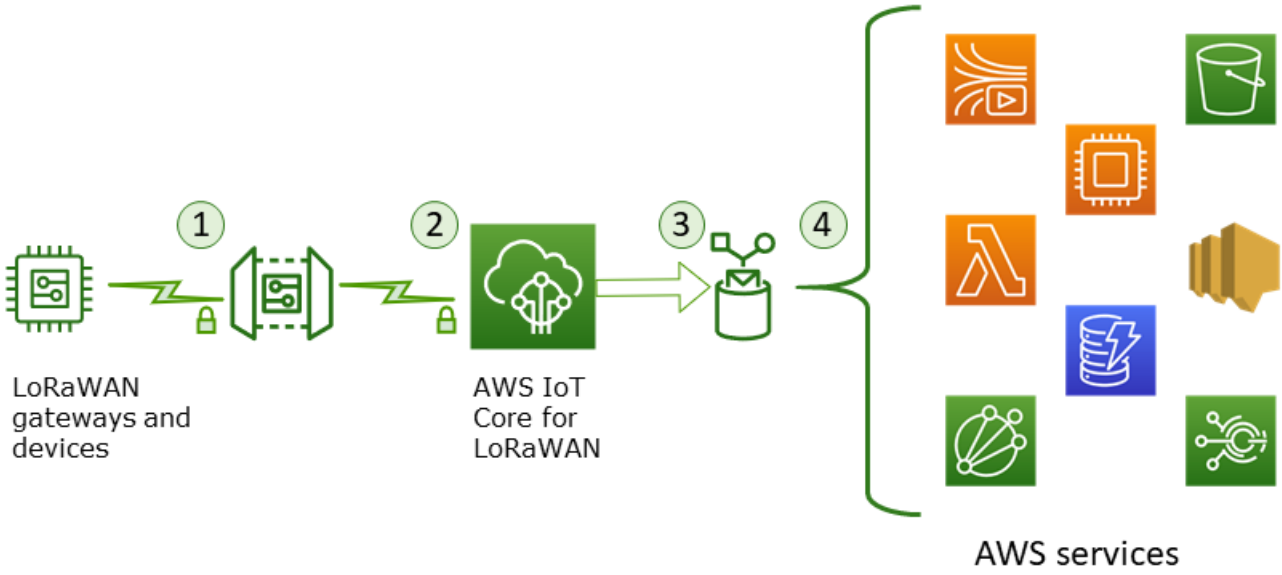
AWS IoTenvoient des messages de périphérique à d'autres services.AWS IoTpeuvent également traiter les messages binaires reçus d'un périphérique LoRaWan pour convertir les messages vers d'autres formats qui peuvent les rendre plus faciles à utiliser pour d'autres services.

AWS IoT Core pour LoRaWAN (p. 1079)associer un périphérique sans fil à la règle qui traite les données de message du périphérique à envoyer à d'autres services. La règle agit sur les données de l'appareil dès que AWS IoT Core pour LoraWan le reçoit. AWS IoT Core pour LoRaWAN (p. 1079)peuvent être partagés par tous les appareils dont les messages ont le même format de données et qui envoient leurs données au même service.

ProcéduresAWS IoTrègles traiter les messages de périphérique

Comment unAWS IoTtraite les données de message d'un périphérique dépendent du service qui recevra les données, du format des données de message du périphérique et du format de données requis par le service. En règle générale, la règle appelle unAWS Lambdapour convertir les données de message du périphérique au format requis par un service, puis envoie le résultat au service.

L'illustration suivante montre comment les données des messages sont sécurisées et traitées au fur et à mesure qu'elles passent du périphérique sans fil à unAWSservice.



1. Le périphérique sans fil LoRaWan crypte ses messages binaires en utilisant le mode CTR AES128 avant de les transmettre.
2. AWS IoT Core pour LoraWan déchiffre le message binaire et code la charge utile du message binaire déchiffré sous la forme d'une chaîne base64.

3. Le message encodé en base64 résultant est envoyé sous la forme d'une charge utile de message binaire (une charge utile de message qui n'est pas formatée en tant que document JSON) auAWS IoTdécrite dans la destination affectée au périphérique.
4. La .AWS IoTdirige les données du message vers le service décrit dans la configuration de la règle.

La charge utile binaire chiffrée reçue du périphérique sans fil n'est ni altérée ni interprétée par AWS IoT Core pour LoRaWAN. La charge utile de message binaire déchiffrée est encodée uniquement en tant que chaîne de base64. Pour que les services puissent accéder aux éléments de données de la charge utile de message binaire, les éléments de données doivent être analysés hors de la charge utile par une fonction appelée par la règle. La charge utile de message codée en base64 est une chaîne ASCII, elle peut donc être stockée en tant que telle pour être analysée ultérieurement.

Création de règles pour LoRaWAN

AWS IoT Core pour LoRaWAN utiliseAWS IoTpour envoyer en toute sécurité des messages de l'appareil directement à d'autresAWSsans avoir besoin d'utiliser le courtier de messages. En supprimant le courtier de messages du chemin d'ingestion, il réduit les coûts et optimise le flux de données.

Pour un AWS IoT Core pour que la règle LoRaWan envoie des messages de périphérique à d'autresAWS, il nécessite un AWS IoT Core pour la destination LoRaWan et unAWS IoTattribuée à cette destination. La .AWS IoTdoit contenir une instruction de requête SQL et au moins une action de règle.

En règle générale, la stratégieAWS IoTL'instruction requête de requête de règle se présente ainsi :

- Clause SQL SELECT qui sélectionne et met en forme les données de la charge utile du message
- Filtre de rubrique (objet FROM dans l'instruction de requête de règle) qui identifie les messages à utiliser
- Instruction conditionnelle facultative (clause SQL WHERE) qui spécifie les conditions sur lesquelles agir

Voici un exemple d'instruction de requête de règle :

```
SELECT temperature FROM iot/topic' WHERE temperature > 50
```

Lors de la constructionAWS IoTpour traiter les charges utiles des périphériques LoRaWan, vous n'avez pas besoin de spécifier la clause FROM dans le cadre de l'objet de requête de règle. L'instruction de requête de règle doit avoir la clause SQL SELECT et peut éventuellement avoir la clause WHERE. Si l'instruction query utilise la clause FROM, elle est ignorée.

Voici un exemple d'instruction de requête de règle qui peut traiter les charges utiles des appareils LoRaWan :

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
       PayloadData
```

Dans cet exemple, la méthode`PayloadData`est une charge utile binaire codée en base64 envoyée par votre appareil LoRaWan.

Voici un exemple d'instruction de requête de règle qui peut effectuer un décodage binaire de la charge utile entrante et la transformer en un format différent tel que JSON :

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,
       WirelessMetadata.LoRaWAN.DevEui as DevEui,
       aws_lambda("arn:aws:lambda:<region>:<account>:function:<name>",

                  {
                   "PayloadData":PayloadData,
                   "Fport": WirelessMetadata.LoRaWAN.FPort
```

```
                }) as decodingoutput
```

Pour plus d'informations sur l'utilisation des clauses SELECT AND WHERE, consultez.Référence SQL AWS IoT (p. 482)

Pour de plus amples informations surAWS IoTet de la façon de les créer et de les utiliser, consultezRègles pour AWS IoT (p. 394)andDidacticiels concernant les règles AWS IoT (p. 131).

Pour plus d'informations sur la création et l'utilisation AWS IoT Core pour les destinations LoRaWan, consultezAjouter des destinations à AWS IoT Core pour LoRaWAN (p. 1079).

Pour plus d'informations sur l'utilisation des charges utiles de messages binaires dans une règle, voirUtilisation des charges utiles binaires (p. 544).

Pour plus d'informations sur la sécurité des données et le chiffrement utilisés pour protéger la charge utile du message sur son parcours, consultezProtection des données dans AWS IoT Core (p. 312).

Pour obtenir une architecture de référence qui présente un exemple de décodage binaire et d'implémentation pour les règles IoT, voir AWS IoT Core Pour des échantillons de solution LoaWan sur GitHub.

Connect votre appareil LoRaWan et vérifiez son état de connexion

Avant de pouvoir vérifier l'état de la connexion de l'appareil, vous devez avoir déjà ajouté votre appareil et l'avoir connecté à AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la façon d'ajouter votre appareil, consultez la pageAjoutez votre appareil sans fil à AWS IoT Core pour LoRaWAN (p. 1075).

Après avoir ajouté votre appareil, reportez-vous au manuel d'utilisation de votre appareil pour savoir comment lancer l'envoi d'un message de liaison montante à partir de votre appareil LoRaWan.

Vérifier l'état de la connexion du périphérique à l'aide de

Pour vérifier l'état de la connexion à l'aide de la console, accédez à l'ongletAppareilsPage de laAWS IoTet choisissez l'appareil que vous avez ajouté. DansDétailsde la page Détails des périphériques sans fil, vous verrez la date et l'heure de réception de la dernière liaison montante.

Vérifier l'état de la connexion du périphérique à l'aide de

Pour vérifier l'état de la connexion à l'aide de l'API, utilisez le`GetWirelessDeviceStatistics`API. Cette API n'a pas de corps de requête et contient uniquement un corps de réponse qui indique quand la dernière liaison montante a été reçue.

```
HTTP/1.1 200
Content-type: application/json

{
  “LastUplinkReceivedAt”: “2021-03-24T23:13:08.476015749Z”,
  "LoRaWAN": {
        "DataRate": 5,
        "DevEui": "647fda0000006420",
        "Frequency": 868100000
        "Gateways": [
         {
            "GatewayEui": "c0ee40ffff29df10",
            "Rssi": -67,
            "Snr": 9.75
         }
      ],
  “WirelessDeviceId”: “30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
```

```
}
```

Étapes suivantes

Maintenant que vous avez connecté votre appareil et vérifié l'état de la connexion, vous pouvez observer le format des métadonnées de liaison montante reçues à partir du périphérique à l'aide de l'outilClient de test MQTTsur leTestPage de laAWS IoTconsole Pour plus d'informations, consultez Afficher le format des messages de liaison montante envoyés à partir d'appareils LoRaWan (p. 1100).

Gestion des passerelles avec AWS IoT Core pour LoRaWAN

Les passerelles agissent comme un pont et transportent les données de l'appareil LoRaWan vers et depuis un serveur réseau, généralement sur des réseaux à large bande passante tels que Wi-Fi, Ethernet ou Cellular. Les passerelles LoRaWan connectent des périphériques sans fil à AWS IoT Core pour LoRaWAN.

Voici quelques considérations importantes lors de l'utilisation de vos passerelles avec AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la façon d'ajouter votre passerelle à AWS IoT Core pour LoRaWAN, consultezÀ bord de vos passerelles vers AWS IoT Core pour LoRaWAN (p. 1068).

LoRaBasics™Configuration logicielle requise

Pour connecter à AWS IoT Core pour LoRaWan, votre passerelle LoRaWan doit avoir un logiciel appeléPrincipes de base LoRaMen cours d'exécution dessus. LoRaLes bases™Station est un logiciel open source géré par Semtech Corporation et distribué par leurGitHubrepository. AWS IoT Core pour LoRaWAN prend en charge LoRaWANLes bases™Station version 2.0.4 et ultérieure.

Utilisation de passerelles qualifiées à partir de laAWSCatalogue des appareils partenaires

La .AWSCatalogue des appareils partenairescontient des passerelles et des kits de développement qualifiés pour une utilisation avec AWS IoT Core pour LoRaWAN. Nous vous recommandons d'utiliser ces passerelles qualifiées car vous n'avez pas à modifier le logiciel d'intégration pour connecter les passerelles à AWS IoT Core . Ces passerelles disposent déjà d'une version du logiciel BasicStation compatible avec AWS IoT Core pour LoRaWAN.

> Note
>
> Si vous disposez d'une passerelle qui n'est pas répertoriée dans le catalogue de partenaires en tant que passerelle qualifiée avec AWS IoT Core pour LoRaWan, vous pouvez toujours l'utiliser si la passerelle exécute le logiciel LoRa Basics Station avec la version 2.0.4 et ultérieure. Assurez-vous que vous utilisezAuthentification du serveur et du client TLSpour l'authentification de votre passerelle LoRaWan.

Utilisation des protocoles CUPS et LNS

LoRaLes bases™Le logiciel Station contient deux sous-protocoles pour la connexion des passerelles aux serveurs réseau, le serveur réseau LoRaWan (LNS) et le protocole CUPS (Configuration and Update Server).

Le protocole LNS établit une connexion de données entre un LoRaLes bases™Passerelle compatible avec la station et un serveur réseau. Les messages de liaison montante et de liaison descendante LoRa sont échangés via cette connexion de données via des WebSockets sécurisés.

Le protocole CUPS permet la gestion des informations d'identification, la configuration à distance et la mise à jour du microprogramme des passerelles. AWS IoT Core pour LoRaWan fournit à la fois des points de terminaison LNS et CUPS pour l'ingestion de données LoRaWan et la gestion de passerelle distante respectivement.

Pour de plus amples informations, veuillez consulterProtocole LNSandProtocole CUPS.

Configurer les sous-bandes et les capacités de filtrage de votre passerelle

Les passerelles LoRaWan exécutent unPrincipes de base LoRaMqui permet aux passerelles de se connecter à AWS IoT Core pour LoRaWAN. Pour connecter à AWS IoT Core pour LoRaWan, votre passerelle LoRa interroge d'abord le serveur CUPS pour le point de terminaison LNS, puis établit une connexion de données WebSockets avec ce point de terminaison. Une fois la connexion établie, les trames de liaison montante et descendante peuvent être échangées via cette connexion.

Filtrage des trames de données LoRa reçues par la passerelle

Une fois que votre passerelle LoRaWan a établi une connexion au point de terminaison, AWS IoT Core pour LoraWan répond avec un`router_config`qui spécifie un ensemble de paramètres pour la configuration de la passerelle LoRa, y compris les paramètres de filtrage`NetID`and`JoinEui`. Pour plus d'informations sur`router_config`et comment une connexion est établie avec le serveur réseau LoRaWan (LNS), consultezProtocole LNS.

```
{
"msgtype"    : "router_config"
"NetID"      : [ INT, .. ]
"JoinEui"    : [ [INT,INT], .. ] // ranges: beg,end inclusive
"region"     : STRING             // e.g. "EU863", "US902", ..
"hwspec"     : STRING
"freq_range" : [ INT, INT ]       // min, max (hz)
"DRs"        : [ [INT,INT,INT], .. ]   // sf,bw,dnonly
"sx1301_conf": [ SX1301CONF, .. ]
"nocca"      : BOOL
"nodc"       : BOOL
"nodwell"    : BOOL
}
```

Les passerelles transportent les données des périphériques LoRaWan vers et depuis LNS généralement sur des réseaux à large bande passante tels que Wi-Fi, Ethernet ou Cellular. Les passerelles ramassent généralement tous les messages et passent par le trafic qui lui arrive à AWS IoT Core pour LoRaWAN. Toutefois, vous pouvez configurer les passerelles pour filtrer une partie du trafic de données de l'appareil, ce qui permet de conserver l'utilisation de la bande passante et de réduire le flux de trafic entre la passerelle et LNS.

Pour configurer votre passerelle LoRa pour filtrer les blocs de données, vous pouvez utiliser les paramètres`NetID`and`JoinEui`dans le`router_config`Message.`NetID`est une liste de valeurs NetID acceptées. Tout bloc de données LoRa portant un bloc de données autre que ceux répertoriés sera supprimé.`JoinEui`est une liste de paires de valeurs entières codant des plages de valeurs JoineUI. Les trames de requête de jointure seront supprimées par la passerelle à moins que le champ`JoinEui`dans le message se trouve dans la plage [BegeUI, EnduUI].

Canaux de fréquence et sous-bandes

Pour les régions RF US915 et AU915, les périphériques sans fil ont le choix entre 64 canaux de liaison montante 125kHz et 8 500kHz pour accéder aux réseaux LoRaWan à l'aide des passerelles LoRa. Les canaux de fréquence de liaison montante sont divisés en 8 sous-bandes, chacune avec 8 canaux 125kHz

et un canal 500kHz. Pour chaque passerelle régulière dans la région AU915, un ou plusieurs sous-canaux seront pris en charge.

Certains périphériques sans fil ne peuvent pas sauter entre les sous-bandes et utiliser les canaux de fréquence dans une seule sous-bande lorsqu'ils sont connectés à AWS IoT Core pour LoRaWAN. Pour que les paquets de liaison montante de ces périphériques soient transmis, configurez les passerelles LoRa pour qu'elles utilisent ce sous-canal particulier. Pour les passerelles situées dans d'autres régions RF, telles que EU868, cette configuration n'est pas requise.

Configurez votre passerelle pour utiliser le filtrage et les sous-bandes à l'aide de la console

Vous pouvez configurer votre passerelle pour utiliser un sous-canal particulier et activer également la possibilité de filtrer les blocs de données LoRa. Pour spécifier ces paramètres à l'aide de la console :

1. Accédez à . AWS IoT Core pour LoRaWAN PasserellesPage de laAWS IoTet choisissezAjout d'un.
2. Spécifiez les détails de la passerelle, tels queEui de la passerelle,Bande de fréquence (RFRegion)et unNomandDescription, et choisissez d'associer ou non unAWS IoTUne chose à votre passerelle. Pour plus d'informations sur la façon d'ajouter une passerelle, consultez la pageAjouter une passerelle à l'aide de la console (p. 1070).
3. DansConfiguration LoRaWAN, vous pouvez spécifier les sous-canaux et les informations de filtrage.
   - `SubBands` : Pour ajouter un sous-canal, choisissezAjout d'un sous-et spécifiez une liste de valeurs entières qui indiquent quels sous-canaux sont pris en charge par la passerelle. La .`SubBands`ne peut être configuré que dans l'`RfRegion`US915 et AU915 et doit avoir des valeurs dans la plage`[1,8]`Dans l'une de ces régions soutenues.
   - `NetIdFilters` : Pour filtrer les images de liaison montante, choisissezAjout d'un IDet spécifiez une liste de valeurs de chaîne utilisées par la passerelle. L'ID netId de la trame de liaison montante entrante du périphérique sans fil doit correspondre à au moins une des valeurs répertoriées, sinon la trame est supprimée.
   - `JoinEuiFilters`Choisissez :Ajouter la gamme JoineUIet spécifiez une liste de paires de valeurs de chaîne utilisées par une passerelle pour filtrer les trames LoRa. La valeur JoinEUI spécifiée dans le cadre de la demande de jointure à partir du périphérique sans fil doit se situer dans la plage d'au moins une des valeurs JoineUirange, chacune répertoriée sous la forme d'une paire de [BEGEUI, EnDEUI], sinon la trame est supprimée.
4. Vous pouvez ensuite continuer à configurer votre passerelle en suivant les instructions décrites dansAjouter une passerelle à l'aide de la console (p. 1070).

Une fois que vous avez ajouté une passerelle, dans la fenêtre AWS IoT Core pour LoRaWAN PasserellesPage de laAWS IoT, si vous sélectionnez la passerelle que vous avez ajoutée, vous pouvez voir la fenêtre`SubBands`et filtres`NetIdFilters`and`JoinEuiFilters`dans leDétails spécifiques à LoRaWANsection de la page de détails de la passerelle.

Configurez votre passerelle pour utiliser le filtrage et les sous-canaux avec l'API

Vous pouvez utiliser la stratégieCreateWirelessGatewayque vous utilisez pour créer une passerelle pour configurer les sous-canaux que vous souhaitez utiliser et activer la fonctionnalité de filtrage. Utilisation du kit`CreateWirelessGateway`, vous pouvez spécifier les sous-canaux et les filtres dans le cadre des informations de configuration de la passerelle que vous fournissez à l'aide de l'`LoRaWAN`field. Voici le jeton de demande qui inclut ces informations.

```
POST /wireless-gateways HTTP/1.1
Content-type: application/json
```

```
{
"Arn": "arn:aws:iotwireless:us-east-1:400232685877aa:WirelessGateway/
       a11e3d21-e44c-471c-afca-6716c228336a",
"Description": "Using my first LoRaWAN gateway",
   "LoRaWAN": {
      "GatewayEui": "a1b2c3d4567890ab",
      "JoinEuiFilters": [
        ["0000000000000001", "00000000000000ff"],
        ["000000000000ff00", "000000000000ffff"]
      ],
      "NetIdFilters": ["000000", "000001"],
      "RfRegion": "US915",
      "SubBands": [2]
   },
   "Name": "myFirstLoRaWANGateway"
   "ThingArn": null,
   "ThingName": null
}
```

Vous pouvez également utiliser laUpdateWirelessGatewaypour mettre à jour les filtres, mais pas les sous-canaux. Si l'icône`JoinEuiFilters`and`NetIdfilters`sont nulles, cela signifie qu'il n'y a pas de mise à jour pour les champs. Si les valeurs ne sont pas nulles et que des listes vides sont incluses, la mise à jour est appliquée. Pour obtenir les valeurs des champs que vous avez spécifiés, utilisez la commandeGetWirelessGatewayAPI.

Mettre à jour le firmware de la passerelle à l'aide du AWS IoT Core pour LoRaWAN

La .Principes de base LoRaMqui s'exécute sur votre passerelle fournit la gestion des informations d'identification et l'interface de mise à jour du microprogramme à l'aide du protocole CUPS (Configuration and Update Server). Le protocole CUPS fournit une mise à jour sécurisée du microprogramme avec des signatures ECDSA.

Vous devrez fréquemment mettre à jour le firmware de votre passerelle. Vous pouvez utiliser le service CUPS avec AWS IoT Core pour que LoRaWan fournisse des mises à jour du microprogramme à la passerelle où les mises à jour peuvent également être signées. Pour mettre à jour le firmware de la passerelle, vous pouvez utiliser le SDK ou l'interface de ligne de commande, mais pas la console.

Le processus de mise à jour dure environ 45 minutes. Cela peut prendre plus de temps si vous configurez votre passerelle pour la première fois pour vous connecter à AWS IoT Core pour LoRaWAN. Les fabricants de passerelle fournissent généralement leurs propres fichiers et signatures de mise à jour du microprogramme afin que vous puissiez l'utiliser à la place et passer àChargez le fichier du firmware dans un compartiment S3 et ajoutez un rôle IAM (p. 1092).

Si vous ne disposez pas des fichiers de mise à jour du micrologiciel, consultezGénérer le fichier de mise à jour et la signature (p. 1088)pour un exemple que vous pouvez utiliser pour vous adapter à votre application.

Pour effectuer la mise à jour du firmware de votre passerelle :

- Générer le fichier de mise à jour et la signature (p. 1088)
- Chargez le fichier du firmware dans un compartiment S3 et ajoutez un rôle IAM (p. 1092)
- Planifier et exécuter la mise à jour du microprogramme à l'aide d'une définition de tâche (p. 1095)

Générer le fichier de mise à jour et la signature

Les étapes de cette procédure sont facultatives et dépendent de la passerelle que vous utilisez. Les fabricants de passerelle fournissent leur propre mise à jour du firmware sous la forme d'un fichier de

mise à jour ou d'un script et Basics Station exécute ce script en arrière-plan. Dans ce cas, vous trouverez probablement le fichier de mise à jour du micrologiciel dans les notes de mise à jour de la passerelle que vous utilisez. Vous pouvez ensuite utiliser ce fichier ou ce script de mise à jour à la place et passer àChargez le fichier du firmware dans un compartiment S3 et ajoutez un rôle IAM (p. 1092).

Si vous ne disposez pas de ce script, la suite affiche les commandes à exécuter pour générer le fichier de mise à jour du microprogramme. Les mises à jour peuvent également être signées pour s'assurer que le code n'a pas été modifié ou endommagé et que les appareils exécutent du code publié uniquement par des auteurs de confiance.

Dans cette procédure, vous devez :

- Générer le fichier de mise à jour (p. 1089)
- Générer une signature pour la mise à jour du (p. 1091)
- Passez en revue les étapes suivantes (p. 1092)

Générer le fichier de mise à jour

Le logiciel LoRa Basics Station s'exécutant sur la passerelle est capable de recevoir les mises à jour du microprogramme dans la réponse CUPS. Si vous n'avez pas de script fourni par le fabricant, reportez-vous au script de mise à jour du micrologiciel suivant qui est écrit pour la passerelle Raspberry Pi RakWireless Gateway. Nous avons un script de base et la nouvelle station binaire, fichier de version, et`station.conf`y sont attachés.

> **Note**
>
> Le script est spécifique à RAKWireless Gateway, vous devrez donc l'adapter à votre application en fonction de la passerelle que vous utilisez.

Script de base

Voici un exemple de script de base pour Raspberry Pi basé sur RakWireless Gateway. Vous pouvez enregistrer les commandes suivantes dans un fichier`base.sh`, puis exécutez le script dans le terminal sur le navigateur Web de Raspberry Pi.

```
*#!/bin/bash*
execution_folder=/home/pi/Documents/basicstation/examples/aws_lorawan
station_path="$execution_folder/station"
version_path="$execution_folder/version.txt"
station_conf_path="$execution_folder/station_conf"

# Function to find the Basics Station binary at the end of this script
# and store it in the station path
function prepare_station()
{
 match=$(grep --text --line-number '^STATION:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
 match_end=$(grep --text --line-number '^END_STATION:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_version()
{
  match=$(grep --text --line-number '^VERSION:$' $0 | cut -d ':' -f 1)
  payload_start=$((match + 1))
  match_end=$(grep --text --line-number '^END_VERSION:$' $0 | cut -d ':' -f 1)
  payload_end=$((match_end - 1))
  lines=$(($payload_end-$payload_start+1))
```

```
  head -n $payload_end $0 | tail -n $lines  > $version_path
}

# Function to find the version.txt at the end of this script
# and store it in the location for version.txt
function prepare_station_conf()
{
 match=$(grep --text --line-number '^CONF:$' $0 | cut -d ':' -f 1)
 payload_start=$((match + 1))
 match_end=$(grep --text --line-number '^END_CONF:$' $0 | cut -d ':' -f 1)
 payload_end=$((match_end - 1))
 lines=$(($payload_end-$payload_start+1))
 head -n $payload_end $0 | tail -n $lines  > $station_conf_path
}

# Stop the currently running Basics station so that it can be overwritten
# by the new one
killall station

# Store the different files
prepare_station
prepare_versionp
prepare_station_conf

# Provide execute permission for Basics station binary
chmod +x $station_path

# Remove update.bin so that it is not read again next time Basics station starts
rm -f /tmp/update.bin

# Exit so that rest of this script which has binaries attached does not get executed
exit 0
```

Ajouter un script de charge utile

Au script de base, nous ajoutons le binaire Basics Station, le version.txt qui identifie la version à mettre à jour, et`station.conf`dans un script appelé`addpayload.sh`. Ensuite, exécutez ce script.

```
*#!/bin/bash
*
base.sh > fwstation

# Add station
echo "STATION:" >> fwstation
cat $1 >> fwstation
echo "" >> fwstation
echo "END_STATION:" >> fwstation

# Add version.txt
echo "VERSION:" >> fwstation
cat $2 >> fwstation
echo "" >> fwstation
echo "END_VERSION:" >> fwstation

# Add station.conf
echo "CONF:" >> fwstation
cat $3 >> fwstation
echo "END_CONF:" >> fwstation

# executable
chmod +x fwstation
```

Après avoir exécuté ces scripts, vous pouvez exécuter la commande suivante dans le terminal pour générer le fichier de mise à jour du firmware,`fwstation`.

```
$ ./addpayload.sh station version.txt station.conf
```

Générer une signature pour la mise à jour du

Le logiciel LoRa Basics Station fournit des mises à jour signées du firmware avec des signatures ECDSA. Pour prendre en charge les mises à jour signées, vous avez besoin des

- Signature qui doit être générée par une clé privée ECDSA et inférieure à 128 octets.
- Clé privée utilisée pour la signature et qui doit être stockée dans la passerelle avec le nom de fichier au format`sig-%d.key`. Nous vous recommandons d'utiliser le nom de fichier`sig-0.key`.
- Un CRC 32 bits sur la clé privée.

La signature et le CRC seront transmis au AWS IoT Core pour les API LoRaWAN. Pour générer les fichiers précédents, vous pouvez utiliser le script suivant`gen.sh`qui s'inspire de labasicstationexemple dans le référentiel GitHub.

```
*#!/bin/bash

*function ecdsaKey() {
    # Key not password protected for simplicity
    openssl ecparam -name prime256v1 -genkey | openssl ec -out $1
}

# Generate ECDSA key
ecdsaKey sig-0.prime256v1.pem

# Generate public key
openssl ec -in sig-0.prime256v1.pem -pubout -out sig-0.prime256v1.pub

# Generate signature private key
openssl ec -in sig-0.prime256v1.pub -inform PEM -outform DER -pubin | tail -c 64 >
 sig-0.key

# Generate signature
openssl dgst -sha512 -sign sig-0.prime256v1.pem $1 > sig-0.signature

# Convert signature to base64
openssl enc -base64 -in sig-0.signature -out sig-0.signature.base64

# Print the crc
crc_res=$(crc32 sig-0.key)printf "The crc for the private key=%d\n" $((16#$crc_res))

# Remove the generated files which won't be needed later
rm -rf sig-0.prime256v1.pem sig-0.signature sig-0.prime256v1.pub
```

La clé privée générée par le script doit être enregistrée dans la passerelle. Le fichier clé est au format binaire.

```
./gen_sig.sh fwstation
read EC key
writing EC key
read EC key
writing EC key
read EC key
writing EC key
The crc for the private key=3434210794

$ cat sig-0.signature.base64
```

```
MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBROen+XlIdMScv
AsfVfU/ZScJCalkVNZh4esyS8mNIgA==

$ ls sig-0.key
sig-0.key

$ scp sig-0.key pi@192.168.1.11:/home/pi/Documents/basicstation/examples/iotwireless
```

Passez en revue les étapes suivantes

Maintenant que vous avez généré le firmware et la signature, passez à la rubrique suivante pour télécharger le fichier du firmware,`fwstation`, dans un compartiment Amazon S3. Le compartiment est un conteneur qui stocke le fichier de mise à jour du microprogramme en tant qu'objet. Vous pouvez ajouter un rôle IAM qui donnera au serveur CUPS l'autorisation de lire le fichier de mise à jour du microprogramme dans le compartiment S3.

Chargez le fichier du firmware dans un compartiment S3 et ajoutez un rôle IAM

Vous pouvez utiliser Amazon S3 pour créer unbucket, qui est un conteneur qui peut stocker votre fichier de mise à jour du micrologiciel. Vous pouvez télécharger votre fichier dans le compartiment S3 et ajouter un rôle IAM qui permet au serveur CUPS de lire votre fichier de mise à jour à partir du compartiment. Pour de plus amples informations sur Amazon S3, veuillez consulterPremiers pas avec Amazon S3.

Le fichier de mise à jour du firmware que vous souhaitez importer dépend de la passerelle que vous utilisez. Si vous avez suivi une procédure similaire à celle décrite dansGénérer le fichier de mise à jour et la signature (p. 1088), vous allez télécharger le`fwstation`généré par l'exécution des scripts.

Cette procédure dure environ 20 minutes.

Pour télécharger votre fichier de firmware :

- Créez un compartiment Amazon S3 et chargez le fichier de mise à jour (p. 1092)
- Créer un rôle IAM avec les autorisations de lire le compartiment S3 (p. 1093)
- Passez en revue les étapes suivantes (p. 1094)

Créez un compartiment Amazon S3 et chargez le fichier de mise à jour

Vous allez créer un compartiment Amazon S3 à l'aide de l'AWS Management Console, puis téléchargez votre fichier de mise à jour du micrologiciel dans le compartiment.

Création d'un compartiment S3

Pour créer un compartiment S3, ouvrez l'Console Amazon S3. Connectez-vous si vous ne l'avez pas déjà fait, puis effectuez les opérations suivantes :

1. Choisissez Créer un compartiment.
2. Saisissez un nom unique et significatif pour leNom du compartiment, (par exemple,`iotwirelessfwupdate`). Pour connaître la convention d'attribution de noms recommandée pour votre compartiment, consultezhttps://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html.
3. Assurez-vous que vous avez sélectionné l'option Région AWS sélectionné comme celui que vous avez utilisé pour créer votre passerelle et votre périphérique LoRaWan, et leBlock all public access (Bloquer tous les accès publics)est sélectionné afin que votre compartiment utilise les autorisations par défaut.
4. ChoisissezActiver lespourCréation de versions du compartimentCe qui vous aidera à conserver plusieurs versions du fichier de mise à jour du firmware dans le même compartiment.
5. ConfirmerChiffrement côté serveura la valeurDésactiveret choisissezCréer un compartiment.

Chargez votre fichier de mise à jour du

Vous voyez désormais votre compartiment dans la liste des compartiments affichés dans la listeAWS Management Console. Choisissez votre compartiment et procédez comme suit pour télécharger votre fichier.

1. Choisissez votre compartiment, puis choisissezCharger.
2. ChoisissezAjouter un fichier, puis téléchargez le fichier de mise à jour du microprogramme. Si vous avez suivi la procédure décrite dansGénérer le fichier de mise à jour et la signature (p. 1088), vous allez télécharger le`fwstation`, sinon téléchargez le fichier fourni par le fabricant de votre passerelle.
3. Assurez-vous que tous les paramètres sont définis sur leur valeur par défaut. Vérifiez les éléments suivants :ACL prédéfiniesa la valeurprivéet choisissezChargerpour télécharger votre fichier.
4. Copiez l'URI S3 du fichier que vous avez téléchargé. Choisissez votre compartiment et vous verrez le fichier que vous avez téléchargé s'afficher dans la liste desObjets. Choisissez votre fichier, puis choisissezCopier URI S3. L'URI se présente sous la forme :`s3://iotwirelessfwupdate/fwstation`si vous avez nommé votre compartiment similaire à l'exemple décrit précédemment (`fwstation`). Vous utiliserez l'URI S3 lors de la création du rôle IAM.

Créer un rôle IAM avec les autorisations de lire le compartiment S3

Vous allez maintenant créer un rôle et une stratégie IAM qui donneront à CUPS l'autorisation de lire votre fichier de mise à jour du micrologiciel à partir du compartiment S3.

Créer une stratégie IAM pour votre rôle

Pour créer une stratégie IAM pour votre AWS IoT Core pour le rôle de destination LoRaWan, ouvrez leHub de stratégies de la console IAMEnsuite, procédez comme suit :

1. ChoisissezCréer une stratégie, puis choisissez l'optionJSONOnglet.
2. Supprimez tout contenu de l'éditeur et collez ce document de stratégie. La stratégie fournit des autorisations pour accéder au`iotwireless`et le fichier de mise à jour du microprogramme,`fwstation`, stocké à l'intérieur d'un objet.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:ListBucketVersions",
                "s3:ListBucket",
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::iotwirelessfwupdate/fwstation",
                "arn:aws:s3:::iotwirelessfwupdate"
            ]
        }
    ]
}
```

3. ChoisissezExaminer une stratégie, et dansNom, attribuez un nom à cette stratégie (par exemple,`IoTWirelessFwUpdatePolicy`). Vous aurez besoin de ce nom à utiliser dans la procédure suivante.
4. Choisissez Créer une stratégie.

Créer un rôle IAM avec la stratégie attachée

Vous allez maintenant créer un rôle IAM et attachez la stratégie précédemment créée pour accéder au compartiment S3. Ouverture d'Hub de rôles de la console IAMet procédez comme suit :

1. Sélectionnez Créer un rôle.
2. DansZone Select type of trusted entity, choisissezAutre Compte AWS .
3. DansID de compte, entrez votre Compte AWS ID, puis choisissezSuivant: Permissions (Autorisations).
4. Dans la zone de recherche, saisissez le nom de la stratégie IAM que vous avez créée lors de la procédure précédente. Vérifiez la stratégie IAM (par exemple,`IoTWirelessFwUpdatePolicy`) que vous avez créé précédemment dans les résultats de recherche et sélectionnez-le.
5. Choisissez Next (Suivant) Tags (Balises), puis.Suivant: Review (Examiner).
6. DansNom de rôle, saisissez le nom de ce rôle (par exemple,`IoTWirelessFwUpdateRole`), puis.Création d'un rôle.

Modifier la relation d'approbation du rôle IAM

Dans le message de confirmation qui s'affiche après l'étape précédente, choisissez le nom du rôle que vous avez créé pour le modifier. Vous modifiez le rôle pour ajouter la relation d'approbation suivante.

1. DansRécapitulatifdu rôle que vous avez créé, choisissez l'Relations d'approbation, puis.Modification de la relation d'approbation.
2. DansDocument de stratégie, modifiez`Principal`se présente ainsi.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

Une fois que vous avez modifié le`Principal`, le document de stratégie complet doit se présenter comme cet exemple.

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

3. Pour enregistrer vos modifications et quitter, choisissezStratégie d'approbation.
4. Obtenez l'ARN de votre rôle. Choisissez votre rôle IAM et dans la section Résumé, vous verrez unARN de rôle, comme`arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`. Copiez cetteARN de rôle.

Passez en revue les étapes suivantes

Maintenant que vous avez créé le compartiment S3 et un rôle IAM qui permet au serveur CUPS de lire le compartiment S3, passez à la rubrique suivante pour planifier et exécuter la mise à jour du microprogramme. Conserver lesS3 URIandARN de rôleque vous avez copié précédemment afin de pouvoir les saisir pour créer une définition de tâche qui sera exécutée pour effectuer la mise à jour du microprogramme.

Planifier et exécuter la mise à jour du microprogramme à l'aide d'une définition de tâche

Vous pouvez utiliser une définition de tâche pour inclure des détails sur la mise à jour du microprogramme et définir la mise à jour. AWS IoT Core pour LoRaWan fournit une mise à jour du microprogramme basée sur les informations des trois champs suivants associés à la passerelle.

- Station

  Version et temps de construction du logiciel Basics Station. Pour identifier ces informations, vous pouvez également les générer à l'aide du logiciel Basics Station exécuté par votre passerelle (par exemple,`2.0.5(rpi/std) 2021-03-09 03:45:09`).
- PackageVersion

  La version du firmware, spécifiée par le fichier`version.txt`dans la passerelle. Bien que ces informations ne soient pas présentes dans la passerelle, nous vous recommandons de les définir comme un moyen de définir la version de votre firmware (par exemple,`1.0.0`).
- Model

  Plateforme ou modèle utilisé par la passerelle (par exemple, Linux).

Cette procédure prend 20 minutes.

Pour suivre cette procédure :

- Obtenir la version actuelle en cours d'exécution sur votre passerelle (p. 1095)
- Création d'une définition de tâche de passerelle sans fil (p. 1096)
- Exécutez la tâche de mise à jour du micrologiciel et suivez la progression (p. 1097)

Obtenir la version actuelle en cours d'exécution sur votre passerelle

Pour déterminer l'éligibilité de votre passerelle à une mise à jour du microprogramme, le serveur CUPS vérifie les trois champs,`Station`,`PackageVersion`, et`Model`, pour une correspondance lorsque la passerelle les présente lors d'une requête CUPS. Lorsque vous utilisez une définition de tâche, ces champs sont stockés dans le`CurrentVersion`field.

Vous pouvez utiliser la stratégie AWS IoT Core pour LoRaWAN ouAWS CLIpour obtenir`CurrentVersion`pour votre passerelle. Les commandes suivantes montrent comment obtenir ces informations avec l'interface de ligne de commande.

1. Si vous avez déjà provisionné une passerelle, vous pouvez obtenir des informations sur la passerelle à l'aide de laget-sans-filCommande l'.

```
aws iotwireless get-wireless-gateway \
    --identifier 5a11b0a85a11b0a8 \
        --identifier-type GatewayEui
```

   Voici un exemple de sortie pour la commande.

```
{
    "Name": "Raspberry pi",
    "Id": "1352172b-0602-4b40-896f-54da9ed16b57",
    "Description": "Raspberry pi",
    "LoRaWAN": {
        "GatewayEui": "5a11b0a85a11b0a8",
```

```
        "RfRegion": "US915"
    },
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGateway/1352172b-0602-4b40-896f-54da9ed16b57"
}
```

2. À l'aide de l'ID de passerelle sans fil signalé par le`get-wireless-gateway`, vous pouvez utiliser la commandeget-sans-fil-gateway-firmware-informationspour obtenir la commande`CurrentVersion`.

```
aws iotwireless get-wireless-gateway-firmware-information \
    --id "3039b406-5cc9-4307-925b-9948c63da25b"
```

   Voici un exemple de sortie pour la commande, avec les informations des trois champs affichés par le`CurrentVersion`.

```
{
    "LoRaWAN": {
        "CurrentVersion": {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
        }
    }
}
```

Création d'une définition de tâche de passerelle sans fil

Lorsque vous créez la définition de tâche, nous vous recommandons de spécifier la création automatique de tâches à l'aide du paramètre AutoCreateTasks.`AutoCreateTasks`s'applique à toute passerelle qui a une correspondance pour les trois paramètres mentionnés précédemment. Si ce paramètre est désactivé, les paramètres doivent être affectés manuellement à la passerelle.

Vous pouvez créer la définition de tâche de passerelle sans fil à l'aide de l'outil AWS IoT Core pour LoRaWAN ouAWS CLI. Les commandes suivantes montrent comment créer la définition de tâche avec l'interface de ligne de commande.

1. Créez un fichier,`input.json`, qui contiendra les informations à transmettre à la`CreateWirelessGatewayTaskDefinition`API. Dans`input.json`, indiquez les informations suivantes que vous avez obtenues précédemment :
   - UpdateDataSource

     Fournissez le lien vers votre objet contenant le fichier de mise à jour du microprogramme que vous avez chargé dans le compartiment S3. (par exemple,`s3://iotwirelessfwupdate/fwstation`.
   - UpdateDataRole

     Fournissez le lien vers l'ARN de rôle pour le rôle IAM que vous avez créé, qui fournit des autorisations pour lire le compartiment S3. (par exemple,`arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole`.
   - SigkeyCRC et UpdateSignature

     Ces informations peuvent être fournies par le fabricant de votre passerelle, mais si vous avez suivi la procédure décrite dansGénérer le fichier de mise à jour et la signature (p. 1088), vous trouverez ces informations lors de la génération de la signature.
   - CurrentVersion

     Fournissez`CurrentVersion`que vous avez obtenu précédemment en exécutant la commande`get-wireless-gateway-firmware-information` Commande l'.

```
cat input.json
```

Voici le contenu de l'`input.json`dans le fichier.

```
{
    "AutoCreateTasks": true,
    "Name": "FirmwareUpdate",
    "Update":
    {
        "UpdateDataSource" : "s3://iotwirelessfwupdate/fwstation",
        "UpdateDataRole" : "arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole",
        "LoRaWAN" :
        {
            "SigKeyCrc": 3434210794,
            "UpdateSignature": "MEQCIDPY/p2ssgXIPNCOgZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBROen
+XlIdMScvAsfVfU/ZScJCalkVNZh4esyS8mNIgA==",
            "CurrentVersion" :
            {
            "PackageVersion": "1.0.0",
            "Model": "rpi",
            "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"
            }
        }
    }
}
```

2. Transmettre le`input.json`dans`création-passerelles sans fil-définition de tâches`commande pour créer la définition de tâche.

```
aws iotwireless create-wireless-gateway-task-definition \
    --cli-input-json file://input.json
```

Voici la sortie de la commande.

```
{
    "Id": "4ac46ff4-efc5-44fd-9def-e8517077bb12",
    "Arn": "arn:aws:iotwireless:us-
east-1:231894231068:WirelessGatewayTaskDefinition/4ac46ff4-efc5-44fd-9def-e8517077bb12"
}
```

Exécutez la tâche de mise à jour du micrologiciel et suivez la progression

La passerelle est prête à recevoir la mise à jour du firmware et, une fois mise sous tension, elle se connecte au serveur CUPS. Lorsque le serveur CUPS trouve une correspondance dans la version de la passerelle, il planifie une mise à jour du microprogramme.

Une tâche est une définition de tâche en cours. Comme vous avez spécifié la création automatique de tâches en définissant`AutoCreateTasks`sur`True`, la tâche de mise à jour du microprogramme démarre dès qu'une passerelle correspondante est trouvée.

Vous pouvez suivre l'avancement de la tâche à l'aide de la commande`GetWirelessGatewayTask`API. Lorsque vous exécutez le`get-sans-fil-gateway-task`la première fois, il affichera l'état de la tâche comme`IN_PROGRESS`.

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

Voici la sortie de la commande.

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "IN_PROGRESS"
}
```

Lorsque vous exécutez la commande la prochaine fois, si la mise à jour du firmware prend effet, elle affiche les champs mis à jour,`Package`,`Version`, et`Model`et l'état de la tâche passe à`COMPLETED`.

```
aws iotwireless get-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

Voici la sortie de la commande.

```
{
    "WirelessGatewayId": "1352172b-0602-4b40-896f-54da9ed16b57",
    "WirelessGatewayTaskDefinitionId": "ec11f9e7-b037-4fcc-aa60-a43b839f5de3",
    "LastUplinkReceivedAt": "2021-03-12T09:56:12.047Z",
    "TaskCreatedAt": "2021-03-12T09:56:12.047Z",
    "Status": "COMPLETED"
}
```

Dans cet exemple, nous vous avons montré la mise à jour du firmware à l'aide de la passerelle Raspberry Pi RakWireless. Le script de mise à jour du micrologiciel arrête la BasicStation en cours d'exécution pour stocker la mise à jour`Package`,`Version`, et`Model`afin que BasicStation doive être redémarré.

```
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided update.bin
2021-03-12 09:56:13.108 [CUP:INFO] CUPS provided signature len=70 keycrc=37316C36
2021-03-12 09:56:13.148 [CUP:INFO] ECDSA key#0 -> VERIFIED
2021-03-12 09:56:13.148 [CUP:INFO] Running update.bin as background process
2021-03-12 09:56:13.149 [SYS:VERB] /tmp/update.bin: Forked, waiting...
2021-03-12 09:56:13.151 [SYS:INFO] Process /tmp/update.bin (pid=6873) completed
2021-03-12 09:56:13.152 [CUP:INFO] Interaction with CUPS done - next regular check in 10s
```

Si la mise à jour du microprogramme échoue, vous voyez l'état`FIRST_RETRY`à partir du serveur CUPS, et la passerelle envoie la même requête. Si le serveur CUPS n'est pas en mesure de se connecter à la passerelle après une`SECOND_RETRY`, il affichera un état de`FAILED`.

Après la tâche précédente`COMPLETED`ou`FAILED`, supprimez l'ancienne tâche à l'aide de la commandedelete-sans-fil-gateway-taskavant d'en démarrer une nouvelle.

```
aws iotwireless delete-wireless-gateway-task \
    --id 1352172b-0602-4b40-896f-54da9ed16b57
```

Gestion des appareils avec AWS IoT Core pour LoRaWAN

Les appareils LoRaWan communiquent avec AWS IoT Core pour LoRaWan via les passerelles LoRaWan. Ajout d'appareils à AWS IoT Core pour LoRaWANAWS IoTtraiter les messages reçus des appareils pour les utiliser parAWS IoTet autres services.

Voici quelques considérations importantes lors de l'utilisation de vos appareils avec AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la façon d'ajouter votre appareil à AWS IoT Core pour LoRaWAN, consultezÀ bord de vos appareils AWS IoT Core pour LoRaWAN (p. 1074).

Considérations relatives aux

Lorsque vous sélectionnez un appareil que vous souhaitez utiliser pour communiquer avec AWS IoT Core Pour LoraWan, veuillez prendre en compte les points suivants.

- Détecteurs disponibles
- Capacité de la batterie
- Consommation d'énergie
- Coût
- Type d'antenne et portée de transmission

Utilisation d'appareils dotés de passerelles qualifiées pour AWS IoT Core pour LoRaWAN

Les périphériques que vous utilisez peuvent être couplés avec des passerelles sans fil qualifiées pour une utilisation avec AWS IoT Core pour LoRaWAN. Vous pouvez trouver ces passerelles et ces kits de développement dans laAWSCatalogue des appareils partenaires. Nous vous recommandons également de considérer la proximité de ces appareils à vos passerelles. Pour plus d'informations, consultez Utilisation de passerelles qualifiées à partir de laAWSCatalogue des appareils partenaires (p. 1085).

Version LoRaWAN

AWS IoT Core pour LoRaWan prend en charge tous les appareils conformes aux spécifications 1.0.x ou 1.1 LoRaWan standardisées par LoRa Alliance.

Modes d'activation

Avant que votre appareil LoRaWan puisse envoyer des données de liaison montante, vous devez effectuer un processus appeléactivationoujoinProcédure Pour activer votre appareil, vous pouvez soit utiliser OTAA (Over the Air Activation), soit ABP (Activation par personnalisation). Nous vous recommandons d'utiliser OTAA pour activer votre appareil car de nouvelles clés de session sont générées pour chaque activation, ce qui le rend plus sécurisé.

La spécification de votre périphérique sans fil est basée sur la version et le mode d'activation de LoRaWan, qui déterminent les clés racine et les clés de session générées pour chaque activation. Pour plus d'informations, consultez Ajoutez votre spécification de périphérique sans fil à AWS IoT Core pour LoRaWAN à l'aide de la console (p. 1075).

Classes de périphériques

Les appareils LoRaWan peuvent envoyer des messages de liaison montante à tout moment. L'écoute des messages de liaison descendante consomme la capacité de la batterie et réduit la durée de la batterie. Le protocole LoRaWan spécifie trois classes de périphériques LoRaWan.

- Les appareils de classe A sont en veille la plupart du temps et n'écoutent les messages de liaison descendante que pendant une courte période. Ces appareils sont principalement des capteurs alimentés par batterie avec une durée de vie de la batterie allant jusqu'à 10 ans.

- Les périphériques de classe B peuvent recevoir des messages dans des emplacements de liaison descendante planifiés. Ces dispositifs sont pour la plupart des actionneurs alimentés par batterie.
- Les appareils de classe C ne dorment jamais et n'écoutent jamais les messages entrants et il n'y a donc pas beaucoup de retard dans la réception des messages. Ces dispositifs sont pour la plupart des actionneurs alimentés par le secteur.

Pour plus d'informations sur ces considérations relatives aux périphériques sans fil, reportez-vous aux ressources mentionnées dansRessources d'apprentissage pour AWS IoT Core pour LoRaWAN (p. 1063).

Afficher le format des messages de liaison montante envoyés à partir d'appareils LoRaWan

Une fois que vous avez connecté votre appareil LoRaWan à AWS IoT Core pour LoRaWan, vous pouvez observer le format du message de liaison montante que vous recevrez de votre appareil sans fil.

Avant de pouvoir observer les messages de liaison montante

Vous devez avoir intégré votre appareil sans fil et connecté votre appareil àAWS IoTafin qu'il puisse transmettre et recevoir des données. Pour plus d'informations sur l'intégration de votre appareil à AWS IoT Core pour LoRaWAN, consultezÀ bord de vos appareils AWS IoT Core pour LoRaWAN (p. 1074).

Que contiennent les messages de liaison montante ?

Les appareils LoRaWAN se connectent à AWS IoT Core pour LoRaWan à l'aide de passerelles LoRaWan. Le message de liaison montante que vous recevez de l'appareil contiendra les informations suivantes.

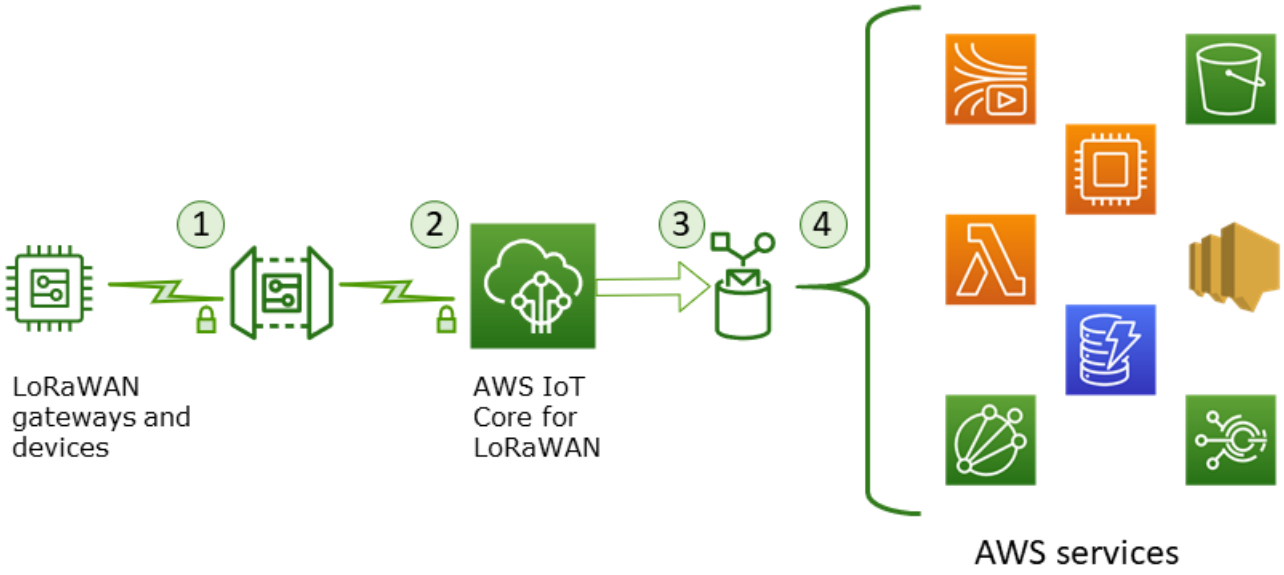
- Données de charge utile qui correspondent au message de charge utile crypté envoyé à partir du périphérique sans fil.
- Métadonnées sans fil comprenant :
  - Informations sur le périphérique telles que DevEUI, le débit de données et le canal de fréquence dans lequel le périphérique fonctionne.
  - Paramètres supplémentaires facultatifs et informations de passerelle pour les passerelles connectées au périphérique. Les paramètres de passerelle incluent l'IUE de la passerelle, le SNR et le RSSi.

  En utilisant les métadonnées sans fil, vous pouvez obtenir des informations utiles sur le périphérique sans fil et les données transmises entre votre appareil etAWS IoT. Par exemple, vous pouvez utiliser l'`AckedMessageId`pour vérifier si le dernier message de liaison descendante confirmé a été reçu par le périphérique. Le cas échéant, si vous choisissez d'inclure les informations de passerelle, vous pouvez déterminer si vous souhaitez passer à un canal de passerelle plus solide, plus proche de votre appareil.

Comment observer les messages de liaison montante ?

Une fois que vous avez intégré votre appareil, vous pouvez utiliser leClient de test MQTTsur leTestPage de laAWS IoTPour vous abonner à la rubrique que vous avez spécifiée lors de la création de votre destination. Vous commencerez à voir les messages après la connexion de votre appareil et commencerez à envoyer des données de charge utile.

Ce diagramme identifie les éléments clés d'un système LoRaWan connecté à AWS IoT Core pour LoRaWan, qui montre le plan de données principal et la façon dont les données circulent à travers le système.



Lorsque le périphérique sans fil commence à envoyer des données de liaison montante, AWS IoT Core pour LoRaWan encapsule les informations de métadonnées sans fil avec la charge utile, puis les envoie à votreAWSapplications.

Exemple de message de liaison montante

L'exemple suivant illustre le format du message de liaison montante reçu de votre appareil.

```
{
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",
    "PayloadData": "Cc48AAAAAAAAAA=",
    "WirelessMetadata":
    {
        "LoRaWAN":
        {
            "ADR": false,
            "Bandwidth": 125,
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "0",
            "DevAddr": "00b96cd4",
            "DevEui": "58a0cb000202c99",
            "FOptLen": 2,
            "FCnt": 1,
            "Fport": 136,
            "Frequency": "868100000",
            "Gateways": [
             {
                    "GatewayEui": "80029cfffe5cf1cc",
                    "Snr": -29,
                    "Rssi": 9.75
             }
             ],
            "MIC": "7255cb07",
            "MType": "UnconfirmedDataUp",
            "Major": "LoRaWANR1",
            "Modulation": "LORA",
            "PolarizationInversion": false,
            "SpreadingFactor": 12,
```

```
            "Timestamp": "2021-05-03T03:24:29Z"

        }
    }
}
```

Pour plus d'informations sur les différents paramètres dans les métadonnées de liaison montante, consultez.SendDataToWirelessDevice. Remarque :`SendDataToWirelessDevice`est utilisée pour les données envoyées à partir de AWS IoT Core pour LoRaWan sur votre appareil sans fil. Vous pouvez utiliser les paramètres de l'API comme référence pour en savoir plus sur les différents champs de votre message de liaison montante.

Exclure les métadonnées de passerelle des métadonnées de liaison montante

Si vous souhaitez exclure les informations de métadonnées de la passerelle de vos métadonnées de liaison montante, désactivez l'optionAddGwMetadataLorsque vous créez le profil de service. Pour de plus amples informations sur la désactivation de ce paramètre, veuillez consulter.Ajouter des profils de service (p. 1078).

Dans ce cas, vous ne pouvez pas voir l'`Gateways`dans les métadonnées de liaison montante, comme illustré dans l'exemple suivant.

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "ffffff10000163b0",
            "FCnt": 1,
            "FPort": 5,
            "Timestamp": "2021-04-29T05:19:43.646Z"
    }
  }
}
```

Surveillance et journalisation pour AWS IoT Core pour LoRaWAN

Vous pouvez surveiller AWS IoT Core pour les ressources et applications LoRaWan qui s'exécutent en temps réel à l'aide d'Amazon CloudWatch. Utilisez CloudWatch pour collecter et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et applications. Pour de plus amples informations sur les avantages de l'utilisation de la surveillance, veuillez consulter.Surveillance de AWS IoT (p. 352).

Suivez la surveillance de AWS IoT Core Ressources LoRaWAN

Pour enregistrer et surveiller vos ressources sans fil, procédez comme suit.

1. Créez un rôle de journalisation pour enregistrer vos AWS IoT Core pour les ressources LoRaWan, comme décrit dansCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103).
2. Les messages de journal de la console CloudWatch Logs ont un niveau de journal par défaut de`ERROR`, qui est moins verbeux et contient uniquement des informations sur les erreurs. Si vous souhaitez afficher

plus de messages détaillés, nous vous recommandons d'utiliser l'interface de ligne de commande pour configurer d'abord la journalisation, comme décrit dansConfigurer la journalisation pour AWS IoT Core Ressources LoRaWAN (p. 1106).

3. Ensuite, vous pouvez surveiller vos ressources en affichant les entrées de journal dans la console CloudWatch Logs. Pour plus d'informations, consultez Afficher CloudWatch AWS IoT Core pour les entrées LoRaWAN (p. 1115).
4. Vous pouvez créer des expressions de filtre en utilisantGroupes de journauxmais nous vous recommandons de créer d'abord des filtres simples et d'afficher les entrées de journal dans les groupes de journaux, puis d'aller dans CloudWatch Insights pour créer des requêtes pour filtrer les entrées de journal en fonction de la ressource ou de l'événement que vous surveillez. Pour plus d'informations, consultez Utilisez CloudWatch Insights pour filtrer les journaux AWS IoT Core pour LoRaWAN (p. 1121).

Les rubriques suivantes montrent comment configurer la journalisation pour AWS IoT Core pour LoRaWan et pour collecter des métriques à partir de CloudWatch. Outre les appareils LoRaWan, vous pouvez utiliser ces rubriques pour configurer la journalisation de tous les appareils Sidewalk que vous avez ajoutés à votre compte et les surveiller. Pour plus d'informations sur la façon d'ajouter ces périphériques, consultez la rubriqueIntégration d'Amazon Sidewalk pour AWS IoT Core (p. 1125).

Rubriques

- Configurer la journalisation pour AWS IoT Core pour LoRaWAN (p. 1103)
- Contrôle AWS IoT Core pour LoRaWan à l'aide des CloudWatch Logs (p. 1114)

Configurer la journalisation pour AWS IoT Core pour LoRaWAN

Avant de pouvoir surveiller et enregistrerAWS IoT, activez d'abord la journalisation pour AWS IoT Core pour les ressources LoRaWan à l'aide de l'interface de ligne de commande ou de l'API.

Lorsque vous envisagez de configurer votre AWS IoT Core pour la journalisation LoRaWan, la configuration de journalisation par défaut détermine commentAWS IoTL'activité sera enregistrée à moins que vous ne spécifiez autrement. À partir de là, vous pouvez obtenir des journaux détaillés avec un niveau de journal par défaut de`INFO`.

Après avoir examiné les journaux initiaux, vous pouvez modifier le niveau de journal par défaut en`ERROR`, qui est moins détaillé, et définissez un niveau de journal plus détaillé, spécifique à la ressource sur les ressources qui pourraient nécessiter plus d'attention. Les niveaux de journal peuvent être modifiés quand vous le souhaitez.

Les rubriques suivantes montrent comment configurer la journalisation pour AWS IoT Core pour les ressources LoRaWAN.

Rubriques

- Créer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103)
- Configurer la journalisation pour AWS IoT Core Ressources LoRaWAN (p. 1106)

Créer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN

La commande suivante montre comment créer un rôle de journalisation pour uniquement AWS IoT Core pour les ressources LoRaWAN. Si vous souhaitez également créer un rôle de journalisation pour AWS IoT Core , voirCréation d'un rôle de journalisation (p. 353).

Création d'un rôle de journalisation pour AWS IoT Core pour LoRaWAN

Avant de pouvoir activer la journalisation, vous devez créer un rôle IAM et une stratégie qui donne àAWSAutorisation de surveiller AWS IoT Core Pour l'activité LoRaWan en votre nom.

Création d'un rôle IAM pour journalisation

Pour créer un rôle de journalisation pour AWS IoT Core pour LoRaWAN, ouvrezHub de rôles de la console IAMet choisissezCréation d'un rôle.

1. UnderZone Select type of trusted entity, choisissezAutreAWScompte.
2. DansID de compte, entrez votreAWSID de compte, puis choisissezSuivant: Permissions (Autorisations).
3. Dans la zone de recherche, saisissez **`AWSIoTWirelessLogging`**.
4. Cochez la case en regard de la stratégie nomméeAWSIOTWirelessLogging, puis.Suivant: Tags (Balises).
5. Choisissez Next (Suivant) Review (Examiner).
6. DansNom de rôle, saisissez**`IoTWirelessLogsRole`**, puis.Création d'un rôle.

Modifier la relation d'approbation du rôle IAM

Dans le message de confirmation qui s'affiche après l'étape précédente, choisissez le nom du rôle que vous avez créé,IOTWIrelessLogsRole. Ensuite, vous allez modifier le rôle pour ajouter la relation d'approbation suivante.

1. DansRécapitulatifsection du rôleIOTWIrelessLogsRole, choisissez leRelations d'approbation, puisModification de la relation d'approbation.
2. DansDocument de stratégie, modifiez`Principal`se présente ainsi.

```
"Principal": {
    "Service": "iotwireless.amazonaws.com"
},
```

   Une fois que vous avez modifié le`Principal`, le document de stratégie complet doit se présenter comme cet exemple.

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

3. Pour enregistrer vos modifications et quitter, choisissezStratégie d'approbation.

Stratégie de journalisation pour AWS IoT Core pour LoRaWAN

Le document de stratégie suivant fournit la stratégie de rôle et la stratégie d'approbation qui permettent à AWS IoT Core pour que LoRaWan soumette des entrées de journal à CloudWatch en votre nom.

Note

CetteAWSLe document de stratégie géré a été créé automatiquement pour vous lorsque vous avez créé le rôle de journalisation,IOTWIrelessLogsRole.

Stratégie de rôle

Ce qui suit illustre le document de stratégie de rôle.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/iotwireless*"
        }
    ]
}
```

Stratégie d'approbation pour journaliser uniquement AWS IoT Core pour LoRaWAN

Ce qui suit montre la stratégie d'approbation pour la journalisation uniquement AWS IoT Core pour l'activité LoRaWAN.

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

Si vous avez créé le rôle IAM pour également journaliser AWS IoT Core , les documents de stratégie vous permettent d'enregistrer les deux activités. Pour plus d'informations sur la création d'un rôle de journalisation pour AWS IoT Core , voirCréation d'un rôle de journalisation (p. 353).

Étapes suivantes

Vous avez appris à créer un rôle de journalisation pour journaliser votre AWS IoT Core pour les ressources LoRaWAN. Par défaut, les journaux ont un niveau de journalisation`ERROR`, donc si vous voulez voir uniquement les informations d'erreur, accédez àAfficher CloudWatch AWS IoT Core pour les entrées LoRaWAN (p. 1115)pour surveiller vos ressources sans fil en affichant les entrées de journal.

Si vous souhaitez plus d'informations dans les entrées de journal, vous pouvez configurer le niveau de journal par défaut pour vos ressources ou pour différents types d'événements, par exemple en définissant

le niveau de journal surINFO. Pour plus d'informations sur la configuration de la journalisation pour vos ressources, consultezConfigurer la journalisation pour AWS IoT Core Ressources LoRaWAN (p. 1106).

Configurer la journalisation pour AWS IoT Core Ressources LoRaWAN

Pour configurer la journalisation pour AWS IoT Core Pour les ressources LoRawan, vous pouvez utiliser l'API ou l'interface de ligne de commande. Lorsque vous commencez à surveiller AWS IoT Core Pour les ressources LoRawan, vous pouvez utiliser la configuration par défaut. Pour ce faire, vous pouvez ignorer cette rubrique et passer àContrôle AWS IoT Core pour LoRaWan à l'aide des CloudWatch Logs (p. 1114)pour surveiller vos journaux.

Après avoir commencé à surveiller les journaux, vous pouvez utiliser l'interface de ligne de commande pour modifier les niveaux de journal en une option plus détaillée, par exemple en fournissantINFOandERRORet l'activation de la journalisation pour plus de ressources.

AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation

Avant d'utiliser l'API ou l'interface de ligne de commande, utilisez le tableau suivant pour en savoir plus sur les différents niveaux de journal et les ressources pour lesquelles vous pouvez configurer la journalisation. Le tableau affiche les paramètres que vous voyez dans les journaux CloudWatch lorsque vous surveillez les ressources. La façon dont vous configurez la journalisation de vos ressources déterminera les journaux que vous voyez dans la console.

Pour plus d'informations sur l'aspect d'un exemple de journaux CloudWatch et sur la façon dont vous pouvez utiliser ces paramètres pour consigner des informations utiles sur le AWS IoT Core Pour les ressources LoRaWAN, consultezAfficher CloudWatch AWS IoT Core pour les entrées LoRaWAN (p. 1115).

Niveaux des journaux et ressources

Nom	Valeurs possibles	Description
`logLevel`	`INFO`, `ERROR` ou `DISABLED`	• `ERROR` : affiche toute erreur qui entraîne l'échec d'une opération. Les journaux incluent uniquementERRORInformations. • `INFO` : Fournit des informations de haut niveau sur le flux des objets. Les journaux comprennentINFOandERRORInformations. • `DISABLED` : Désactive toute la journalisation.
`resource`	`WirelessGateway` ou `WirelessDevice`	Le type de la ressource, qui peut êtreWirelessGatewayouWirelessDevice.
`wirelessGatewayType`	`LoRaWAN`	Le type de la passerelle sans fil, lorsqueresourceestWirelessGateway, qui est toujours LoRaWAN.
`wirelessDeviceType`	`LoRaWAN` ou `Sidewalk`	Le type de l'appareil sans fil, lorsqueresourceestWirelessDevice, qui peut êtreLoRaWANouSidewalk.
`wirelessGatewayId`	-	Identifiant de la passerelle sans fil, lorsqueresourceestWirelessGateway.
`wirelessDeviceId`	-	Identifiant du périphérique sans fil, lorsqueresourceestWirelessDevice.

Nom	Valeurs possibles	Description
`event`	`Join`,`Rejoin`,`Registration`,`Uplink_Data`,`Downlink_Data`,`Uplink_Request`, etCertificate	Type d'événement pour les journaux de journalisation qui dépend si la ressource que vous enregistrez est un périphérique sans fil ou une passerelle sans fil. Pour plus d'informations, consultez Afficher CloudWatch AWS IoT Core pour les entrées LoRaWAN (p. 1115).

AWS IoTAPI de journalisation sans fil

Vous pouvez utiliser les actions d'API suivantes pour configurer la journalisation des ressources. Le tableau présente également un exemple de stratégie IAM que vous devez créer pour utiliser les actions API. La section suivante explique comment utiliser les API pour configurer les niveaux de journaux de vos ressources.

Journalisation des actions d'API

Nom d'API	Description	Exemple de stratégie IAM
GetLogLevelsByResourceType	Renvoie les niveaux de journal par défaut actuels, ou les niveaux de journal par type de ressource, qui peuvent inclure des options de journal pour les périphériques sans fil ou les passerelles sans fil.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:GetLogLevelsByResourceTypes" ], "Resource": [ "*" ] } ] }`
GetResourceLogLevel	Renvoie le remplacement au niveau du journal pour un identificateur de ressource et un type de ressource donnés. La ressource peut être un périphérique sans fil ou une passerelle sans fil.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:GetResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5`

Nom d'API	Description	Exemple de stratégie IAM
		ab12-cd3a- d00e-1f0e20c1204a",] }] }

PutResourceLogLevel	Définit le remplacement au niveau du journal pour un identificateur de ressource et un type de ressource donnés. La ressource peut être une passerelle sans fil ou un périphérique sans fil.
	
	Note
	
	Cette API a une limite de 200 remplacements de niveau journal par compte.
		{
		"Version": "2012-10-17",
		
		"Statement": [
		{
		"Effect":
		"Allow",
		"Action": [
		
		"iotwireless:PutResourceLogLevel"
		
		],
		"Resource": [
		
		"arn:aws:iotwireless:us-
		east-1:123456789012:WirelessDevice/012bc5
		ab12-cd3a-
		d00e-1f0e20c1204a",
		
		]
		}
		]
		}

Nom d'API	Description	Exemple de stratégie IAM
ResetAllResourceLogLevels	Supprime les remplacements au niveau du journal pour toutes les ressources, qui comprennent à la fois les passerelles sans fil et les périphériques sans fil. Note Cette API n'affecte pas les niveaux de journalisation définis à l'aide de la méthode`UpdateLogLevelsByResourceTypes`API.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:ResetAllResourceLogLevels" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/*", "arn:aws:iotwireless:us-east-1:123456789012:WirelessGateway/* ] } ] }`
ResetResourceLogLevel	Supprime le remplacement au niveau du journal pour un identificateur de ressource et un type de ressource donnés. La ressource peut être une passerelle sans fil ou un périphérique sans fil.	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iotwireless:ResetResourceLogLevel" ], "Resource": [ "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ] } ] }`

Nom d'API	Description	Exemple de stratégie IAM

UpdateLogLevelsByResourceTypes	Définissez le niveau de journal par défaut ou les niveaux de journal par type de ressource. Vous pouvez utiliser cette API pour les options de journal des périphériques sans fil ou des passerelles sans fil, et contrôler les messages de journal qui seront affichés dans CloudWatch.
	Note
	Les événements sont facultatifs et le type d'événement est lié au type de ressource. Pour plus d'informations, consultez Événements et types de ressources (p. 1115).
		<pre>{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Action": [
		"iotwireless:UpdateLogLevelsByResourceTy
		],
		"Resource": [
		"*"
		]
		}
		]
		}</pre>

Configurer les niveaux de journalisation des ressources à l'aide de l'interface

Cette section décrit comment configurer les niveaux de journalisation pour AWS IoT Core pour les ressources LoRaWan à l'aide de l'API ouAWS CLI.

Avant d'utiliser l'interface de ligne de commande :

- Assurez-vous que vous avez créé la stratégie IAM pour l'API pour laquelle vous souhaitez exécuter la commande CLI, comme décrit précédemment.
- Vous avez besoin du nom de ressource Amazon (ARN) du rôle que vous souhaitez utiliser. Si vous devez créer un rôle à utiliser pour la journalisation, consultezCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103).

Pourquoi utiliserAWS CLI

Par défaut, si vous créez le rôle IAM,`IoTWirelessLogsRole`, comme décrit dansCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103), vous verrez les journaux CloudWatch dans laAWS Management Consolequi ont un niveau de journal par défaut`ERROR`. Pour modifier le niveau de journal par défaut pour toutes vos ressources ou pour des ressources spécifiques, utilisez l'outilAWS IoTAPI de journalisation sans fil ou CLI.

Comment utiliser la stratégieAWS CLI

Les actions API peuvent être classées dans les types suivants, selon que vous souhaitez configurer les niveaux de journal pour toutes les ressources ou pour des ressources spécifiques :

- Actions d'API`GetLogLevelsByResourceTypes`and`UpdateLogLevelsByResourceTypes`peut récupérer et mettre à jour les niveaux de journalisation de toutes les ressources de votre compte qui sont d'un type spécifique, comme une passerelle sans fil, un appareil LoRaWan ou Sidewalk.
- Actions d'API`GetResourceLogLevel`,`PutResourceLogLevel`, et`ResetResourceLogLevel`peut récupérer, mettre à jour et réinitialiser les niveaux de journal des ressources individuelles que vous spécifiez à l'aide d'un identificateur de ressource.

- Action d'API`ResetAllResourceLogLevels`réinitialise le remplacement au niveau du journal sur`null`pour toutes les ressources pour lesquelles vous avez spécifié un remplacement au niveau du journal à l'aide de la commande`PutResourceLogLevel`API.

Pour utiliser l'interface de ligne de commande pour configurer la journalisation spécifique aux ressources pour AWS IoT

> **Note**
>
> Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

1. Par défaut, toutes les ressources ont un niveau de journal défini sur`ERROR`. Pour définir les niveaux de journalisation par défaut ou les niveaux de journalisation par type de ressource pour toutes les ressources de votre compte, utilisez l'outilupdate-log-levels-by-resource-typesCommande l'. L'exemple suivant montre comment créer un fichier JSON,`Input.json`, et fournissez-la en tant qu'entrée à la commande CLI. Vous pouvez utiliser cette commande pour désactiver sélectivement la journalisation ou remplacer le niveau de journal par défaut pour des types spécifiques de ressources et d'événements.

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions":
     [
        {
         "Type": "Sidewalk",
         "LogLevel": "INFO",
         "Events":
          [
            {
              "Event": "Registration",
              "LogLevel": "DISABLED"
            }
          ]
        },
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "Join",
             "LogLevel": "DISABLED"
            },
            {
             "Event": "Rejoin",
             "LogLevel": "ERROR"
            }
          ]
        }
      ]
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
```

```
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

où :

WirelessDeviceLogOptions

Liste des options de journal pour un périphérique sans fil. Chaque option de journal comprend le type de périphérique sans fil (Strottoir ou LoRaWan) et une liste des options de journal des événements de périphérique sans fil. Chaque option de journal des événements de périphérique sans fil peut éventuellement inclure le type d'événement et son niveau de journal.

WirelessGatewayLogOptions

Liste des options de journal pour une passerelle sans fil. Chaque option de journal comprend le type de passerelle sans fil (LoRaWan) et une liste des options de journal des événements de passerelle sans fil. Chaque option de journal des événements de passerelle sans fil peut éventuellement inclure le type d'événement et son niveau de journal.

DefaultLogLevel

Le niveau de journalisation à utiliser pour toutes vos ressources. Les valeurs valides sont `ERROR`, `INFO` et `DISABLED`. La valeur par défaut est `INFO`.

LogLevel

Niveau de journal à utiliser pour les types de ressources et les événements individuels. Ces niveaux de journal remplacent le niveau de journal par défaut, tel que le niveau de journal`INFO`pour la passerelle LoRaWan et les niveaux de journalisation`DISABLED`and`ERROR`pour les deux types d'événements.

Exécutez la commande suivante pour fournir l'`Input.json`Comme entrée dans la commande. Cette commande ne génère pas de sortie.

```
aws iotwireless update-log-levels-by-resource-types \
    --cli-input-json Input.json
```

Si vous souhaitez supprimer les options de journal des périphériques sans fil et des passerelles sans fil, exécutez la commande suivante.

```
{
    "DefaultLogLevel":"DISABLED",
    "WirelessDeviceLogOptions": [],
    "WireslessGatewayLogOptions":[]
}
```

2. La commande update-log-levels-by-resource-types ne renvoie aucune sortie. Utilisation de l'get-log-levels-by-resource-typespour récupérer les informations de journalisation spécifiques aux ressources. La commande renvoie le niveau de journal par défaut, ainsi que les options de journal du périphérique sans fil et de la passerelle sans fil.

   Note

   La .get-log-levels-by-resource-typesne peut pas récupérer directement les niveaux de journal dans la console CloudWatch. Vous pouvez utiliser la stratégieget-log-levels-by-resource-

typespour obtenir les dernières informations de niveau journal que vous avez spécifiées pour vos ressources à l'aide de la commandeupdate-log-levels-by-resource-typesCommande de l'

```
aws iotwireless get-log-levels-by-resource-types
```

Lorsque vous exécutez la commande suivante, elle renvoie les informations de journalisation les plus récentes que vous avez spécifiées avecupdate-log-levels-by-resource-types. Par exemple, si vous supprimez les options du journal des périphériques sans fil, puis exécutez la commandeget-log-levels-by-resource-typesretournera cette valeur en tant que`null`.

```
{
    "DefaultLogLevel": "INFO",
    "WirelessDeviceLogOptions": null,
     "WirelessGatewayLogOptions":
      [
        {
         "Type": "LoRaWAN",
         "LogLevel": "INFO",
         "Events":
          [
            {
             "Event": "CUPS_Request",
             "LogLevel": "DISABLED"
            },
            {
              "Event": "Certificate",
              "LogLevel": "ERROR"
            }
          ]
        }
      ]
}
```

3. Pour contrôler les niveaux de journal des passerelles sans fil individuelles ou des ressources de périphériques sans fil, utilisez les commandes CLI suivantes :

   - put-resource-log-level
   - get-resource-log-level
   - reset-resource-log-level

   Pour obtenir un exemple pour savoir quand utiliser ces CLI, imaginez que votre compte comporte un grand nombre de périphériques ou de passerelles sans fil qui sont en cours de journalisation. Si vous souhaitez résoudre les erreurs pour certains de vos périphériques sans fil uniquement, vous pouvez désactiver la journalisation de tous les périphériques sans fil en définissant le paramètre`DefaultLogLevel`sur`DISABLED`, et utilisezput-resource-log-levelpour définir`LogLevel`sur`ERROR`Seuls les appareils de votre compte.

   ```
   aws iotwireless put-resource-log-level \
       --resource-identifier
       --resource-type WirelessDevice
       --log-level ERROR
   ```

   Dans cet exemple, la commande définit le niveau de journalisation sur`ERROR`uniquement pour la ressource de périphérique sans fil spécifiée et les journaux de toutes les autres ressources sont désactivés. Cette commande ne génère pas de sortie. Pour récupérer ces informations et vérifier que les niveaux de journalisation ont été définis, utilisez la commandeget-resource-log-levelCommande de l'

4. À l'étape précédente, après avoir débogué le problème et résolu l'erreur, vous pouvez exécuter la commandereset-resource-log-levelpour réinitialiser le niveau de journal de cette ressource à`null`. Si vous avez utilisé le`put-resource-log-level`pour définir le remplacement au niveau du journal pour plusieurs périphériques sans fil ou ressources de passerelle, par exemple pour résoudre les erreurs de dépannage de plusieurs périphériques, vous pouvez réinitialiser les remplacements au niveau du journal à`null`pour toutes ces ressources en utilisant lereset-all-resource-log-levelsCommande de l'

   ```
   aws iotwireless reset-all-resource-log-levels
   ```

   Cette commande ne génère pas de sortie. Pour récupérer les informations de journalisation des ressources, exécutez la commandeget-resource-log-levelCommande de l'

Étapes suivantes

Vous avez appris à créer le rôle de journalisation et à utiliser leAWS IoTAPI sans fil pour configurer la journalisation pour votre AWS IoT Core pour LoraWanResources. Ensuite, pour en savoir plus sur la surveillance de vos entrées de journal, accédez àContrôle AWS IoT Core pour LoRaWan à l'aide des CloudWatch Logs (p. 1114).

Contrôle AWS IoT Core pour LoRaWan à l'aide des CloudWatch Logs

AWS IoT Core pour LoraWandispose de plus de 50 entrées de journal CloudWatch activées par défaut. Chaque entrée de journal décrit le type d'événement, le niveau de journal et le type de ressource. Pour plus d'informations, consultez AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

Suivez la surveillance de AWS IoT Core Ressources LoRaWAN

Lorsque la journalisation est activée pour AWS IoT Core pour LoRaWAN AWS IoT Core Pour LoRaWan envoie des événements d'avancement concernant chaque message qui passe de vos appareils àAWS IoTet en arrière. Par défaut, AWS IoT Core pour les entrées de journal LoRaWan ont un niveau d'erreur de journal par défaut. Lorsque vous activez la journalisation comme décrit dansCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103), vous verrez apparaître dans la console CloudWatch les messages dont le niveau de journal par défaut est`ERROR`. En utilisant ce niveau de journal, les messages afficheront uniquement des informations d'erreur pour tous les périphériques sans fil et les ressources de passerelle que vous utilisez.

Si vous souhaitez que les journaux affichent des informations supplémentaires, telles que celles dont le niveau de journal est`INFO`ou désactivez les journaux de certains de vos appareils et n'affichez les messages de journal que pour certains de vos appareils, vous pouvez utiliser le AWS IoT Core pour l'API de journalisation LoRaWan. Pour plus d'informations, consultez Configurer les niveaux de journalisation des ressources à l'aide de l'interface (p. 1110).

Vous pouvez également créer des expressions de filtre pour afficher uniquement les messages requis.

Avant de pouvoir afficher AWS IoT Core Pour les journaux LoRaWan dans la console

Pour rendre/aws/iotwirelesss'affichent dans la console CloudWatch, vous devez avoir effectué les opérations suivantes.

- Connexion activée AWS IoT Core pour LoRaWAN. Pour plus d'informations sur la façon d'activer la connexion AWS IoT Core pour LoRaWAN, consultezConfigurer la journalisation pour AWS IoT Core pour LoRaWAN (p. 1103).
- Écrit quelques entrées de journal en exécutant AWS IoT Core pour les opérations LoRaWAN.

Pour créer et utiliser plus efficacement des expressions de filtre, nous vous recommandons d'essayer d'utiliser les informations CloudWatch comme décrit dans les rubriques suivantes. Nous vous recommandons également de suivre les sujets dans l'ordre où ils sont présentés ici. Cela vous aidera à utiliser CloudWatchGroupes de journauxpour en savoir plus sur les différents types de ressources, ses types d'événements et les niveaux de journal que vous pouvez utiliser pour afficher les entrées de journal dans la console. Vous pouvez ensuite apprendre à créer des expressions de filtre à l'aide de CloudWatch Insights pour obtenir des informations plus utiles à partir de vos ressources.

Rubriques

Afficher CloudWatch AWS IoT Core pour les entrées LoRaWAN

Après avoir configuré la journalisation pour AWS IoT Core pour LoRaWAN comme cela est décrit dansCréer un rôle et une stratégie de journalisation pour AWS IoT Core pour LoRaWAN (p. 1103)et écrit certaines entrées de journal, vous pouvez afficher les entrées de journal dans la console CloudWatch en procédant comme suit.

Affichage d'unAWS IoTjournaux dans la console des groupes de journaux CloudWatch

DansConsole CloudWatch, les journaux CloudWatch apparaissent dans un groupe de journaux nommé/aws/iotwireless. Pour de plus amples informations sur CloudWatch Logs, veuillez consulterCloudWatch Logs.

Pour afficher vos rapportsAWS IoTEntrées des journaux dans la console CloudWatch

Accédez àhttps://console.aws.amazon.com/cloudwatch/et choisissezGroupes de journauxDans le volet de navigation.

1. DansFiltrerZone de texte, entrez**`/aws/iotwireless`**, puis`/aws/iotwireless`Groupe de journaux.
2. Pour consulter la liste complète des AWS IoT Core pour les journaux LoRaWan générés pour votre compte, choisissezRechercher tous. Pour consulter un flux de journal individuel, cliquez sur l'icône de développement.
3. Pour filtrer les flux de journaux, vous pouvez également entrer une requête dans la listeFiltrage des événementsZone de texte. Voici quelques demandes à tenter :

   - `{ $.logLevel = "ERROR" }`

     Utilisez ce filtre pour trouver tous les journaux qui ont un niveau de journalisation`ERROR`et vous pouvez développer les flux d'erreurs individuels pour lire les messages d'erreur, ce qui vous aidera à les résoudre.
   - `{ $.resource = "WirelessGateway" }`

     Recherchez tous les journaux pour le`WirelessGateway`quel que soit le niveau du journal.
   - `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

     Trouvez tous les journaux qui ont un type d'événement`CUPS_Request`et un niveau de journal de`ERROR`.

Événements et types de ressources

Le tableau suivant présente les différents types d'événements pour lesquels vous verrez des entrées de journal. Les types d'événements varient également selon que le type de ressource est un périphérique

sans fil ou une passerelle sans fil. Vous pouvez utiliser le niveau de journal par défaut pour les ressources et les types d'événements ou remplacer le niveau de journal par défaut en spécifiant un niveau de journal pour chacun d'eux.

Types d'événements basés sur les ressources utilisées

Ressource	Type de ressource	Type d'événement	
passerelle sans fil	LoRaWAN	• CUPS_Requête • Certificat	
Appareil sans fil	LoRaWAN	• Rejoindre • Rejoindre • Uplink_Données • Link_Données	
Appareil sans fil	Trottoir	• Inscription • Uplink_Données • Link_Données	

La rubrique suivante contient plus d'informations sur ces types d'événements et les entrées de journal pour les passerelles sans fil et les périphériques sans fil.

Rubriques

• Entrées de journal des passerelles sans fil et des ressources de périphériques sans fil (p. 1116)

Entrées de journal des passerelles sans fil et des ressources de périphériques sans fil

Après avoir activé la journalisation, vous pouvez afficher les entrées de journal de vos passerelles sans fil et périphériques sans fil. La section suivante décrit les différents types d'entrées de journal en fonction de vos types de ressources et d'événements.

Entrées des journaux d'accès

Cette section présente quelques exemples d'entrées de journal pour vos ressources de passerelle sans fil que vous verrez dans lehttps://console.aws.amazon.com/cloudwatch/console Ces messages de journal peuvent avoir un type d'événement comme`CUPS_Request`ou`Certificate`, et peut être configuré pour afficher un niveau de journal`INFO`,`ERROR`, ou`DISABLED`au niveau de la ressource ou au niveau de l'événement. Si vous souhaitez afficher uniquement les informations d'erreur, définissez le niveau de journal sur`ERROR`. Message dans la liste`ERROR`contiendra des informations sur la raison pour laquelle elle a échoué.

Les entrées de journal de votre ressource de passerelle sans fil peuvent être classées en fonction des types d'événements suivants :

- CUPS_Requête

  Le LoRaLes bases™Station exécutée sur votre passerelle envoie périodiquement une demande de mise à jour au serveur de configuration et de mise à jour (CUPS). Pour ce type d'événement, si vous définissez le niveau de journal sur`INFO`lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, puis dans les journaux :

  - Si l'événement aboutit, vous verrez des messages de journal qui ont une`logLevel`de`INFO`. Les messages comprendront des détails sur la réponse CUPS envoyée à votre passerelle et les détails de la passerelle. Voici un exemple de cette entrée de journal. Pour en savoir plus sur`logLevel`et d'autres

champs de l'entrée de journal, consultez. AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "INFO",
    "message": "Sending CUPS response of total length 3213 to GatewayEui:
 fefffff00000000e2 with  TC Credentials,"
}
```

- S'il y a une erreur, vous verrez les entrées de journal qui ont un`logLevel`de`ERROR`Les messages contiennent des détails sur l'erreur. Exemples de situations où une erreur se produit dans le`CUPS_Request`incluent : CRC CUPS manquant, inadéquation dans le TC Uri de la passerelle avec AWS IoT Core pour LoRaWAN`IoTWirelessGatewayCertManagerRole`, ou n'est pas en mesure d'obtenir l'enregistrement de passerelle sans fil. L'exemple suivant montre une entrée de journal CRC manquante. Pour résoudre l'erreur, vérifiez la configuration de votre passerelle pour vérifier que vous avez entré le CRC CUPS correct.

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "gatewayEui": "fefffff00000000e2",
    "event": "CUPS_Request",
    "logLevel": "ERROR",
    "message": "The CUPS CRC is missing from the request. Check your gateway setup and
 enter the CUPS CRC,"
}
```

- Certificate

  Ces entrées de journal vous aideront à vérifier si votre passerelle sans fil a présenté le certificat approprié pour authentifier la connexion àAWS IoT. Pour ce type d'événement, si vous définissez le niveau de journal sur`INFO`lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, puis dans les journaux :

  - Si l'événement aboutit, vous verrez des messages de journal qui ont une`logLevel`de`INFO`. Les messages comprendront des détails sur l'ID de certificat et l'identificateur de passerelle sans fil. Voici un exemple de cette entrée de journal. Pour en savoir plus sur`logLevel`et d'autres champs de l'entrée de journal, consultez. AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "Gateway connection authenticated.
    (CertificateId: b5942a7aee973eda24314e416889227a5e0aa5ed87e6eb89239a83f515dea17c,
 WirelessGatewayId: 5da85cc8-3361-4c79-8be3-3360fb87abda)"
}
```

  - S'il y a une erreur, vous verrez les entrées de journal qui ont un`logLevel`de`ERROR`Les messages contiennent des détails sur l'erreur. Exemples de situations où une erreur se produit dans le`Certificate`incluent un ID de certificat, un identificateur de passerelle sans fil non valide ou une

inadéquation entre l'identificateur de passerelle sans fil et l'ID de certificat. L'exemple suivant montre un`ERROR`en raison d'un identifiant de passerelle sans fil non valide. Pour résoudre l'erreur, vérifiez les identifiants de passerelle.

```
{
    "resource": "WirelessGateway",
    "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "wirelessGatewayType": "LoRaWAN",
    "event": "Certificate",
    "logLevel": "INFO",
    "message": "The gateway connection couldn't be authenticated because a provisioned
 gateway associated with the certificate couldn't be found.
    (CertificateId: 729828e264810f6fc7134daf68056e8fd848afc32bfe8082beeb44116d709d9e)"
}
```

Entrées des journaux d'appareil

Cette section présente certaines des entrées de journal des ressources de votre périphérique sans fil que vous verrez dans la sectionhttps://console.aws.amazon.com/cloudwatch/. Le type d'événement de ces messages de journal dépend si vous utilisez un périphérique LoRaWan ou Sidewalk. Chaque ressource de périphérique sans fil ou type d'événement peut être configuré pour afficher un niveau de journal de`INFO`,`ERROR`, ou`DISABLED`.

> **Note**
>
> Votre demande ne doit pas contenir simultanément les métadonnées sans fil LoRaWan et Sidewalk. Pour éviter une`ERROR`pour ce scénario, spécifiez les données sans fil LoRaWan ou Sidewalk.

Entrées des journaux d'appareil LoRaWAN

Les entrées de journal de votre périphérique sans fil LoRaWan peuvent être classées en fonction des types d'événements suivants :

- **`Join`** and **`Rejoin`**

  Lorsque vous ajoutez un appareil LoRaWan et que vous le connectez à AWS IoT Core pour LoRaWan, avant que votre appareil puisse envoyer des données de liaison montante, vous devez effectuer un processus appelé`activation`ou`join procedure`. Pour plus d'informations, consultez Ajoutez votre appareil sans fil à AWS IoT Core pour LoRaWAN (p. 1075).

  Pour ce type d'événement, si vous définissez le niveau de journal sur`INFO`lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, puis dans les journaux :

  - Si l'événement aboutit, vous verrez des messages de journal qui ont une`logLevel`de`INFO`. Les messages comprendront des détails sur l'état de votre demande de jointure ou de réintégration. Voici un exemple de cette entrée de journal. Pour en savoir plus sur`logLevel`et d'autres champs de l'entrée de journal, consultez. AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

```
{
    "timestamp": "2021-05-13T16:56:08.853Z",
    "resource": "WirelessDevice",
    "wirelessDeviceType": "LoRaWAN",
    "WirelessDeviceId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
    "devEui": "fefffff00000000e2",
    "event": "Rejoin",
    "logLevel": "INFO",
    "message": "Rejoin succeeded"
}
```

- S'il y a une erreur, vous verrez les entrées de journal qui ont un`logLevel`de`ERROR`Les messages contiennent des détails sur l'erreur. Exemples de situations où une erreur se produit dans le`Join`and`Rejoin`incluent un paramètre de région LoRaWan non valide ou une vérification du code MIC (Message Integrity Code) non valide. L'exemple suivant montre une erreur de jointure due à la vérification MIC. Pour résoudre l'erreur, vérifiez si vous avez entré les clés racine correctes.

```
{
    "timestamp": "2020-11-24T01:46:50.883481989Z",
    "resource": "WirelessDevice",
    "wirelessDeviceType": "LoRaWAN",
    "WirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
    "devEui": "58a0cb000020255c",
    "event": "Join",
    "logLevel": "ERROR",
    "message": "invalid MIC. It's most likely caused by wrong root keys."
}
```

- Uplink_Data et Downlink_Data

  Type d'événement`Uplink_Data`est utilisé pour les messages générés par AWS IoT Core pour LoRaWan lorsque la charge utile est envoyée à partir du périphérique sans fil versAWS IoT. Type d'événement`Downlink_Data`est utilisé pour les messages qui sont liés à des messages de liaison descendante envoyés à partir deAWS IoTsur le périphérique sans fil.

  > **Note**
  >
  > Events (Événements)`Uplink_Data`and`Downlink_Data`s'appliquent à la fois aux appareils LoRaWan et Strowalk.

  Pour ce type d'événement, si vous définissez le niveau de journal sur`INFO`lors de la configuration de l'interface de ligne de commande pour vos périphériques sans fil, vous verrez dans les journaux :

  - Si l'événement aboutit, vous verrez des messages de journal qui ont une`logLevel`de`INFO`. Les messages comprendront des détails sur l'état du message de liaison montante ou descendante qui a été envoyé et l'identificateur de périphérique sans fil. Voici un exemple de cette entrée de journal pour un périphérique Sidewalk. Pour en savoir plus sur`logLevel`et d'autres champs de l'entrée de journal, consultez. AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "5371db88-d63d-481a-868a-e54b6431845d",
    "wirelessDeviceType": "Sidewalk",
    "event": "Downlink_Data",
    "logLevel": "INFO",
    "messageId": "8da04fa8-037d-4ae9-bf67-35c4bb33da71",
    "message": "Message delivery succeeded.  MessageId: 8da04fa8-037d-4ae9-
bf67-35c4bb33da71. AWS IoT Core: {\"message\":\"OK\",\"traceId\":\"038b5b05-a340-
d18a-150d-d5a578233b09\"}"
}
```

  - S'il y a une erreur, vous verrez les entrées de journal qui ont un`logLevel`de`ERROR`, et les messages incluront des détails sur l'erreur, ce qui vous aidera à la résoudre. Exemples de situations où une erreur se produit dans le`Registration`incluent : problèmes d'authentification, demandes non valides ou trop nombreuses, impossible de chiffrer ou de déchiffrer la charge utile ou impossible de trouver le périphérique sans fil à l'aide de l'ID spécifié. L'exemple suivant montre une erreur d'autorisation rencontrée lors du traitement d'un message.

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "cb4c087c-1be5-4990-8654-ccf543ee9fff",
    "wirelessDeviceType": "LoRaWAN",
```

```
    "event": "Uplink_Data",
    "logLevel": "ERROR",
    "message": "Cannot assume role MessageId: ef38877f-3454-4c99-96ed-5088c1cd8dee.
    Access denied: User: arn:aws:sts::005196538709:assumed-role/
DataRoutingServiceRole/6368b35fd48c445c9a14781b5d5890ed is not authorized to perform:
 sts:AssumeRole on resource: arn:aws:iam::400232685877:role/ExecuteRules_Role\tstatus
 code: 403, request id: 471c3e35-f8f3-4e94-b734-c862f63f4edb"
}
```

Entrées des journaux d'appareil

Les entrées de journal de votre appareil Sidewalk peuvent être classées en fonction des types d'événements suivants :

- **`Registration`**

  Ces entrées de journal vous aideront à surveiller l'état de tous les appareils Sidewalk que vous enregistrez auprès de AWS IoT Core pour LoRaWAN. Pour ce type d'événement, si vous définissez le niveau de journal sur`INFO`lors de la configuration de l'interface de ligne de commande pour votre ressource de périphérique sans fil, dans les journaux, vous verrez les messages de journal qui ont un`logLevel`de`INFO`and`ERROR`. Les messages comprendront des détails sur la progression de l'inscription du début à la fin.`ERROR`contiendra des informations sur la façon de résoudre les problèmes liés à l'enregistrement de votre appareil.

  Voici un exemple pour un message de journal avec le niveau de journal de`INFO`. Pour de plus amples informations sur l'`logLevel`et d'autres champs de l'entrée de journal, consultez. AWS IoT Core pour les ressources LoRaWan et les niveaux de journalisation (p. 1106).

```
{
    "resource": "WirelessDevice",
    "wirelessDeviceId": "8d0b2775-e19b-4b2a-a351-cb8a2734a504",
    "wirelessDeviceType": "Sidewalk",
    "event": "Registration",
    "logLevel": "INFO",
    "message": "Successfully completed device registration. Amazon SidewalkId =
 2000000002"
}
```

- Uplink_Data et Downlink_Data

  Types d'événements`Uplink_Data`and`Downlink_Data`pour les dispositifs de trottoir sont similaires aux types d'événements correspondants pour les appareils LoRaWan. Pour de plus amples informations, consultez la sectionUplink_Data et Downlink_Datadécrite précédemment pour les entrées du journal de périphérique LoRaWan.

Étapes suivantes

Vous avez appris à afficher les entrées de journal de vos ressources et les différentes entrées de journal que vous pouvez afficher dans la console CloudWatch après avoir activé la journalisation pour AWS IoT Core pour LoRaWAN. Alors que vous pouvez créer des flux de filtre en utilisantGroupes de journaux, nous vous recommandons d'utiliser CloudWatch Insights pour créer et utiliser des flux de filtres. Pour plus d'informations, consultez Utilisez CloudWatch Insights pour filtrer les journaux AWS IoT Core pour LoRaWAN (p. 1121).

Utilisez CloudWatch Insights pour filtrer les journaux AWS IoT Core pour LoRaWAN

Bien que vous puissiez utiliser CloudWatch Logs pour créer des expressions de filtre, nous vous recommandons d'utiliser les informations CloudWatch pour créer et utiliser plus efficacement des expressions de filtre en fonction de votre application.

Nous vous recommandons d'utiliser CloudWatch d'abordGroupes de journauxpour en savoir plus sur les différents types de ressources, ses types d'événements et les niveaux de journal que vous pouvez utiliser pour afficher les entrées de journal dans la console. Vous pouvez ensuite utiliser les exemples d'expressions de filtre sur cette page comme référence pour créer vos propres filtres pour votre AWS IoT Core pour les ressources LoRaWAN.

Affichage d'unAWS IoTjournaux dans la console d'informations CloudWatch Logs

DansConsole CloudWatch, les journaux CloudWatch apparaissent dans un groupe de journaux nommé/aws/iotwireless. Pour de plus amples informations sur CloudWatch Logs, veuillez consulterCloudWatch Logs.

Pour afficher vos rapportsAWS IoTJournaux dans la console CloudWatch

Accédez àhttps://console.aws.amazon.com/cloudwatch/et choisissezInformations sur les journauxDans le volet de navigation.

1. DansFiltrerZone de texte, entrez**`/aws/iotwireless`**, puis`/aws/iotwireless`Informations sur les journaux.
2. Pour afficher la liste complète des groupes de journaux, choisissezSélectionnez le ou les groupes de journaux. Pour consulter les groupes de journaux pour AWS IoT Core pour LoRaWAN, choisissez`/aws/iotwireless`.

Vous pouvez maintenant commencer à saisir des requêtes pour filtrer les groupes de journaux. Les sections suivantes contiennent des requêtes utiles qui vous aideront à obtenir des informations sur vos mesures de ressources.

Créez des requêtes utiles pour filtrer et obtenir des informations AWS IoT Core pour LoRaWAN

Vous pouvez utiliser des expressions de filtre pour afficher des informations de journal supplémentaires utiles avec CloudWatch Insights. Voici quelques exemples de requêtes :

Afficher uniquement les journaux pour des types de ressources spécifiques

Vous pouvez créer une requête qui vous aidera à afficher les journaux uniquement pour des types de ressources spécifiques, comme une passerelle LoRaWan ou un périphérique Sidewalk. Par exemple, pour filtrer les journaux de manière à afficher uniquement les messages pour les périphériques Sidewalk, vous pouvez entrer la requête suivante et choisirExécuter une requête. Pour enregistrer cette requête, choisissez Save (Enregistrer).

```
fields @message
| filter @message like /Sidewalk/
```

Une fois la requête exécutée, vous verrez les résultats dans leJournaux, qui affiche les horodatages des journaux liés aux périphériques Sidewalk de votre compte. Vous verrez également un graphique à barres, qui indique l'heure à laquelle les événements se sont produits, s'il y a eu de tels événements qui

se sont produits précédemment liés à votre périphérique Sidewalk. Ce qui suit montre un exemple si vous développez l'un des résultats dans leJournauxOnglet. Sinon, si vous souhaitez résoudre les erreurs liées aux périphériques Sidewalk, vous pouvez ajouter un autre filtre qui définit le niveau de journal sur`ERROR`et afficher uniquement les informations d'erreur.

```
Field           Value
@ingestionTime      1623894967640
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-715adccfb34170214ec2f6667ddfa13cb5af2c3ddfc52fbeee0e554a2e780bed
@message      {
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "Sidewalk",
                    "devEui": "fefffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Successfully sent downlink message. Amazon SidewalkId =
 2000000006, Sequence number = 0"
                    }
@timestamp          1623894967640
devEui            fefffff000000011a
event         Downlink_Data
logLevel            INFO
message           Successfully sent downlink message. Amazon SidewalkId = 2000000006,
 Sequence number = 0
messageId     7e752a10-28f5-45a5-923f-6fa7133fedda
resource      WirelessDevice
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  Sidewalk
```

Afficher des messages ou des événements spécifiques

Vous pouvez créer une requête qui vous aidera à afficher des messages spécifiques et à observer quand les événements se sont produits. Par exemple, si vous voulez voir quand votre message de liaison descendante a été envoyé à partir de votre périphérique sans fil LoRaWan, vous pouvez entrer la requête suivante et choisirExécuter une requête. Pour enregistrer cette requête, choisissez Save (Enregistrer).

```
filter @message like /Downlink message sent/
```

Une fois la requête exécutée, vous verrez les résultats dans leJournaux, qui affiche les horodatages lorsque le message de liaison descendante a été envoyé avec succès à votre périphérique sans fil. Vous verrez également un graphique à barres, qui indique l'heure à laquelle un message de liaison descendante a été envoyé, si des messages de liaison descendante ont déjà été envoyés à votre périphérique sans fil. Ce qui suit montre un exemple si vous développez l'un des résultats dans leJournauxOnglet. Sinon, si un message de liaison descendante n'a pas été envoyé, vous pouvez modifier la requête pour afficher uniquement les résultats lorsque le message n'a pas été envoyé afin que vous puissiez déboguer le problème.

```
Field           Value
@ingestionTime      1623884043676
@log              954314929104:/aws/iotwireless
@logStream    WirelessDevice-
Downlink_Data-42d0e6d09ba4d7015f4e9756fcdc616d401cd85fe3ac19854d9fbd866153c872
@message      {
                    "timestamp": "2021-06-16T22:54:00.770493863Z",
                    "resource": "WirelessDevice",
                    "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d",
                    "wirelessDeviceType": "LoRaWAN",
```

```
                    "devEui": "fefffff000000011a",
                    "event": "Downlink_Data",
                    "logLevel": "INFO",
                    "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda",
                    "message": "Downlink message sent. MessageId:
 7e752a10-28f5-45a5-923f-6fa7133fedda"
                    }
@timestamp          1623884040858
devEui              fefffff000000011a
event          Downlink_Data
logLevel            INFO
message           Downlink message sent. MessageId: 7e752a10-28f5-45a5-923f-6fa7133fedda
messageId      7e752a10-28f5-45a5-923f-6fa7133fedda
resource       WirelessDevice
timestamp      2021-06-16T22:54:00.770493863Z
wirelessDeviceId    3b058d05-4e84-4e1a-b026-4932bddf978d
wirelessDeviceType  LoRaWAN
```

Étapes suivantes

Vous avez appris à utiliser CloudWatch Insights pour obtenir des informations plus utiles en créant des requêtes pour filtrer les messages de journal. Vous pouvez combiner certains des filtres décrits précédemment et concevoir vos propres filtres en fonction de la ressource que vous surveillez. Pour de plus amples informations sur l'utilisation de CloudWatch Insights, veuillez consulterAnalyse des données de journal avec CloudWatch Insights.

Après avoir créé des requêtes avec CloudWatch Insights, si vous les avez enregistrées, vous pouvez charger et exécuter les requêtes enregistrées selon vos besoins. Sinon, si vous cliquez sur le boutonHistoriquedans CloudWatchInformations sur les journaux, vous pouvez afficher les requêtes précédemment exécutées et les réexécuter si nécessaire, ou encore les modifier en créant des requêtes supplémentaires.

Sécurité des données avec AWS IoT Core pour LoRaWAN

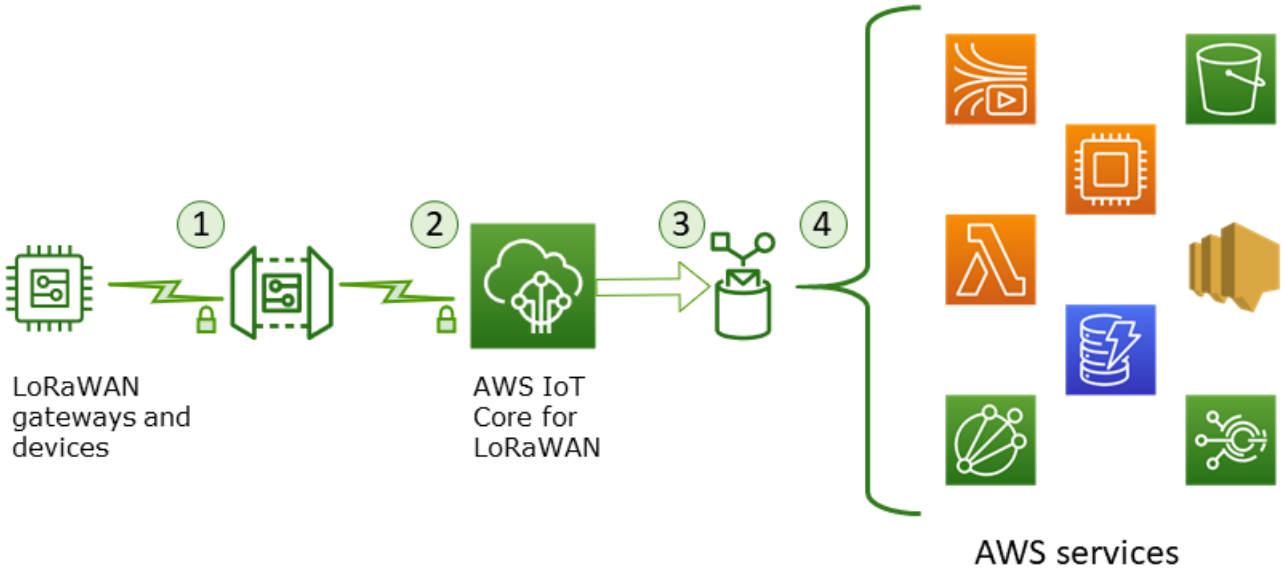
Deux méthodes sécurisent les données de votre AWS IoT Core pour les appareils LoRaWAN :

- Sécurité utilisée par les périphériques sans fil pour communiquer avec les passerelles.

  Les appareils LoRaWan suivent les pratiques de sécurité décrites dans la sectionSÉCURITÉ LoRaWan™ : Livre blanc préparé pour la LoRa Alliance™ par Gemalto, Actility et Semtech.
- La sécurité que AWS IoT Core utilise pour connecter des passerelles à AWS IoT Core pour LoRaWan et envoyer les données à d'autresAWSServices .

  AWS IoT Core la sécurité est décrite dansProtection des données dans AWS IoT Core (p. 312).

Ce diagramme identifie les éléments clés d'un système LoRaWan connecté à AWS IoT Core pour que LoRaWan identifie la manière dont les données sont sécurisées dans l'ensemble.



1. Le périphérique sans fil LoRaWan crypte ses messages binaires en utilisant le mode CTR AES128 avant de les transmettre.
2. Connexions de passerelle AWS IoT Core pour LoRaWan sont sécurisés par TLS comme décrit dansSécurité du transport dans AWS IoT (p. 313). AWS IoT Core pour LoraWan déchiffre le message binaire et code la charge utile du message binaire déchiffré sous la forme d'une chaîne base64.
3. Le message encodé en base64 résultant est envoyé en tant que charge utile du message à laAWS IoTdécrite dans la destination affectée au périphérique. Données à l'intérieurAWSest chiffré à l'aide deAWS- les clés.
4. La .AWS IoTdirige les données du message vers les services décrits dans la configuration de la règle. Données à l'intérieurAWSest chiffré à l'aide deAWS- les clés.

Sécurité du transport de l'appareil et de la passerelle LoRaWan

Appareils LoRaWAN et AWS IoT Core pour LoRaWan Store les clés racine pré-partagées. Les clés de session sont dérivées à la fois par les périphériques LoRaWan et AWS IoT Core pour LoRaWan suivant les protocoles. Les clés de session symétriques sont utilisées pour le chiffrement et le déchiffrement en mode CTR AES-128 standard. Un code d'intégrité de message (MIC) de 4 octets est également utilisé pour vérifier l'intégrité des données à l'aide d'un algorithme CMAC AES-128 standard. Les clés de session peuvent être mises à jour à l'aide du processus Join/Rejoindre.

Les pratiques de sécurité des passerelles LoRa sont décrites dans les spécifications LoRaWan. Les passerelles LoRa se connectent à AWS IoT Core pour LoRaWan via un socket web utilisant un `Basics Station`. AWS IoT Core pour LoRaWAN prend uniquement en charge`Basics Station`version 2.0.4 et ultérieures.

Avant que la connexion de socket Web soit établie, AWS IoT Core pour LoRaWAN utilise laMode d'authentification du serveur TLS et du client (p. 313)pour authentifier la passerelle. AWS IoT Core pour LoRaWan gère également un serveur de configuration et de mise à jour (CUPS) qui configure et met à jour les certificats et les clés utilisés pour l'authentification TLS.

Intégration d'Amazon Sidewalk pour AWS IoT Core

Amazon trottoirest un réseau partagé qui aide des appareils comme Amazon Echo, RingCams de sécurité, les lumières extérieures, les détecteurs de mouvement et les trackers Tile fonctionnent mieux à la maison et au-delà de la porte d'entrée. Lorsque cette option est activée, Amazon Sidewalk peut prendre en charge d'autres appareils Sidewalk dans votre communauté et ouvrir la porte à des innovations telles que la localisation d'articles connectés à Amazon Sidewalk.

Démarrage avec Amazon Sidewalk Integration pour AWS IoT Core

Avec l'intégration d'Amazon Sidewalk pour AWS IoT Core , vous pouvez ajouter votre parc d'appareils Sidewalk à laAWSCloud. Procédez comme suit.

1. Passez en revue le SDK de Sidewalk et la documentation

   En savoir plus sur Amazon Sidewalk et sur la façon dont vos appareils peuvent l'utiliser.

   a. Consultez le Guide de démarrage rapide d'Amazon Sidewalk.
   b. Téléchargez un kit SDK pour Amazon Sidewalk.
   c. Ouverture d'Console Sidewalk Developer Service (SDS).
2. Inscrivez votre dispositif prototype

   Dans la console SDS, enregistrez votre prototype de périphérique auprès d'Amazon Sidewalk.
3. Associez votre ID Amazon Sidewalk à votre Compte AWS

   DansAWS IoTconsole, associez votre ID Amazon Sidewalk à votre Compte AWS .

   Vos appareils Amazon Sidewalk apparaissent dans leTrottoirOnglet de l'AppareilsHub de laAWS IoTconsole.
4. Complétez la configuration de l'appareil Amazon Sidewalk dans leAWS IoTconsole

   Créez les destinations et les règles dont votre appareil Sidewalk a besoin pour acheminer et formater les données pourAWSServices .

Une fois vos appareils Amazon Sidewalk authentifiés, leurs messages sont envoyés à AWS IoT Core . Vous pouvez commencer à développer vos applications métier sur leAWSCloud qui utilise les données de vos appareils Amazon Sidewalk.

Les rubriques suivantes montrent comment ajouter des périphériques Sidewalket connectez-les àAWS IoT. Avant d'ajouter vos appareils, assurez-vous que votre Compte AWS dispose des autorisations IAM requises pour effectuer les opérations suivantesProcédures.

Rubriques

Ajouter vos informations d'identification de compte Sidewalk

Vous pouvez connecter des appareils Sidewalk àAWS IoTen utilisant la stratégieAWS Management Consoleou leAWS IoTAPI sans fil. Pour intégrer votre appareil, nous allons créer un profil de connectivité sans fil pour votre appareil Sidewalk, puis ajouter une destination etAWS IoTpour le profil et les points de terminaison du trottoir.

Avant d'ajouter votre appareil, vous devez ajouter vos informations d'identification de compte Sidewalk. Vous pouvez ajouter vos informations d'identification à l'aide de l'AWS Management Consoleou leAWS IoTAPI sans fil.

Ajouter vos informations d'identification de compte Sidewalk à l'aide de la console

Pour ajouter vos informations d'identification de compte Sidewalk à partir de la console :

1. Accédez à .ProfilsPage de laAWS IoTet choisissez l'optionTrottoirOnglet.

   Note

   Veillez à utiliser l'`us-east-1`Région . Cet onglet n'apparaît pas dans la console si vous utilisez une autre région.
2. Entrez l'ID Amazon du trottoir. Vous obtenez cet ID à partir de laConsole Sidewalk Developer Service (SDS)lors de la conception de votre produit Sidewalk. Pour de plus amples informations, veuillez consulterConcevez votre produit Sidewalk.
3. Charger leAppServerPrivateKey, qui est la clé de serveur fournie par votre fournisseur. La .AppServerPrivateKeyest la clé privée ED25519 (`app-server-ed25519-private.txt`), qui est une valeur hexadécimale de 64 chiffres que vous générez à l'aide de l'outil de génération de certificat Sidewalk lors de la conception de votre produit Sidewalk. Pour de plus amples informations, veuillez consulterConcevez votre produit Sidewalk.
4. Pour ajouter vos informations d'identification Sidewalk, choisissezAjout d'informations d'identification.

Ajouter vos informations d'identification de compte Sidewalk à l'aide de l'API

Vous pouvez utiliser la stratégieAWS IoTAPI sans fil pour ajouter vos informations d'identification de compte Sidewalk. La liste suivante décrit les actions d'API.

AWS IoTActions API sans fil pour le compte Sidewalk

- AssociateAwsAccounWithPartnerAccount
- DisasociateAWSAccountFromPartnerAccount
- GetPartnerAccount
- ListPartnerAccounts
- UpdatePartnerAccount

Pour obtenir la liste complète des actions et types de données disponibles pour créer et gérer AWS IoT Core Pour les ressources LoRaWAN, consultez leAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIpour ajouter un compte

Vous pouvez utiliser la stratégieAWS CLIpour associer un compte Sidewalk à votre Compte AWS en utilisant la stratégieassociation-aws-compte-avec-partenaire-compte-, comme illustré par l'exemple suivant.

> Note
>
> Vous pouvez également effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici.

```
aws iotwireless associate-aws-account-with-partner-account \
    --sidewalk
 AmazonId="12345678901234",AppServerPrivateKey="a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7
```

Étapes suivantes

Maintenant que vous avez ajouté les informations d'identification et configuré un profil de connectivité sans fil, vous pouvez ajouter une destination pour votre périphérique Sidewalk. Les informations d'identification que vous avez ajoutées s'affichent dans laProfilsPage de laAWS IoT, sur la consoleTrottoirOnglet. Vous allez définir un nom de rôle et un nom de règle pour la destination qui peut acheminer les messages envoyés à partir de vos appareils. Pour plus d'informations, consultez Ajouter une destination pour votre appareil Sidewalk (p. 1127).

Ajouter une destination pour votre appareil Sidewalk

Avant de pouvoir ajouter un AWS IoT Core pour la destination LoRaWan et créez une règle pour le routage des messages envoyés depuis votre périphérique Sidewalk, vous devez avoir créé un profil de connectivité sans fil. Pour créer le profil, enregistrez d'abord votre appareil Sidewalk, puis ajoutez les informations d'identification à votre Compte AWS . Pour plus d'informations, consultez Ajouter vos informations d'identification de compte Sidewalk (p. 1126).

La création d'une destination Strottoir est similaire à la façon dont vous créez une destination pour vos appareils LoRaWan. La section suivante illustre comment créer une destination à l'aide de l'AWS Management Consoleou de l'API.

Ajouter une destination à l'aide de la console

Vous pouvez ajouter votre destination Strottoir à partir de laDestinationsPage de laAWS IoTconsole

Spécifiez les champs suivants lors de la création d'un AWS IoT Core pour la destination LoRaWan, puis choisissezAjout de destination.

- Informations sur la destination

  Saisissez uneNom de destinationet une description facultative de votre destination. Pour leNom de destination, saisissez**`SidewalkDestination`**. Vous pouvez éventuellement saisir une description, telle que**`This is a destination for Sidewalk devices`**.
- Nom de la règle

  La .AWS IoTqui est configurée pour traiter les données du périphérique. Votre destination a besoin d'une règle pour traiter les messages qu'elle reçoit. Entrez un nom de règle (disons**`SidewalkRule`**), puisCopierpour copier le nom de la règle que vous allez entrer lors de la création de laAWS IoTRègle Vous pouvez choisirCréer une règlePour créer la règle maintenant ou accédez à l'RèglesHub de laAWS IoTet créez une règle avec le nom que vous avez copié.

Pour plus d'informations surAWS IoTpour les destinations, consultezCréer des règles pour traiter les messages de l'appareil LoRaWan (p. 1082).

- Nom de rôle

  Rôle IAM qui donne aux données de l'appareil l'autorisation d'accéder à la règle nommée dansRule name (Nom de la règle). Pour créer le rôle IAM, suivez les étapes décrites dansCréer un rôle IAM pour vos destinations (p. 1080). Lors de la création du rôle :

  - PourSélectionner le type d'entité de confiance, choisissez Service AWS et choisissezIoTcomme service.
  - Saisissez**`SidewalkRole`**pour laNom de rôle.
  - Utilisez le même document de stratégie comme décrit dansCréer un rôle IAM pour vos destinations (p. 1080).

  Pour plus d'informations sur les rôles IAM, consultez.Utilisation de rôles IAM.

Ajouter une destination à l'aide de l'API

Les listes suivantes décrivent les actions d'API qui effectuent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'une destination.

AWS IoTActions d'API sans fil pour les profils de service

- CreateDestination
- GetDestination
- ListDestinations
- UpdateDestination
- DeleteDestination

Pour obtenir la liste complète des actions et types de données disponibles pour créer et gérer AWS IoT Core Pour les ressources LoRaWAN, consultez leAWS IoTRéférence d'API sans fil.

Comment utiliser la stratégieAWS CLIpour ajouter une destination

Vous pouvez utiliser la stratégieAWS CLIPour ajouter une destination à l'aide de l'CREATE FUNCTIONCommande de l' L'exemple suivant crée une destination.

```
aws iotwireless create-destination \
    --name SidewalkDestination \
    --expression-type RuleName \
    --expression SidewalkRule \
    --role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

L'exécution de cette commande crée une destination avec le nom de destination, le nom de règle et le nom de rôle spécifiés. Pour plus d'informations sur les noms de règles et de rôles pour les destinations, consultezCréer des règles pour traiter les messages de l'appareil LoRaWan (p. 1082)andCréer un rôle IAM pour vos destinations (p. 1080).

Pour plus d'informations sur les CLI que vous pouvez utiliser, consultez.AWS CLIRéférence.

Étapes suivantes

Maintenant que vous avez ajouté la destination, vous pouvez créer la règle de destination pour votre périphérique Sidewalk qui acheminera les messages vers d'autres services. Pour plus d'informations, consultez Créer des règles pour traiter les messages des périphériques de trottoir (p. 1129).

Créer des règles pour traiter les messages des périphériques de trottoir

AWS IoTpeuvent recevoir les messages des appareils Sidewalk et les acheminer vers d'autres services. AWS IoT Core pour LoRaWAN (p. 1079)associer un périphérique Sidewalk à la règle qui traite les données de message du périphérique à envoyer à d'autres services.

Vous pouvez utiliser une règle existante pour votre destination. Dans cette section, nous allons créer la règle,`SidewalkRule`, que vous avez spécifié lors de la création de la destination Strottoir, comme décrit dansAjouter une destination pour votre appareil Sidewalk (p. 1127). Lors de la création de la règle, nous allons créer unAWS LambdaPour republier le message dans unAWS IoTsujet.

Créer une règle de destination de trottoir

Accédez à .RèglesHub de laAWS IoTet exécutez les étapes suivantes.

1. ChoisissezCréer une règlePour créer une règle pour la destination.
2. Saisissez le nom**`SidewalkRule`**pour laNomet spécifiez unDescriptionPour la règle (par exemple,**`Sidewalk rule for lambda action to republish a topic`**).
3. Modifiez l'instruction de requête par défaut en**`SELECT *`**Pour que toutes les actions associées à la règle soient exécutées. Conserver la version SQL sur`2016-03-23`.
4. Sous Définissez une ou plusieurs actions, choisissez Ajouter une action.
5. Pour l'action de règle, choisissezEnvoyer un message. fonction Lambda, puisConfigurer une action.
6. Vous pouvez choisir une fonction Lambda existante ou en créer une. Dans cet exemple, nous allons créer une fonction Lambda. ChoisissezCréer une fonction Lambda.

Créez votre fonction à l'aide deAWS Lambda

ChoisirCréer une fonction LambdaOuverture d' Fonctionsde la console Lambda. Procédez comme suit.

1. Pour créer votre fonction, choisissezCréer à partir de zéro.
2. PourNom de la fonction, saisissez un nom (par exemple,**`Sidewalk_Handler`**), choisissez`Python 3.8`commeRuntime (Exécution), puisCréation d'une fonction.
3. Cliquez sur l'ongletlambda.pyFonction dans le documentSource du codede la console.
4. Dans le corps de la fonction, supprimez tout code à l'intérieur du corps de la fonction et ajoutez une instruction d'impression pour votre fonction Lambda. Vous pouvez également décoder en base64 le`PayloadData`pour recevoir les données d'application que votre appareil envoie àAWS IoT. Voici un exemple de fonction Lambda.

```
import json
import base64

def lambda_handler(event, context):

    message = json.dumps(event)
    print (message)

    payload_data = base64.b64decode(event["PayloadData"])
    print(payload_data)
    print(int(payload_data,16))
```

5. Pour déployer votre code de fonction, choisissezdeploy.

6. Revenez à laRèglesHub de la console et actualisez la page. Choisissez la fonction Lambda que vous avez créée et choisissezAjout d'action.

Republiez un message dans unAWS IoTtopic

Vous pouvez ajouter une seconde action pour republier un message dans unAWS IoTà partir de la rubriqueRèglesHub de la console.

1. Choisissez Add action.
2. ChoisissezRepubliez un message dans unAWS IoTtopicet choisissezConfigurer une action.
3. Saisissez**`project/sensor/observed`**pour laRubriqueet assurez-vous que la stratégieQualité de servicea la valeur0 - Le message est remis zéro ou plusieurs fois.
4. Choisissez Create Role (Créer le rôle). SaisissezSidewalkrePublishRolePour le nom de rôle, puis choisissezCréation d'un rôle.
5. Choisissez Add action.

   Les deux actions apparaissent dans leRèglesHub de laAWS IoTconsole
6. Choisissez Create rule.

   La règle apparaît dans la fenêtreRèglesqui affiche la liste des règles.

Étapes suivantes

Maintenant que vous avez créé la règle de destination pour votre appareil Sidewalk, vous pouvez connecter votre appareil et observer les messages sur la rubrique à laquelle vous vous êtes abonné. Pour plus d'informations, consultez Connect votre appareil Sidewalk et affichez le format de métadonnées de liaison montante (p. 1130).

Connect votre appareil Sidewalk et affichez le format de métadonnées de liaison montante

Après avoir ajouté vos informations d'identification Sidewalk et ajouté la destination, vous pouvez provisionner vos points de terminaison Sidewalk et connecter votre appareil.

Connect de votre dispositif Strowalk

Vous pouvez provisionner votre appareil en tant que point de terminaison Sidewalk en générant les certificats de périphérique et les certificats de serveur d'applications à partir de laConsole Sidewalk Developer Service (SDS). Pour de plus amples informations, veuillez consulterProvisionner et configurer vos points de terminaison de trottoir.

Une fois que vous avez connecté votre appareil, vous verrez votre dispositif Sidewalk dans leAppareilsPage de laAWS IoT, sur la consoleTrottoirOnglet. Lorsque votre appareil est connecté et commence à envoyer des données, vous verrez la date et l'heure de laDernière liaison montante reçue àchamp.

Afficher le format des messages de liaison montante

Une fois que vous avez connecté votre appareil, vous pouvez vous abonner à la rubrique (par exemple,`project/sensor/observed`) que vous avez spécifié lors de la création de la règle de

destination Strottoir pour observer les messages de liaison montante à partir du périphérique. Pour s'abonner à la rubrique, accédez à laClient de test MQTTsur leTestPage de laAWS IoT, saisissez le nom de rubrique (par exemple,`project/sensor/observed`), puis choisissezS'abonner.

L'exemple suivant illustre le format des messages de liaison montante envoyés par les périphériques Sidewalk àAWS IoT. La .`WirelessMetadata`contient des métadonnées relatives à la requête de message.

```
{
    “WirelessDeviceId”: “8dccc5978df94950b57a58116c6f52e6",
    “PayloadData”: “AAAAAAAA//8=“,
    “TransmitMode”: “0",
    “WirelessMetadata”:
    "WirelessMetadata(sidewalk=Sidewalk(seq=1983,
     messageType=null, cmdExStatus=null,
     nackExStatus=null))
  }
}
```

Le tableau suivant présente une définition des différents paramètres dans les métadonnées de liaison montante. Pour plus d'informations sur les paramètres`WirelessDeviceID`,`PayloadData`, et`messageType`, voirSidewalkDeviceandSideWalksEndDatatoDevice.

Paramètres de métadonnées de liaison montante

Paramètre	Description	Type	Obligatoire
`TransmitMode`	Mode de transmission des données envoyées à partir du périphérique sans fil. Peut être`0`pour`unconfirmed`Mode,`1`pour`confirmed`, et`2`pour`unused`.	Integer	Oui
`Sidewalk.CmdExStatus`	CommanderunHeure de l'état. Les messages de type réponse doivent comprendre le code d'état,`COMMAND_EXEC_STATUS_SUCCESS`. Toutefois, les notifications peuvent ne pas inclure le code d'état.	Énumération	Non
`Sidewalk.NackExStatus`	État de réponse nack, qui peut être`RADIO_TX_ERROR`ou`MEMORY_ERROR`.	Tableau de chaînes	Non

Intégration Alexa Voice Service (AVS) pourAWS IoT

Intégration Alexa Voice Service (AVS) pourAWS IoTest une nouvelle fonctionnalité qui associe de manière rentable Alexa Voice à n'importe quel appareil connecté sans entraîner decoûts de messagerie. AVS pourAWS IoTréduit le coût et la complexité de l'intégration d'Alexa. Cette fonctionnalité tire partie d'AWS IoT pour décharger les tâches audio intensives de calcul et de mémoire de l'appareil vers le cloud. Étant donnée la réduction du coût de la nomenclature des matériaux d'ingénierie (eBoM) qui en résulte, les fabricants d'appareils peuvent désormais associer Alexa à des appareils IoT dont les ressources sont limitées, et permettre aux consommateurs de parler directement à Alexa à certains endroits de leur maison, de leur bureau ou de leur chambre d'hôtel. Ils profitent ainsi d'une expérience ambiante.

Actuellement, les appareils IoT de maison connectée sont composés de microcontrôleurs (MCU) économiques qui ont une mémoire limitée pour exécuter les systèmes d'exploitation en temps réel. Auparavant, les solutions AVS pour les produits dans lesquels Alexa était intégré nécessitaient des appareils coûteux basés sur des processeurs d'applications, avec plus de 50 Mo de mémoire et fonctionnant sous Linux ou Android. Ces exigences matérielles coûteuses n'ont pas permis d'intégrer Alexa Voice sur des appareils IoT ayant des ressources limitées. il n'est pas possible d'utiliser ces appareils. AVS pourAWS IoTactive la fonctionnalité intégrée Alexa sur les microcontrôleurs, tels que les processeurs ARM Cortex-M avec moins de 1 Mo de RAM intégrée. Pour ce faire, AVS décharge la mémoire et les tâches de calcul vers un appareil virtuel avec Alexa intégré dans le cloud. Cela réduit le coût de la nomenclature eBoM jusqu'à 50 %.

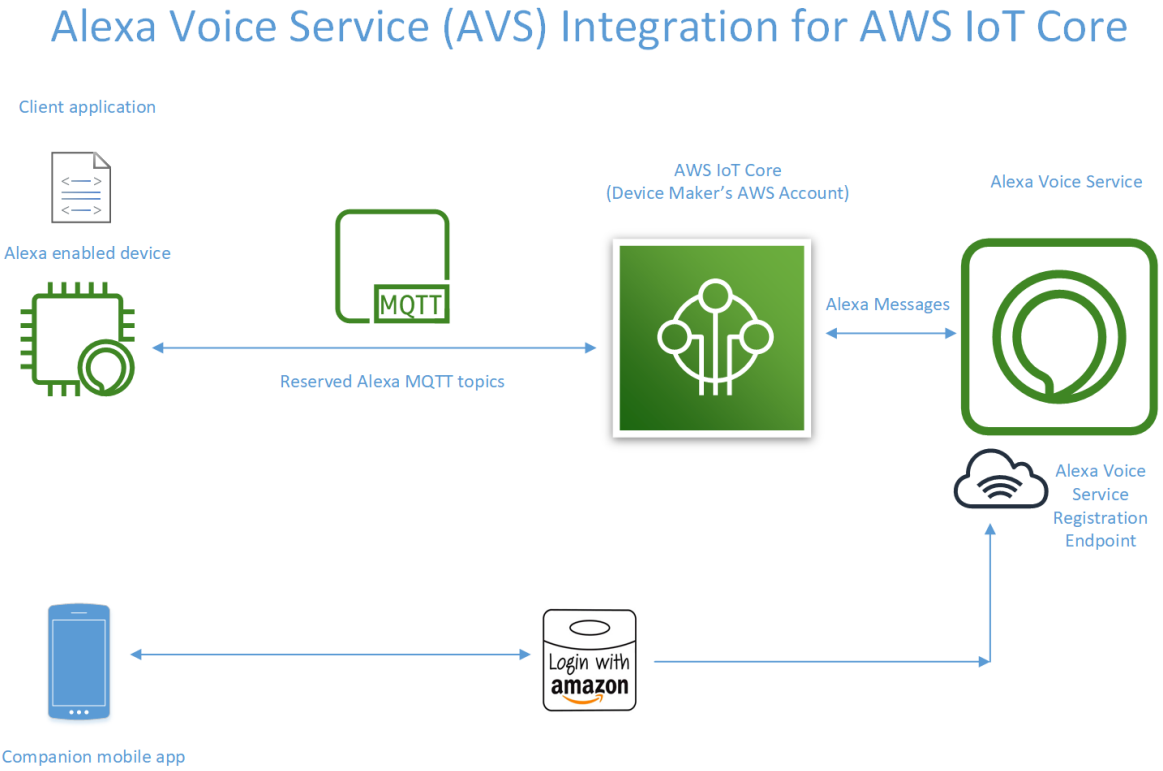
Pour de plus amples informations sur les processeurs de la série Arm Cortex-M, veuillez consulter Arm ou Wikipedia. Pour plus d'informations sur la configuration matérielle requise pour les produits intégrés Alexa, consultez la section Dimensionnement de l'UC, de la mémoire et du stockage pour votre appareil Alexa intégré sur le portail de développement Amazon Alexa.

> **Note**
>
> AVS pourAWS IoTest disponible dans tous les Région AWS s oùAWS IoTest disponible sauf dans les régions Chine (Beijing et Ningxia). Pour la liste actuelle desAWSRégions, consultez laAWSTable des régions.

AVS pourAWS IoTVoici les trois composantes d' :

- Un ensemble de rubriques MQTT réservées pour transférer des messages audio entre les appareils compatibles Alexa et AVS.
- Un appareil virtuel compatible Alexa dans le cloud qui déplace les tâches liées à la récupération multimédia, au décodage audio, au mixage audio et à la gestion de l'état de l'appareil physique vers l'appareil virtuel.
- Un ensemble d'API qui prennent en charge la réception et l'envoi de messages via les rubriques réservées, l'interface avec le microphone et le haut-parleur de l'appareil et la gestion de l'état de l'appareil.

Le diagramme suivant illustre la façon dont ces composants fonctionnent ensemble. Il montre également de quelle manière les fabricants d'appareils utilisent le service Login with Amazon pour authentifier AVS.



Les fabricants d'appareils ont deux options pour bien démarrer avec l'intégration AVS pour AWS IoT.

- Kits de développement— Les kits de développement lancés par nos partenaires facilitent la mise en route. Le kit NXP i.MX RT 106 A et le kit Qualcomm Home Hub 100 pour Amazon AVS sont les deux premiers kits disponibles sur le marché. Vous pouvez les trouver sur la page des kits de développement pour AVS. Les kits comprennent une connectivité prête à l'emploi vers AWS IoT, des algorithmes audio AVS éligibles pour la détection des voix lointaines, la suppression des échos, le mot-clé Alexa ainsi que le code d'application AVS pour AWS IoT. Vous pouvez utiliser le code d'application pour réaliser rapidement un prototype d'appareil et importer l'implémentation sur la carte de microcontrôleur de votre choix pour tester et configurer l'appareil dès que vous êtes prêt.
- Code d'application personnalisé côté appareil— Les développeurs peuvent également écrire un AVS personnalisé pourAWS IoTà l'aide de l'API accessible au public. La documentation de cette API est disponible sur la page destinée aux développeurs AVS. Vous pouvez télécharger le logiciel FreeRTOS etAWS IoTKit SDK de périphérique depuis la console FreeRTOS (https://console.aws.amazon.com/freertos/) ouGitHub.

Pour commencer à l'aide d'un kit de développement NXP i.MX 106A, consultezDémarrage avec l'intégration Alexa Voice Service (AVS) pourAWS IoTsur un appareil NXP.

Démarrage avec l'intégration Alexa Voice Service (AVS) pourAWS IoTsur un appareil NXP

Le kit de développement NXP i.MX 106A vous permet d'utiliser la version préliminaire de l'intégration Alexa Voice Service (AVS) pourAWS IoTà l'aide d'un compte NXP préconfiguré. Après avoir utilisé la version préliminaire de la fonctionnalité avec le compte NXP, vous devez personnaliser le microprogramme, le

code source de l'application et l'application mobile NXP fournis avec le kit afin de pouvoir utiliser votre propre compte. Cette rubrique couvre le processus d'utilisation de la version préliminaire avec le compte préconfiguré et de personnalisation de votre appareil avec votre propre compte.

Rubriques

Utiliser la version préliminaire de l'intégration Alexa Voice Service (AVS) pourAWS IoTavec un compte NXP préconfiguré

Prerequisites

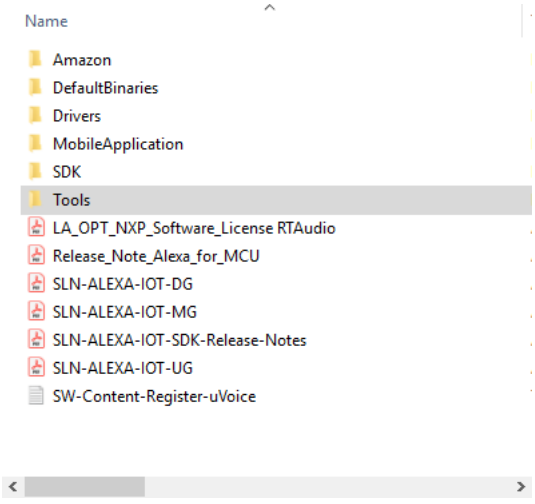
Pour suivre ces étapes, vous avez besoin des ressources suivantes.

- Kit de développement NXP i.MX 106A
- Un ordinateur Mac, Windows 7/10 ou Linux
- Un pilote série qui fonctionne avec votre ordinateur
- Un appareil mobile Android ou iOS
- Android Debug Bridge (ADB)
- Un compte Amazon Alexa

Installer et démarrer le kit de développement

1. Activez le kit de l'appareil. Le kit est livré avec une carte Get Started Onboarding. Cette carte contient les instructions nécessaires pour activer le kit et acquérir le package logiciel nécessaire et les fichiers de conception de référence. Le package logiciel contient le code source du SDK et l'application complémentaire Android (`VoiceCompanionApp.apk`). Si vous utilisez un appareil iOS, vous devez demander l'accès à l'application iOS TestFlight à votre représentant NXP local.

   Après avoir téléchargé et extrait le package logiciel, la structure de fichier suivante s'affiche.



Le dossier `MobileApplication` contient l'application mobile Android.

Le dossier `Tools` contient les scripts que vous utilisez pour configurer votre appareil.

Vous pouvez également trouver la documentation suivante :

- `SLN-ALEXA-IOT-DG.pdf` dans le Guide du développeur NXP
- `SLN-ALEXA-IOT-MG.pdf` dans le Guide de migration NXP
- `SLN-ALEXA-IOT-UG.pdf` dans le Guide de l'utilisateur NXP

2. Démarrez le kit de l'appareil. Le câble séparateur USB livré avec le kit dispose de connexions pour l'alimentation et pour les données.

   a. Branchez les deux connexions USB-A à votre ordinateur.

   b. Branchez le connecteur USB-C à votre kit. Votre configuration peut s'apparenter à l'image suivante.

Lorsque la carte est alimentée, le voyant D1 s'allume et devient vert. Le voyant D2 (le plus proche du haut-parleur) indique l'état de l'appareil. Lorsque l'appareil s'allume, ce voyant clignote de plusieurs couleurs. Lorsque l'appareil exécute le code d'application, le voyant D2 clignote en jaune. Lorsque l'initialisation de l'appareil est terminée, le voyant D2 est orange, ce qui indique qu'il attend l'ajout des informations d'identification Wi-Fi à l'appareil.

Le Guide de l'utilisateur NXP contient une description des contrôles physiques de l'appareil.

3. Utilisez l'application de terminal pour comprendre le fonctionnement de l'application cliente sur l'appareil. Vous pouvez également l'utiliser pour déboguer et vous assurer que l'état de l'appareil est correct.

   Entrez les commandes suivantes pour commencer à utiliser l'application :

   - `help`— Affiche la liste des commandes disponibles.
   - `enable_usb_log`— Active la connexion sur l'appareil. Cela vous aide à comprendre ce que fait l'application.
   - `logs`— Affiche les dernières lignes des journaux. À ce stade, les journaux doivent indiquer que l'appareil attend les informations d'identification Wi-Fi.

Connectez l'appareil à AWS IoT et à Amazon Alexa

1. Installez l'application mobile appropriée. Si vous utilisez un appareil Android, vous pouvez utiliser ADB dans le terminal pour installer le fichier `VoiceCompanionApp.apk`. Si vous utilisez un appareil iOS, utilisez l'application TestFlight.
2. Mettez en service les informations d'identification Wi-Fi. Vous pouvez mettre en service les informations d'identification Wi-Fi à l'aide de l'application mobile complémentaire ou du terminal.

   a. Procédez comme suit pour mettre en service les informations d'identification Wi-Fi à l'aide de l'application mobile.

      i. Lorsque l'appareil démarre à partir de son nouvel état d'usine, il crée un point d'accès Wi-Fi. Ouvrez l'application mobile complémentaire et choisissez WIFI PROVISION (MISE EN SERVICE WI-FI) pour vous connecter à ce point d'accès.

      ii. Choisissez un réseau dans la liste.

iii. Entrez votre mot de passe et choisissez Send (Envoyer) pour transmettre les informations d'identification au kit de l'appareil.

b. Dans l'application de terminal, entrez la commande suivante.

```
setup YourWiFiNetworkName YourWiFiNetworkPassword
```

c. Choisissez Enter (Entrer).

3. Authentifiez le kit de l'appareil et connectez-vous à AWS IoT et à Amazon Alexa. Assurez-vous que votre appareil mobile et le kit de l'appareil utilisent le même réseau Wi-Fi, afin que les deux appareils puissent communiquer.

   Pour ce faire, appuyez sur le bouton Discover (Détecter) de l'application mobile. Une fois la détection terminée, la liste des numéros de série des kits d'appareils disponibles près de vous s'affiche. Si plusieurs appareils sont disponibles, vous pouvez confirmer le numéro de série unique de votre kit en saisissant **`serial_number`** dans l'application du terminal.

4. Choisissez le kit d'appareil à utiliser. L'application mobile utilise le service Login with Amazon pour vous demander vos informations d'identification utilisateur Amazon. Le kit de l'appareil commence à enregistrer l'appareil avec AVS. Le voyant D2 s'allume en violet pour indiquer que l'enregistrement de l'appareil a commencé.

   Note

   Si l'application shopping Amazon est déjà installée sur votre appareil mobile, l'application mobile complémentaire utilise automatiquement le compte Amazon qui est connecté à l'application shopping.

Une fois que vous êtes connecté à votre compte Amazon, l'application complémentaire effectue les étapes suivantes :

1. Le kit de l'appareil reçoit le jeton d'accès du service Login with Amazon et commence à connecter l'appareil à AWS IoT et à AVS. L'application mobile affiche le pourcentage de progression. Le voyant D2 de l'appareil s'allume en orange.
2. Le voyant D1 clignote en vert toutes les 500 millisecondes jusqu'à ce que l'appareil se connecte à AWS IoT.
3. Lorsque l'appareil se connecte à AWS IoT, le kit de l'appareil commence à connecter l'appareil à AVS. Le voyant D1 clignote en vert toutes les 250 millisecondes.
4. Lorsque l'appareil se connecte à AVS, le numéro de série du kit de l'appareil devient jaune dans l'application mobile. L'application mobile affiche le message Complete (Terminé). Le voyant D1 s'éteint et l'appareil émet un son. L'appareil est maintenant connecté au compte AWS IoTdu NXP.

Vous pouvez essayer quelques commandes Alexa Voice, par exemple :

- « Alexa, quel temps fait-il ? »
- « Alexa, lance un bulletin d'information. »
- « Alexa, joue de la musique. »

Utiliser votreAWSet les comptes de développeur Alexa Voice Service pour configurer AVS pourAWS IoT

Prerequisites

Pour cette procédure, vous avez besoin des ressources suivantes.

- Kit de développement NXP i.MX 106A
- MCUXPresso v10.3.1
- Sonde de débogage J-Link Segger
- Un Mac
- Un appareil mobile Android
- Android Studio
- Android Debug Bridge
- Un compte Amazon Alexa
- Un Compte AWS

Configurez votre compte AWS IoT et mettez en service votre appareil

1. Générez des informations d'identification pour vous authentifier avec AWS IoT. Pour communiquer avec AWS IoT, tous les appareils doivent disposer d'un certificat, d'une clé privée et d'un certificat d'autorité de certification racine. Suivez les instructions de Enregistrement d'un appareil dans le registre pour enregistrer votre appareil auprès de AWS IoT. Pour de plus amples informations sur les certificats X.509, veuillez consulter Certificats client X.509 (p. 236).
2. Accédez au dossier racine et ouvrez le Guide du développeur NXP. Suivez les instructions de la « Section 9 relative au système de fichiers » pour convertir vos clés et certificats client dans un format binaire que le système de fichiers NXP peut lire.

   Utilisez le script et les commandes suivants pour convertir les certificats et les clés. Le script `bin_dump.py` se trouve dans le dossier `Tools\Ivaldi.zip\Scripts\sln_alexa_iot_utils`.

   ```
   python3 bin_dump.py CertificateName-certificate.pem.crt CertificateName-
   certificate.pem.crt.bin
   ```

   ```
   python3 bin_dump.py CertificateName-private.pem.key CertificateName-private.pem.key.bin
   ```

Pour de plus amples informations sur les options de génération d'informations d'identification pour les appareils de production à grande échelle, veuillez consulter Mise en service des appareils (p. 728).

Pour configurer votre appareil dans la console de développeur Alexa Voice Service

1. Accédez au dossier racine et ouvrez le Guide de migration NXP.
2. Pour obtenir les signatures MD5 et SHA256 dont vous avez besoin pour créer une clé d'API AVS, suivez les instructions de la « Section 2 relative à la création d'un magasin de clés pour l'application complémentaire Android AVS » dans le Guide de migration NXP. (Cela implique de naviguer vers la console Alexa Voice Service Developer.)
3. Pour créer un profil de sécurité Login with Amazon et créer un produit AVS, suivez les instructions de la « Section 3 » du Guide de migration NXP.

Pour recréer l'application mobile Android

1. Ouvrez le projet VoiceCompanionApp dans Android Studio. Le fichier `VoiceCompanionApp.apk` est inclus dans le package logiciel NXP.
2. Dans Android Studio, ajoutez la clé API que vous avez générée dans la procédure précédente au fichier `api_key.txt` du dossier assets . Vous synchronisez ainsi l'application mobile complémentaire avec votre profil de sécurité AVS. L'application utilise la clé API lorsque vous vous connectez à

l'aide du service Login with Amazon. Lorsque l'application obtient un code d'autorisation d'Amazon, elle le transfère au microprogramme de l'appareil. L'appareil obtient ensuite des jetons d'accès et d'actualisation du service Login with Amazon pour passer des appels à AVS.

3. Dans Android Studio, choisissez Build Project (Construire un projet) et Build Project (Générer un ensemble/APK signé). Pour de plus amples informations, veuillez consulter la « Section 4 relative à la création et à la génération de l'application mobile » dans le Guide de migration NXP. Pour de plus amples informations sur l'autorisation avec Alexa de cette façon, veuillez consulter la page relative à l' autorisation à partir d'une application complémentaire (Android/iOS).
4. Installez l'application mobile mise à jour sur votre appareil mobile. Configurez ADB sur votre Mac et installez le fichier APK Android mis à jour sur votre téléphone mobile à l'aide de la console du terminal. Si vous avez installé l'application mobile pour prévisualiser l'AVS pourAWS IoTavec le compte NXP préconfiguré, vous devez désinstaller, puis réinstaller l'application.

Pour mettre à jour l'application cliente avec vos informations d'identification AWS

1. Suivez les instructions de la section 3 et de la section 4 du Guide du développeur NXP pour apporter les modifications nécessaires au pilote Segger J-Link. Ces instructions expliquent également comment importer les projets `Bootloader`, `ais_demo` et `bootstrap` à l'aide de l'IDE MCUXPresso.
2. Suivez les instructions pour mettre à jour le code source et reconstruire l'application dans la « Section 7 relative aux modifications du microprogramme RT106A » dans le Guide de migration NXP. Nous vous recommandons d'utiliser MCUXPresso v10.3.1.

   Ces modifications du code source permettent au microprogramme de l'appareil de communiquer avec votre Compte AWS et votre produit AVS au lieu du compte NXP par défaut.

Pour configurer et connecter votre appareil à AWS IoT et à Amazon Alexa

1. Réinitialisez l'appareil aux paramètres d'usine par défaut. Si vous avez prévisualisé l'AVS pourAWS IoTavec le compte NXP préconfiguré, vous devez réinitialiser le microprogramme de l'appareil avant de vous reconnecter avec l'application mobile complémentaire. Cela garantit que l'application mobile et l'appareil utilisent votre profil de sécurité. Assurez-vous que l'appareil est connecté à votre ordinateur. Appuyez sur SW1 pendant 10 secondes pour lancer la réinitialisation d'usine.

   Le Guide de l'utilisateur NXP inclus dans le package logiciel NXP contient une description des contrôles physiques de l'appareil.
2. Reprogrammez le microprogramme pour utiliser le certificat et la clé que vous avez générés pour votre appareil. Ajoutez la clé et le certificat aux fichiers binaires `Bootstrap` et `Ais_demo` mis à jour que vous avez créés dans la procédure précédente. Nous vous recommandons également de reprogrammer le microprogramme avec les fichiers binaires `Intelligent toolbox`, `app_crt` et `CA_crt` par défaut.

   Pour obtenir des instructions, veuillez consulter la « Section 5 relative à la création et à la programmation » dans le Guide du développeur NXP.
3. Suivez les instructions de la procédure précédente, Connecter l'appareil à AWS et à Amazon Alexa (p. 1136). Cela vous connecte à AVS pourAWS IoTavec votre propre profil de sécurité et vos propres informations d'identification.

AWS IoTKit SDK pour appareils, kits SDK mobiles etAWS IoTClient de l'appareil

Cette page résume lesAWS IoTDes kits SDK pour appareils, des bibliothèques open source, des guides pour développeurs, des exemples d'applications et des guides de portage pour vous aider à créer des solutions IoT innovantes avecAWS IoTet votre choix de plates-formes matérielles.

Ces kits SDK sont destinés à être utilisés sur votre appareil IoT. Si vous développez une application IoT pour une utilisation sur un appareil mobile, consultez laKits SDK AWS Mobile (p. 1142). Si vous développez une application IoT ou un programme côté serveur, consultez laKits SDK AWS (p. 70).

Kits SDK pour les appareils AWS IoT

Les kits SDK pour les appareils AWS IoT incluent des bibliothèques open source, des manuels pour développeurs avec des exemples, ou encore des manuels de portage afin de vous permettre de créer des produits et des solutions IoT innovantes sur les plateformes matérielles de votre choix.

Ces kits SDK vous aident à connecter vos appareils IoT àAWS IoTà l'aide des protocoles MQTT et WSS.

C++

AWS IoT Kit SDK des appareils C++

La .AWS IoTLe kit SDK des appareils C++ permet aux développeurs de générer des applications connectées à l'aideAWSet l'AWS IoTAPI. Ce kit SDK a été conçu en particulier pour les appareils qui ne sont pas limités en ressources et qui nécessitent des fonctions avancées, telles que la mise en file d'attente des messages, la prise en charge du multithreading et les dernières fonctions de langue. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT SDK de périphérique C++ v2 sur GitHub
- AWS IoT Lisez-moi du SDK de périphérique C ++ v2
- AWS IoT Exemples SDK des appareils C++ v2

Python

AWS IoT Kit SDK des appareils pour Python

Le kit SDK des appareils AWS IoT pour Python permet aux développeurs d'écrire des scripts Python afin d'utiliser leurs appareils pour accéder à la plateforme AWS IoT via le protocole MQTT ou MQTT sur WebSocket. En connectant leurs appareils àAWS IoT, les utilisateurs peuvent utiliser de façon sécurisée les règles et le courtier de messages, ainsi que les shadows fournis parAWS IoTet avec d'autresAWSServices commeAWS Lambda, Kinesis et Amazon S3, et bien plus encore.

- AWS IoT Kit SDK des appareils pour Python v2 sur GitHub
- AWS IoT Kit SDK des appareils pour Python v2 - Readme
- AWS IoT Kit SDK des appareils pour Python v2 Samples

- AWS IoT Documentation SDK des périphériques pour Python v2 API

JavaScript

AWS IoT Kit SDK des appareils pour JavaScript

Le package aws-iot-device-sdk.js permet aux développeurs d'écrire des applications JavaScript qui ont accès à AWS IoT à l'aide du protocole MQTT ou MQTT sur WebSocket. Il peut être utilisé dans des environnements Node.js et des applications de navigateur. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour JavaScript v2 sur GitHub
- AWS IoT Kit SDK des périphériques pour JavaScript v2 – Readme
- AWS IoT Samples SDK des appareils pour JavaScript v2
- AWS IoT Documentation du kit SDK des appareils pour JavaScript v2

Java

AWS IoT Kit SDK des appareils pour Java

Le kit SDK des appareils AWS IoT pour Java permet aux développeurs Java d'accéder à la plateforme AWS IoT via le protocole MQTT ou MQTT sur WebSocket. Le kit SDK est intégré à la prise en charge des shadows. Vous pouvez accéder au service Shadows à l'aide des méthodes HTTP, notamment GET, UPDATE et DELETE. Le kit SDK prend également en charge un modèle d'accès aux shadows simplifié, qui permet aux développeurs d'échanger des données avec des shadows en utilisant uniquement des méthodes getter et setter, sans avoir à sérialiser ou désérialiser des documents JSON. Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour Java v2 sur GitHub
- AWS IoT Kit SDK des périphériques pour Java v2 – Readme
- AWS IoT Samples SDK des appareils pour Java v2

Kit SDK des appareils AWS IoT pour Embedded C

Note

Ce kit SDK est destiné à être utilisé par les développeurs de logiciels intégrés expérimentés.

La . AWS IoT Device SDK for Embedded C (C-SDK) est un ensemble de fichiers source C sous la licence open source MIT qui peuvent être utilisés dans des applications incorporées pour connecter en toute sécurité des appareils IoT à AWS IoT Core . Il comprend un MQTT, JSON Parser etAWS IoTBibliothèque Device Shadow. Il est distribué sous forme source et est conçu pour être intégré dans un microprogramme client avec un code d'application, d'autres bibliothèques et, éventuellement, un système d'exploitation temps réel (RTOS).

Pour le provisionnement de flotte, utilisez le`v4_beta_deprecated`Version de l' AWS IoT Device SDK for Embedded C àhttps://github.com/aws/aws-iot-device-sdk-embedded-C/tree/v4_beta_deprecated. Veuillez consulter le fichier README dans cette branche pour plus de détails.

Le AWS IoT Device SDK for Embedded C est généralement destiné aux appareils à ressources limitées qui nécessitent un moteur d'exécution optimisé en langage C. Vous pouvez utiliser le kit SDK sur n'importe quel système d'exploitation et l'héberger sur n'importe quel type de processeur (par exemple, microcontrôleurs et MPU).

Pour de plus amples informations, consultez les ressources suivantes :

- AWS IoT Kit SDK des périphériques pour Embedded C GitHub
- AWS IoT Kit SDK des appareils pour Embedded C – Readme
- AWS IoT Kit SDK des appareils pour les échantillons C incorporés

Plus tôtAWS IoTVersions SDK des appareils

Ce sont des versions antérieures deAWS IoTLes kits SDK de périphérique qui ont été remplacés par les versions plus récentes répertoriées ci-dessus. Ces kits SDK reçoivent uniquement des mises à jour de sécurité et de maintenance. Ils ne seront pas mis à jour pour inclure de nouvelles fonctionnalités et ne devraient pas être utilisés sur de nouveaux projets.

- AWS IoT SDK de périphérique C++ sur GitHub
- AWS IoT Kit SDK des appareils C++
- AWS IoT Kit SDK des appareils pour Python v1 sur GitHub
- AWS IoT Kit SDK des appareils pour Python v1 - Readme
- AWS IoT Kit SDK des appareils pour Java sur GitHub
- AWS IoT Kit SDK des périphériques pour Java Readme
- AWS IoT Kit SDK des appareils pour JavaScript sur GitHub
- AWS IoT Kit SDK des périphériques pour JavaScript Readme
- Kit SDK Arduino Yún sur GitHub
- Kit SDK Arduino Yún – Readme

Kits SDK AWS Mobile

La .AWSLes SDK mobiles fournissent aux développeurs d'applications mobiles une prise en charge spécifique de la plate-forme pour les API du AWS IoT Core , la communication des appareils IoT à l'aide de MQTT et les API d'autresAWSServices .

Android

AWS Mobile SDK for Android

La .AWS Mobile SDK for Androidcontient une bibliothèque, des exemples et une documentation pour développeurs pour développer des applications mobiles connectées à l'aide deAWS. Ce kit SDK comprend également la prise en charge des communications des appareils MQTT et l'appel des API de l'outil AWS IoT Core Services . Pour de plus amples informations, consultez les ressources suivantes :

- AWS Mobile SDK for Android sur GitHub
- AWS Mobile SDK for Android Readme
- AWS Mobile SDK for Android Exemples
- AWS Mobile SDK for Android Référence d'API
- Documentation de référence de la classe AWSIOTClient

iOS

AWS Mobile SDK for iOS

La .AWS Mobile SDK for iOSest un kit de développement logiciel open source, distribué sous une licence open source Apache. La .AWS Mobile SDK for iOSfournit une bibliothèque, des exemples

de code et une documentation pour aider les développeurs à développer des applications mobiles connectées à l'aide deAWS. Ce kit SDK comprend également la prise en charge des communications des appareils MQTT et l'appel des API de l'outil AWS IoT Core Services . Pour de plus amples informations, consultez les ressources suivantes :

- AWS Mobile SDK for iOS sur GitHub
- AWS Mobile SDK for iOS Readme
- AWS Mobile SDK for iOS Exemples
- Docs de référence de classe AWSIoT dans la AWS Mobile SDK for iOS

AWS IoTClient de l'appareil

La .AWS IoTDevice Client fournit du code pour aider votre appareil à se connecter àAWS IoT, exécutez des tâches de provisionnement de flotte, prenez en charge les stratégies de sécurité des périphériques, connectez-vous à l'aide d'un tunnel sécurisé et traitez les tâches sur votre appareil. Vous pouvez installer ce logiciel sur votre appareil pour gérer ces tâches de routine afin que vous puissiez vous concentrer sur votre solution spécifique.

Note

La .AWS IoTDevice Client fonctionne avec des périphériques IoT basés sur un microprocesseur avec des processeurs x86_64 ou ARM et des systèmes d'exploitation Linux courants.

C++

AWS IoT Client de l'appareil

Pour en savoir plus sur l'AWS IoTClient de périphérique en C ++, consultez les points suivants :

- AWS IoT Client de périphérique dans un code source C++ sur GitHub
- AWS IoT Client de périphérique en C ++ Lisez-moi

Résolution des problèmes de AWS IoT

Les informations suivantes peuvent vous aider à résoudre les problèmes courants dans AWS IoT.

Tâches

Diagnostic des problèmes de connectivité

Une connexion réussie àAWS IoTnécessite :

- Une connexion valide
- Un certificat valide et actif
- Une stratégie qui permet la connexion et l'opération souhaitées

Connection

Comment trouver le point de terminaison correct ?

- L'adresse `endpointAddress` renvoyée par aws iot describe-endpoint --endpoint-type iot:Data-ATS

  ou

- L'adresse `domainName` renvoyée par aws iot describe-domain-configuration –-domain-configuration-name "***`domain_configuration_name`***"

Comment puis-je trouver la valeur SNI (Server Name Indication) correcte ?

La valeur SNI correcte est la valeur`endpointAddress`Retourné pardescribe-endpointoudescribe-domain-configuration. Il s'agit de la même adresse que le point de terminaison à l'étape précédente.

Authentication

Les appareils doivent êtreauthentifié (p. 236)Pour connecter àAWS IoTPoints de terminaison . Pour les appareils qui utilisentCertificats client X.509 (p. 236)pour l'authentification, les certificats doivent être enregistrés auprès deAWS IoTet être actif.

Comment mes appareils authentifient-ils des points de terminaison AWS IoT ?

Ajoutez le certificat d'autorité de certification (CA) AWS IoT au référentiel d'approbations de votre client. Reportez-vous à la documentation surServeur AWS IoT Core , puis suivez les liens pour télécharger le certificat CA approprié.

Ce qui est vérifié lorsqu'un périphérique se connecte àAWS IoT?

Lorsqu'un appareil tente de se connecter àAWS IoT :

1. AWS IoTvérifie si un certificat valide et une valeur SNI (Server Name Indication).
2. AWS IoTvérifie que le certificat utilisé est enregistré auprès duAWS IoTet qu'il a été activé.
3. Lorsqu'un périphérique tente d'effectuer une action dansAWS IoT, par exemple pour s'abonner ou publier un message, la stratégie attachée au certificat utilisé pour se connecter est vérifiée pour confirmer que l'appareil est autorisé à effectuer cette action.

Comment puis-je valider un certificat correctement configuré ?

Utilisez la commande OpenSSL `s_client` pour tester une connexion à un point de terminaison AWS IoT :

```
 openssl s_client -connect custom_endpoint.iot.aws-region.amazonaws.com:8443 -
CAfile CA.pem -cert cert.pem -key privateKey.pem
```

Pour plus d'informations sur l'utilisation d'`openssl s_client`, consultez la documentation OpenSSL s_client.

Comment vérifier l'état d'un certificat ?

- Affichage des certificats

  Si vous ne connaissez pas l'ID de certificat, vous pouvez voir l'état de tous vos certificats à l'aide de la commandeaws iot list-certificates.

- Afficher les détails d'un certificat

  Si vous connaissez l'ID du certificat, cette commande affiche des informations plus détaillées sur le certificat.

  ```
  aws iot describe-certificate --certificate-id "certificateId"
  ```

- Examinez le certificat dans la fenêtreAWS IoTConsole

  DansAWS IoTconsoleDans le menu de gauche, choisissezSecure, puisCertificats.

  Choisissez le certificat que vous utilisez pour vous connecter dans la liste pour ouvrir sa page détaillée.

  Dans la page détaillée du certificat, vous pouvez voir son état actuel.

  L'état du certificat peut être modifié à l'aide de la commandeActionsdans le coin supérieur droit de la page de détails.

Authorization

AWS IoTUtilisation des ressourcesStratégies AWS IoT Core (p. 270)pour autoriser ces ressources à effectuerActions (p. 270). Pour qu'une action soit autorisée, leAWS IoTdoivent être accompagnées d'un document de stratégie qui autorise l'exécution de cette action.

J'ai reçu une réponse `PUBNACK` ou `SUBNACK` de l'agent. Que puis-je faire ?

Assurez-vous qu'une stratégie est attachée au certificat que vous utilisez pour appeler AWS IoT. Toutes les opérations de publication/abonnement sont rejetées par défaut.

Assurez-vous que la stratégie ci-jointe autorise leActions (p. 270)Vous essayez de jouer.

Assurez-vous que la stratégie ci-jointe autorise lesresources (p. 272)qui tentent d'effectuer les actions autorisées.

J'ai unAUTHORIZATION_FAILUREdans mes journaux.

Assurez-vous qu'une stratégie est attachée au certificat que vous utilisez pour appeler AWS IoT. Toutes les opérations de publication/abonnement sont rejetées par défaut.

Assurez-vous que la stratégie ci-jointe autorise leActions (p. 270)Vous essayez de jouer.

Assurez-vous que la stratégie ci-jointe autorise lesresources (p. 272)qui tentent d'effectuer les actions autorisées.

Comment puis-je vérifier ce que la police autorise ?

DansAWS IoTconsoleDans le menu de gauche, choisissezSecure, puisCertificats.

Choisissez le certificat que vous utilisez pour vous connecter dans la liste pour ouvrir sa page détaillée.

Dans la page détaillée du certificat, vous pouvez voir son état actuel.

Dans le menu de gauche de la page détaillée du certificat, choisissezStratégiesPour afficher les stratégies attachées au certificat.

Choisissez la stratégie souhaitée pour voir sa page de détails.

Dans la page de détails de la stratégie, consultez leDocument de stratégiepour voir ce qu'il autorise.

ChoisissezModifier le document de stratégiePour apporter des modifications au document de stratégie.

Diagnostic des problèmes de règles

Cette section décrit certains éléments à vérifier lorsque vous rencontrez un problème avec la règle.

Configuration des CloudWatch Logs pour le dépannage

La meilleure façon de résoudre les problèmes de débogage que vous rencontrez avec les règles consiste à utiliser CloudWatch Logs. Lorsque vous activez CloudWatch Logs pourAWS IoT, vous pouvez voir les règles qui sont déclenchées, ainsi que leur réussite ou leur échec. Vous obtenez également des informations concernant la correspondance ou non des conditions de clause WHERE. Pour plus d'informations, consultez ContrôleAWS IoTUtilisation CloudWatch Logs (p. 373).

Le problème le plus fréquent avec les règles est celui de l'autorisation. Les journaux indiquent si votre rôle n'est pas autorisé à effectuer des opérations AssumeRole sur la ressource. Voici un exemple de journal généré par la journalisation affinée (p. 357) :

```
{
```

```
    "timestamp": "2017-12-09 22:49:17.954",
    "logLevel": "ERROR",
    "traceId": "ff563525-6469-506a-e141-78d40375fc4e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleExecution",
    "clientId": "iotconsole-123456789012-3",
    "topicName": "test-topic",
    "ruleName": "rule1",
    "ruleAction": "DynamoAction",
    "resources": {
        "ItemHashKeyField": "id",
        "Table": "trashbin",
        "Operation": "Insert",
        "ItemHashKeyValue": "id",
        "IsPayloadJSON": "true"
    },
    "principalId": "ABCDEFG1234567ABCD890:outis",
    "details": "User: arn:aws:sts::123456789012:assumed-role/dynamo-testbin/5aUMInJH
 is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-
east-1:123456789012:table/testbin (Service: AmazonDynamoDBv2; Status Code: 400; Error Code:
 AccessDeniedException; Request ID: AKQJ987654321AKQJ123456789AKQJ987654321AKQJ987654321)"
}
```


Voici un exemple similaire de journal généré par la journalisation globale (p. 356) :


```
2017-12-09 22:49:17.954 TRACEID:ff562535-6964-506a-e141-78d40375fc4e
PRINCIPALID:ABCDEFG1234567ABCD890:outis [ERROR] EVENT:DynamoActionFailure
TOPICNAME:test-topic CLIENTID:iotconsole-123456789012-3
MESSAGE:Dynamo Insert record failed. The error received was User:
 arn:aws:sts::123456789012:assumed-role/dynamo-testbin/5aUMInJI is not authorized to
 perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:123456789012:table/
testbin
(Service: AmazonDynamoDBv2; Status Code: 400; Error Code: AccessDeniedException; Request
 ID: AKQJ987654321AKQJ987654321AKQJ987654321AKQJ987654321).
Message arrived on: test-topic, Action: dynamo, Table: trashbin, HashKeyField: id,
 HashKeyValue: id, RangeKeyField: None, RangeKeyValue: 123456789012
No newer events found at the moment. Retry.
```


Pour plus d'informations, consultez the section called “Affichage d'unAWS IoTLogs dans la console CloudWatch” (p. 373).

Diagnostic de services externes

Les services externes sont contrôlés par l'utilisateur final. Avant l'exécution d'une règle, vérifiez que les services externes que vous avez liés à votre règle sont configurés et disposent d'un débit et d'unités de capacité suffisants pour votre application.

Diagnostic de problèmes SQL

Si votre requête SQL ne renvoie pas les données que vous attendez :

- Consultez les journaux pour trouver des messages d'erreur.
- Vérifiez que votre syntaxe SQL correspond au document JSON du message.

  Examinez les noms d'objet et de propriété utilisés dans la requête avec ceux utilisés dans le document JSON de la charge utile de message de la rubrique. Pour plus d'informations sur la mise en forme JSON dans les requêtes SQL, consultezExtensions JSON (p. 540).

- Vérifiez si les noms d'objet ou de propriété JSON incluent des caractères réservés ou numériques.

  Pour plus d'informations sur les caractères réservés dans les références d'objet JSON dans les requêtes SQL, consultezExtensions JSON (p. 540).

Diagnostic des problèmes de shadows

Diagnostic de shadows

Problème	Consignes pour la résolution des problèmes
Un document shadow d'appareil est rejeté avec `Invalid JSON document`.	Si vous ne connaissez pas JSON, modifiez les exemples fournis dans ce manuel pour les adapter à votre utilisation. Pour plus d'informations, consultez Exemples de documents shadow (p. 585).
J'ai envoyé un code JSON correct, mais aucune ou seulement quelques parties sont stockées dans le document shadow d'appareil.	Vérifiez que vous avez suivi les consignes de formatage JSON. Seuls les champs JSON des sections `desired` et `reported` sont stockés. Le contenu JSON à l'extérieur de ces sections est ignoré (même s'il est formaté correctement).
J'ai reçu un message d'erreur indiquant que le shadow d'appareil dépasse la taille autorisée.	Le shadow d'appareil prend en charge seulement 8 Ko de données. Essayez de raccourcir les noms de champs au sein de votre document JSON ou créez simplement des shadows supplémentaires en créant plus d'objets. Un appareil peut avoir un nombre illimité d'objets/de shadows associés. La seule condition est que chaque nom d'objet soit unique dans votre compte.
Lorsque je reçois un shadow d'appareil, celui-ci fait plus de 8 Ko. Comment est-ce possible ?	À la réception, le service AWS IoT ajoute des métadonnées au shadow d'appareil. Le service inclut ces données dans sa réponse, mais elles ne comptent pas dans la limite de 8 Ko. Seules les données d'état `desired` et `reported` au sein du document d'état envoyé au shadow d'appareil comptent pour le calcule de la limite.
Ma demande a été rejetée car la version était incorrecte. Que dois-je faire ?	Exécutez une opération GET pour effectuer une synchronisation avec la dernière version du document d'état. Lors de l'utilisation de MQTT, abonnez-vous à la rubrique /update/accepted pour recevoir des notifications concernant les changements d'état et recevoir la dernière version du document JSON.
L'horodatage est décalé de quelques secondes.	L'horodatage des champs individuels et de l'ensemble du document JSON est mis à jour lorsque le document est reçu par le service AWS IoT ou lorsque le document d'état est publié dans les messages ./update/accepted et ./update/delta. Les messages peuvent être retardés sur le réseau, ce qui peut entraîner un décalage de l'horodatage de quelques secondes.

Problème	Consignes pour la résolution des problèmes
Mon appareil peut publier et s'abonner aux rubriques de shadow correspondantes, mais lorsque je tente de mettre à jour le document de shadow via l'API HTTP REST, un message HTTP 403 s'affiche.	Vérifiez que vous avez créé des stratégies dans IAM pour autoriser l'accès à ces rubriques et pour l'action (UPDATE/GET/DELETE) correspondant aux informations d'identification que vous utilisez. Les stratégies IAM et les stratégies de certificats sont indépendantes.
Autres problèmes.	Le service Device Shadow enregistre les erreurs dans CloudWatch Logs. Pour identifier les problèmes d'appareils et de configuration, activez les CloudWatch Logs et consultez les journaux d'informations de débogage.

Diagnostic des problèmes liés aux actions de flux d'entrée Salesforce IoT

Trace d'exécution

Comment consulter la trace d'exécution d'une action Salesforce ?

Consultez la section ContrôleAWS IoTUtilisation CloudWatch Logs (p. 373). Après avoir activé les journaux, vous pouvez consulter la trace d'exécution de l'action Salesforce.

Succès et échec d'une action

Comment vérifier que des messages ont été correctement envoyés à un flux d'entrée Salesforce IoT ?

Consultez les journaux générés par l'exécution de l'action Salesforce dans les CloudWatch Logs. Si vous pouvez lire `Action executed successfully`, cela signifie que le moteur de règles AWS IoT a reçu une confirmation de Salesforce IoT que le message a été correctement transmis au flux d'entrée ciblé.

Si vous rencontrez des problèmes avec la plateforme Salesforce IoT, consultez le support Salesforce IoT.

Que faire si des messages ne sont pas correctement envoyés à un flux d'entrée Salesforce IoT ?

Consultez les journaux générés par l'exécution de l'action Salesforce dans les CloudWatch Logs. Selon la nature de l'entrée du journal, vous pouvez tenter les opérations suivantes :

`Failed to locate the host`

Vérifiez que le paramètre `url` de l'action est correct et que le flux d'entrée Salesforce IoT Input existe bien.

`Received Internal Server Error from Salesforce`

Réessayer. Si le problème persiste, contactez le support Salesforce IoT.

`Received Bad Request Exception from Salesforce`

Vérifiez qu'il n'y a pas d'erreurs dans la charge utile que vous envoyez.

`Received Unsupported Media Type Exception from Salesforce`

Salesforce IoT ne prend pas en charge les charges utiles binaires pour le moment. Vérifiez que vous envoyez bien une charge utile JSON.

`Received Unauthorized Exception from Salesforce`

Vérifiez que le paramètre `token` de l'action est correct et que votre jeton est toujours valide.

`Received Not Found Exception from Salesforce`

Vérifiez que le paramètre `url` de l'action est correct et que le flux d'entrée Salesforce IoT Input existe bien.

Si vous rencontrez une erreur qui n'est pas répertoriée ici, contactezAWS IoTSupport

Dépannage des requêtes d'agrégation pour le service d'indexation de parc

Si vous rencontrez des erreurs d'incompatibilité de type, vous pouvez utiliser CloudWatch Logs pour résoudre le problème. CloudWatch Logs doivent être activés avant que les journaux ne soient écrits par le service d'indexation de la flotte. Pour plus d'informations, consultez ContrôleAWS IoTUtilisation CloudWatch Logs (p. 373).

Lorsque vous effectuez des requêtes d'agrégation sur des champs non gérés, vous pouvez uniquement spécifier un champ que vous avez défini dans l'argument `customFields` passé à `UpdateIndexingConfiguration` ou update-indexing-configuration. Si la valeur du champ n'est pas cohérente avec le type de données du champ configuré, cette valeur est ignorée lorsque vous effectuez une requête d'agrégation.

Le service d'indexation de flotte envoie un journal des erreurs dans les CloudWatch Logs lorsqu'un champ ne peut pas être indexé en raison d'un type non compatible. Le journal des erreurs contient le nom du champ, la valeur qui n'a pas pu être convertie et le nom d'objet de l'appareil. Voici un exemple de journal des erreurs.

```
{
  "timestamp": "2017-02-20 20:31:22.932",
  "logLevel": "ERROR",
  "traceId": "79738924-1025-3a00-a669-7bec69f7f07a",
  "accountId": "000000000000",
  "status": "SucceededWithIssues",
  "eventType": "IndexingCustomFieldFailed",
  "thingName": "thing0",
  "failedCustomFields": [
    {
      "Name": "attributeName1",
      "Value": "apple",
      "ExpectedType": "String"
    },
    {
      "Name": "attributeName2",
      "Value": "2",
      "ExpectedType": "Boolean"
    }
  ]
}
```

Si un appareil a été déconnecté pendant environ une heure, la valeur `timestamp` du statut de connectivité peut être manquante. En ce qui concerne les sessions permanentes, la valeur peut être manquante quand

un client a été déconnecté pendant une durée supérieure à la durée de vie (TTL) configurée pour la session permanente. Les données de statut de connectivité sont indexées uniquement pour les connexions où l'ID client contient un nom d'objet correspondant. (L'ID client est la valeur utilisée pour connecter un appareil à AWS IoT Core .)

Résolution des erreurs « Limite de flux dépassée pour votreAWS account"

Si vous voyez `"Error: You have exceeded the limit for the number of streams in your AWS account."`, vous pouvez nettoyer les flux inutilisés de votre compte au lieu de demander une augmentation de la limite.

Pour nettoyer un flux inutilisé que vous avez créé à l'aide de l'outilAWS CLIou SDK :

```
aws iot delete-stream -stream-id value
```

Pour plus de détails, consultez delete-stream.

> Note
>
> Vous pouvez utiliser la stratégie`list-streams`Pour rechercher les ID de flux.

Guide de dépannage AWS IoT Device Defender

General

Q : Existe-t-il des prérequis pour utiliserAWS IoT Device Defender?

A: Si vous souhaitez utiliser des métriques signalées par les appareils, vous devez d'abord déployer un agent sur votreAWS IoTpériphériques connectés ou passerelles de périphériques. Les appareils doivent fournir un identifiant client ou un nom d'objet cohérent.

Audit

Q : J'ai activé une vérification et mon audit indique « En cours » depuis un certain temps. Y a-t-il un problème ? Quand puis-je espérer des résultats ?

A: La collecte des données commence immédiatement après l'activation du contrôle. Toutefois, si votre compte doit collecter un volume important de données (certificats, objets, stratégies, etc.), les résultats du contrôle peuvent nécessiter un certain temps après que vous avez activé celui-ci.

Detect

Q : Comment puis-je connaître les seuils à définir dans unAWS IoT Device Defendercomportement du profil de sécurité ?

A: Commencez par créer comportement de profil de sécurité avec des seuils bas et attachez-le à un groupe d'objets contenant un ensemble représentatif d'appareils. Vous pouvez utiliser AWS IoT Device Defender pour afficher les métriques actuelles, puis affiner les seuils de comportement de l'appareil pour les adapter à votre cas d'utilisation.

Q : J'ai créé un comportement, mais il ne déclenche pas de violation quand je le souhaite. Comment dois-je résoudre le problème ?

A: Lorsque vous définissez un comportement, vous indiquez comment vous souhaitez que votre appareil se comporte normalement. Par exemple, si vous disposez d'une caméra de sécurité qui se connecte uniquement à un serveur central sur le port TCP 8888, vous ne vous attendez pas à qu'elle effectue d'autres connexions. Pour être alerté si la caméra se connecte sur un autre port, définissez un comportement tel que celui-ci :

```
{
  "name": "Listening TCP Ports",
  "metric": "aws:listening-tcp-ports",
  "criteria": {
    "comparisonOperator": "in-port-set",
    "value": {
      "ports": [ 8888 ]
    }
  }
}
```

Si la caméra effectue une connexion TCP sur le port TCP 443, le comportement de l'appareil est violé et une alerte est déclenchée.

Q : Un ou plusieurs de mes comportements sont en violation. Comment puis-je effacer la violation ?

A: Les alarmes s'effacent lorsque l'appareil revient au comportement souhaité, comme défini dans les profils de comportement. Les profils de comportement sont évalués à la réception des données de métriques pour votre appareil. Si l'appareil ne publie aucune métrique pendant plus de deux jours, l'événement de violation est défini sur `alarm-invalidated` automatiquement.

Q : J'ai supprimé un comportement qui était en violation, comment puis-je arrêter les alertes ?

A: La suppression d'un comportement arrête toutes les violations et les alertes futures pour ce comportement. Les alertes antérieures doivent être vidés à partir de votre mécanisme de notification. Lorsque vous supprimez un comportement, l'enregistrement des violations de celui-ci est conservé aussi longtemps que toutes les autres violations de votre compte.

Métriques d'appareil

Q : Je soumets des rapports de métriques qui enfreignent mes comportements, mais aucune violation n'a été déclenchée. Que se passe-t-il ?

A: Vérifiez que vos rapports de métriques sont acceptés en vous abonnant aux rubriques MQTT suivantes :

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

où `THING_NAME` est le nom de l'objet signalant la métrique, et `FORMAT` est « json » ou « cbor », selon le format du rapport de métriques envoyé par l'objet.

Une fois abonné, vous recevrez des messages sur ces rubriques pour chaque rapport de métriques envoyé. Un message `rejected` indique qu'un problème s'est produit lors de l'analyse du rapport de métriques. Un message d'erreur est inclus dans la charge utile du message pour vous aider à corriger les erreurs dans votre rapport de métriques. Un message `accepted` indique que le rapport de métriques a été analysé correctement.

Q : Que se passe-t-il si j'envoie une métrique vide dans mon rapport de métriques ?

A: Une liste vide de ports ou d'adresses IP est toujours considérée comme conforme au comportement correspondant. Si le comportement correspondant est en violation, la violation est effacée.

Q : Pourquoi mes rapports de métriques d'appareil contiennent-ils des messages pour des appareils qui ne sont pas dans la zoneAWS IoTregistre ?

Si vous avez un ou plusieurs profils de sécurité attachés à tous les objets ou à tous les objets non enregistrés, AWS IoT Device Defender inclut les métriques des objets non enregistrés. Si vous souhaitez exclure les métriques des objets non enregistrés, vous pouvez attacher les profils à tous les appareils enregistrés au lieu de tous les appareils.

Q : Je ne vois pas les messages d'un ou de plusieurs appareils non enregistrés même si j'ai appliqué un profil de sécurité à tous les appareils non enregistrés ou à tous les appareils. Comment résoudre ce problème ?

Vérifiez que vous envoyez un rapport de métriques bien formé à l'aide d'un des formats pris en charge. Pour plus d'informations, consultez Spécifications des métriques d'appareil (p. 951). Vérifiez que les appareils non enregistrés utilisent un identifiant de client ou un nom d'objet cohérent. Les messages notifiés par les appareils sont rejetés si le nom d'objet contient des caractères de contrôle ou est composé de plus de 128 octets de caractères codés UTF-8.

Q : Que se passe-t-il si un appareil non enregistré est ajouté au registre ou si un appareil enregistré devient non enregistré ?

A: Si un appareil est ajouté au registre ou en est supprimé :

- Vous voyez deux violations distinctes pour le périphérique (une sous son nom d'objet enregistré, une sous son identité non enregistrée) s'il continue à publier des métriques de violation. Les violations actives pour l'ancienne identité n'apparaissent plus après deux jours, mais sont disponibles dans l'historique des violations pendant 14 jours.

Q : Quelle valeur dois-je fournir dans le champ ID de rapport de mon rapport de métriques d'appareil ?

A: Utilisez une valeur unique pour chaque rapport de métriques, exprimée sous la forme d'un entier positif. Une pratique courante consiste à utiliser un horodatage epoch Unix.

Q : Dois-je créer une connexion MQTT dédiée pourAWS IoT Device Defendermétriques ?

A: Une connexion MQTT séparée n'est pas requise.

Q : Quel ID client dois-je utiliser lorsque je me connecte pour publier des métriques d'appareil ?

Pour les appareils (objets) qui se trouvent dans le registre AWS IoT, utilisez le nom d'objet enregistré. Pour les appareils qui ne se trouvent pas dans le registre AWS IoT, utilisez un identificateur cohérent lorsque vous vous connectez à AWS IoT. Cette pratique contribue à faire correspondre les violations et le nom d'objet.

Q : Puis-je publier des métriques pour un appareil avec un ID client différent ?

Il est possible de publier des métriques pour le compte d'un autre objet. Pour ce faire, vous devez publier les métriques dans la rubrique réservée AWS IoT Device Defender de cet appareil. Par exemple, `Thing-1` souhaite publier des métriques pour lui-même et pour le compte de `Thing-2`. `Thing-1` recueille ses propres métriques et les publie sur la rubrique MQTT :

```
$aws/things/Thing-1/defender/metrics/json
```

`Thing-1` obtient ensuite les métriques de la part de `Thing-2` et les publie sur la rubrique MQTT :

```
$aws/things/Thing-2/defender/metrics/json
```

Q : Combien de profils de sécurité et de comportements puis-je avoir dans mon compte ?

A: VoirAWS IoT Device DefenderPoints de terminaison et quotas.

Q : À quoi ressemble le prototype d'un rôle cible pour une cible d'alerte ?

A: Un rôle qui permetAWS IoT Device DefenderPour publier des alertes sur une cible d'alerte (rubrique SNS) exige deux objets :

- Une relation d'approbation spécifiant iot.amazonaws.com en tant qu'entité approuvée.
- Une stratégie attachée qui accorde à AWS IoT l'autorisation de publier dans une rubrique SNS spécifiée. Exemples :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "sns:Publish",
            "Resource": "<sns-topic-arn>"
        }
    ]
}
```

- Si la rubrique SNS utilisée pour la publication d'alertes est une rubrique chiffrée, avec l'autorisation de publier dans la rubrique SNS,AWS IoTdoit recevoir deux autorisations supplémentaires. Exemples :

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "sns:Publish",
                "kms:Decrypt",
                "kms:GenerateDataKey"
            ],
            "Resource": "<sns-topic-arn>"
       }
    ]
}
```

Device Advisor

General

Q : Puis-je exécuter plusieurs suites de tests en parallèle ?

A: Non. Actuellement Device Advisor ne prend pas en charge l'exécution de plusieurs suites de tests, car un seul point de terminaison Device Advisor est disponible par compte. Toutefois, vous pouvez tester plusieurs types de périphériques en séquençant les suites de test l'une après l'autre.

Q : J'ai vu sur mon appareil que la connexion TLS a été refusée par Device Advisor. C'est prévu ?

A: Oui. Device Advisor refuse la connexion TLS avant et après chaque exécution de test. Nous recommandons aux utilisateurs d'implémenter un mécanisme de nouvelle tentative de périphérique afin d'avoir une expérience de test entièrement automatisée avec Device Advisor. Si vous exécutez une suite de tests avec plus d'un cas de test, dites - TLS connect, MQTT connect et MQTT publier, alors nous vous recommandons d'avoir un mécanisme conçu pour votre appareil pour essayer de se connecter à notre point de fin de test toutes les 5 secondes pendant une minute à deux. Cela vous permettra d'exécuter plusieurs cas de test en séquence de manière automatisée.

Q : Puis-je obtenir un historique des appels d'API Device Advisor effectués sur mon compte à des fins d'analyse de sécurité et de résolution des problèmes opérationnels ?

A: Oui. Pour recevoir un historique des appels d'API Device Advisor effectués sur votre compte, il vous suffit d'activer CloudTrail dans la sectionAWS IoTManagement Console et filtrer la source de l'événement pour être`iotdeviceadvisor.amazonaws.com`.

Q : Comment afficher les journaux Device Advisor dans CloudWatch ?

Les journaux générés lors de l'exécution d'une suite de tests sont téléchargés dans CloudWatch si vous ajoutez la stratégie requise (par exemple,CloudWatchFullAccess) à votre rôle de service (voirConfiguration de (p. 998). Un groupe de journaux « AWS/IOT/DeviceAdvisor/$TestSuiteId » sera créé. Dans ce groupe de journaux, deux flux de journaux seront créés s'il y a au moins un cas de test dans votre suite de tests. L'un est nommé « $TestRunid » et inclut les journaux des actions effectuées avant et après l'exécution des cas de test dans votre suite de tests, telles que les étapes de configuration et de nettoyage. Un autre est « $suiteRunid_$TestRunid » qui est spécifique à une exécution de suite de tests. Événements envoyés depuis les appareils et AWS IoT Core sera enregistré dans ce flux de journal.

Q : Quel est le but du rôle d'autorisation de périphérique ?

A: Device Advisor se trouve entre votre appareil de test et AWS IoT Core pour simuler des scénarios de test. Il accepte les connexions et les messages de vos périphériques de test et les transmet à AWS IoT Core en assumant le rôle d'autorisation de votre appareil et en lançant une connexion en votre nom. Il est important de vous assurer que les autorisations de rôle de périphérique sont les mêmes que celles du certificat que vous utilisez pour exécuter des tests.AWS IoTne sont pas appliquées lorsque Device Advisor initie une connexion à AWS IoT Core en votre nom à l'aide du rôle d'autorisation de l'appareil. Toutefois, les autorisations du rôle d'autorisation de périphérique que vous définissez sont appliquées.

Q : Dans quelles régions Device Advisor est-il pris en charge ?

A: Device Advisor est pris en charge dans les régions us-east-1, us-west-2, ap-northeast-1 et eu-west-1.

Q : Que faire si je vois des résultats incohérents ?

A: L'une des principales causes de résultats incohérents est la définition de`EXECUTION_TIMEOUT`À une valeur trop faible. Pour plus d'informations sur les recommandations`EXECUTION_TIMEOUT`, consultezDevice Advisor.

Q : Quel protocole MQTT prend en charge Device Advisor ?

A: Device Advisor prend en charge MQTT avec les certificats clients X509.

Résolution des erreurs de déconnexions de la flotte de périphériques

AWS IoTLes déconnexions de la flotte de périphériques peuvent se produire pour plusieurs raisons. Cet article explique comment diagnostiquer une raison de déconnexion et comment gérer les déconnexions causées par la maintenance régulière deAWS IoTou une limite de limitation.

Pour diagnostiquer la raison de déconnexion

Vous pouvez consulter leAWSIOTLOGSV2log groupCloudWatchpour identifier la raison de la déconnexion dans la fenêtre`disconnectReason`de l'entrée de journal.

Vous pouvez également utiliserAWS IoT's Événements du cycle de viepour identifier la raison de la déconnexion. Si vous êtes abonné àévénement de déconnexion du cycle de vie(`$aws/events/presence/disconnected/`*`clientId`*), vous recevrez une notification deAWS IoTlorsque la déconnexion se produit. Vous pouvez identifier la raison de déconnexion dans`disconnectReason`de la notification.

Pour de plus amples informations, veuillez consulterCloudWatchAWS IoTentrées du journalandÉvénements du cycle de vie.

Pour résoudre les problèmes de déconnexion dus àAWS IoTmaintenance

Déconnexions causées parAWS IoTsont consignées en tant que`SERVER_INITIATED_DISCONNECT`inAWS IoTet CloudWatch. Pour gérer ces déconnexions, ajustez votre configuration côté client pour vous assurer que vos appareils peuvent être automatiquement reconnectés auAWS IoT.

Pour résoudre les problèmes de déconnexion dus à une limite de limitation

Les déconnexions causées par une limite de limitation sont enregistrées en tant que`THROTTLED`inAWS IoTet CloudWatch. Pour gérer ces déconnexions, vous pouvez demanderlimite du courtier de messages augmenteà mesure que le nombre d'appareils augmente.

Pour de plus amples informations, veuillez consulterAWS IoTAgent de messages.

Erreurs AWS IoT

Cette section présente les codes d'erreur envoyés par AWS IoT.

Codes d'erreur de l'agent de messages

Code d'erreur	Description de l'erreur.
400	Demande erronée.
401	Accès non autorisé.
403	Accès interdit.
503	Service non disponible.

Identités et des codes d'erreur de sécurité

Code d'erreur	Description de l'erreur.
401	Accès non autorisé.

Codes d'erreur Device Shadow

Code d'erreur	Description de l'erreur.
400	Demande erronée.
401	Accès non autorisé.
403	Accès interdit.
404	Introuvable.
409	Conflit.
413	Demande trop longue.
422	Impossible de traiter la demande.
429	Nombre de demandes trop élevé.

Code d'erreur	Description de l'erreur.
500	Erreur interne.
503	Service non disponible.

Quotas AWS IoT

Pour AWS IoT Core informations sur les quotas, voir AWS IoT Core Points de terminaison et quotasdans leAWSRéférence générale.

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.