
AWS IoT Core

Guide du développeur



AWS IoT Core: Guide du développeur

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Qu'est-ce que AWS IoT ?	1
Comment accéder à vos appareils et applicationsAWS IoT	1
QueAWS IoT puis-je faire	2
Lemoteur	2
L'IoT dans la domotique	3
Fonctionnement d'AWS IoT	3
IoT univers	3
AWS IoTaperçu des services	6
AWS IoT Coreservices	9
En savoir plus sur AWS IoT	12
Ressources de formation pourAWS IoT	12
AWS IoTressources et guides	12
AWS IoTsur les réseaux sociaux	13
AWSservices utilisés par le moteur deAWS IoT Core règles	13
Protocoles de communication pris en charge parAWS IoT Core	14
Nouveeeeeautés dans la nouvelleAWS IoT clé de	14
Légende	16
Démarrer avec AWS IoT Core	18
Connect votre premier appareil à AWS IoT Core	18
Configurez votre Compte AWS	19
S'inscrire à un Compte AWS	19
Création d'un utilisateur administratif	20
Ouvrez la AWS IoT console.	20
Essayez le didacticiel AWS IoT Core interactif	21
Connexion des appareils IoT	21
Enregistrer l'état de l'appareil hors ligne	22
Routage des données de l'appareil vers les services	23
Essayez la connexion AWS IoT rapide	23
Étape 1. Démarrez le didacticiel	24
Étape 2. Création d'un objet	25
Étape 3. Télécharger des fichiers sur votre appareil	28
Étape 4. Exécutez l'exemple	30
Étape 5. Explorez plus loin	34
Tester la connectivité avec le point de terminaison de données de votre appareil	34
Découvrez les AWS IoT Core services dans le cadre d'un didacticiel pratique	38
Quelle option d'appareil vous convient le mieux ?	39
Création de AWS IoT ressources	40
Configurer votre appareil	43
Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT	71
Affichage des messages MQTT dans le client MQTT	71
Publication de messages MQTT à partir du client MQTT	73
Tester les abonnements partagés dans le client MQTT	74
Connexion à AWS IoT Core	76
AWS IoT Core- points d'extrémité du plan de contrôle	76
AWS IoTpoints de terminaison de l'appareil	77
AWS IoT Corepour passerelles et appareils LoRa WAN	78
Connexion aux points de terminaison AWS IoT Core de service	79
AWS CLI pour AWS IoT Core	79
Kits de développement logiciel (SDK) AWS	80
Kits SDK AWS Mobile	84
API REST des AWS IoT Core services	84
Connexion d'appareils à AWS IoT	85
AWS IoTdonnées de l'appareil et points de terminaison de service	85
Kits SDK pour les appareils AWS IoT	87

Protocoles de communication des appareils	89
Rubriques MQTT	115
Points de terminaison configurables	131
Connexion aux points de AWS IoT terminaison FIPS	140
AWS IoT Core- points d'extrémité du plan de contrôle	140
AWS IoT Core- points de terminaison du plan de données	141
AWS IoT Device Management- points de terminaison des données sur les emplois	141
AWS IoT Device Management- Points de terminaison Fleet Hub	141
AWS IoT Device Management- points de terminaison de tunneling sécurisés	142
Didacticiels AWS IoT	143
Construire des démonstrations avec leAWS IoTClient d'appareil	143
Conditions préalables à la création de démonstrations avec leAWS IoTClient d'appareil	144
Préparation de vos appareils pour leAWS IoTClient d'appareil	146
Installation et configuration de l'AWS IoTClient d'appareil	156
Démontrer la communication des messages MQTT avec leAWS IoTAppareil client	165
Démontrez des actions à distance (tâches) avec AWS IoTPériphérique Périphérique	179
Nettoyage	189
Construire des solutions avec leAWS IoTKits SDK pour les appareils	196
Commencez à créer des solutions avec leAWS IoTKits SDK pour les appareils	196
Connexion d'un appareil à AWS IoT Core l'aide du SDK de l'AWS IoTAppareil	196
Création de AWS IoT règles pour acheminer les données de l'appareil vers d'autres services	213
Conservation de l'état de l'appareil lorsque l'appareil est hors connexion avec Device Shadows	244
Création d'un système d'autorisation personnalisé pour AWS IoT Core	265
Surveillance de l'humidité du sol avec un AWS IoT Raspberry Pi	277
Gestion des appareils avec AWS IoT	287
Comment gérer des objets avec le registre	287
Créer un objet	287
Liste des objets	288
Décrire les choses	289
Mettre à jour un objet	290
Supprimer un objet	290
Attacher un mandataire à un objet	290
Détacher un mandataire d'un objet	291
Types d'objets	291
Créer un type d'objet	291
Liste des types d'objets	292
Décrire un type d'objet	292
Associer un type d'objet à un objet	292
Rendre obsolète un type d'objet	293
Supprimer un type d'objet	294
Groupes d'objets statiques	294
Créer un groupe d'objets statiques	295
Décrire un groupe d'objets	296
Ajouter un objet à un groupe d'objets statiques	297
Supprimer un objet d'un groupe d'objet statiques	297
Répertorier les objets d'un groupe d'objets	297
Répertorier les groupes d'objets	298
Répertorier les groupes d'un objet	299
Mettre à jour un groupe d'objets statiques	300
Supprimer un groupe d'objets	300
Attacher une stratégie à un groupe d'objets statiques	301
Détacher une stratégie d'un groupe d'objets statiques	301
Répertorier les stratégies attachées à un groupe d'objets statiques	301
Répertorier les groupes d'une stratégie	302
Obtenir des stratégies efficaces pour un objet	302
Tester l'autorisation pour les actions MQTT	303
Groupes d'objets dynamiques	304

Créer un groupe d'objets dynamique	305
Décrire un groupe d'objets dynamique	305
Mettre à jour un groupe d'objets dynamique	306
Supprimer un groupe d'objets dynamique	307
Limitations et conflits	307
Balisage de vos ressources AWS IoT	310
Principes de base des balises	310
Limites et restrictions liées aux balises	311
Utilisation des balises avec des stratégies IAM	311
Groupes de facturation	313
Affichage des données de répartition des coûts et d'utilisation	314
Sécurité	315
Sécurité dans AWS IoT	315
Authentification	316
Formation et certification AWS	316
Présentation des certificats X.509	317
Authentification du serveur	317
Authentification client	320
Authentification et autorisation personnalisées	343
Autorisation	355
Formation et certification AWS	357
Stratégies AWS IoT Core	357
Autorisation d'appels directs vers des AWS services à l'aide du fournisseur AWS IoT Core d'informations d'identification	402
Accès multicomppte avec IAM	407
Protection des données	408
Chiffrement des données dans AWS IoT	409
Sécurité du transport dans AWS IoT Core	409
Chiffrement des données	413
Gestion des identités et des accès	414
Public ciblé	414
Authentification avec des identités IAM	414
Gestion des accès à l'aide de politiques	417
Fonctionnement de AWS IoT avec IAM	419
Exemples de politiques basées sur l'identité	438
Politiques gérées par AWS	441
Résolution des problèmes	451
Journalisation et surveillance	453
Outils de supervision	453
Validation de la conformité	454
Résilience	455
Utilisation d'AWS IoT Core avec des points de terminaison d'un VPC	456
Création des points de terminaison d'un VPC pour AWS IoT Core	456
Contrôle de l'accès aux points de AWS IoT Core terminaison de plusieurs VPC	457
Limites des points de terminaison d'un VPC	458
Mise à l'échelle des points de terminaison VPC avec IoT Core	458
Utilisation de domaines personnalisés avec des points de terminaison VPC	458
Disponibilité des points de terminaison VPC pour AWS IoT Core	458
Sécurité de l'infrastructure	458
Surveillance de la sécurité	459
Bonnes pratiques de sécurité	459
Protection des connexions MQTT dans AWS IoT	459
Veiller à la synchronisation de l'horloge de votre appareil	461
Valider le certificat de serveur	462
Utiliser une identité unique par appareil	462
Utilisez-en un second Région AWS comme sauvegarde	462
Utiliser la mise en service juste à temps	463

Autorisations pour exécuter des tests AWS IoT Device Advisor	463
Prévention confus entre services pour Device Advisor	464
Formation et certification AWS	465
Surveillance des AWS IoT	466
Configurer la journalisation AWS IoT	467
Configurer le rôle et la stratégie de journalisation	467
Configurez la journalisation par défaut dans AWS IoT (console)	469
Configurer la connexion par défaut dans AWS IoT (interface de ligne de commande)	470
Configurer la connexion spécifique aux ressources AWS IoT (interface de ligne de commande)	472
Niveaux de journalisation	474
Surveillez les AWS IoT alarmes et les métriques à l'aide d'Amazon CloudWatch	474
Utilisation de métriques AWS IoT	474
Création d'alarmes CloudWatch dans AWS IoT	475
Métriques et dimensions AWS IoT	478
Surveiller AWS IoT à l'aide CloudWatch des journaux	490
Affichage des journaux AWS IoT dans la console CloudWatch	490
CloudWatch AWS IoT entrées du journal	491
Importer les journaux côté appareil sur Amazon CloudWatch	512
Comment ça marche	513
Chargement des journaux côté appareil à l'aide de règles AWS IoT	513
Journalisation des appels d'API AWS IoT avec AWS CloudTrail	521
Informations AWS IoT dans CloudTrail	521
Présentation des AWS IoT entrées des fichiers journaux	522
Règles	524
Accorder à une AWS IoT règle l'accès dont elle a besoin	525
Transmettre les autorisations de rôle	526
Création d'une règle AWS IoT	527
Affichage des règles	531
Suppression d'une règle	531
Actions de règle AWS IoT	531
Apache Kafka	533
Alarmes CloudWatch	541
Journaux CloudWatch	542
Métriques CloudWatch	544
DynamoDB	546
DynamoDBv2	548
Elasticsearch	549
HTTP	551
IoT Analytics	578
AWS IoT Events	580
AWS IoT SiteWise	582
Kinesis Data Firehose	586
Kinesis Data Streams	587
Lambda	589
Emplacement	591
OpenSearch	594
Republish	595
S3	597
Salesforce IoT	599
SNS	600
SQS	601
Step Functions	603
Timestream	604
Résolution des problèmes d'une règle	609
Accès aux ressources multicomptes à l'aide AWS IoT de règles	609
Prérequis	610
Configuration multicomppte pour Amazon SQS	610

Configuration intercompte pour Amazon SNS	611
Configuration intercompte pour Amazon S3	612
Configuration intercompte pour AWS Lambda	614
Gestion des erreurs (action d'erreur)	615
Format du message d'action d'erreur	616
Exemple d'action d'erreur	617
Réduire les coûts de messagerie grâce à Basic Ingest	617
Utilisation de Basic Ingest	618
Référence SQL AWS IoT	618
Clause SELECT	619
Clause FROM	621
Clause WHERE	622
Types de données	622
Opérateurs	626
Fonctions	632
Littéraux	678
Instructions Case	679
Extensions JSON	680
Modèles de substitution	681
Requêtes d'objets imbriqués	683
Charges utiles binaires	684
Versions de SQL	688
Service Device Shadow	690
Utilisation des shadows	690
Choix d'utilisation de shadows nommés ou non nommés	690
Accès aux shadows	691
Utilisation des shadows sur les appareils, dans les applications et dans d'autres services cloud	691
Ordre des messages	692
Suppression des messages de shadow	693
Utilisation des shadows sur les appareils	694
Initialisation de l'appareil lors de la première connexion à AWS IoT	695
Traitement des messages lorsque l'appareil est connecté à AWS IoT	696
Traitement des messages lorsque l'appareil se reconnecte à AWS IoT	697
Utilisation des shadows dans les applications et les services	697
Initialisation de l'application ou du service lors de la connexion à AWS IoT	698
Traitement des changements d'état lorsque l'application ou le service sont connectés à AWS IoT ..	698
Détection d'un appareil connecté	698
Simulation des communications du service Device Shadow	700
Configuration de la simulation	700
Initialisation de l'appareil	700
Envoi d'une mise à jour à partir de l'application	703
Réponse à une mise à jour sur l'appareil	705
Observation de la mise à jour dans l'application	708
Au-delà de la simulation	709
Interaction avec les shadows	709
Support du protocole	710
Demande d'état et génération de rapport d'état	710
Mise à jour d'un shadow	710
Récupération d'un document Shadow	714
Suppression de données shadow	714
API REST Device Shadow	716
GetThingShadow	717
UpdateThingShadow	718
DeleteThingShadow	719
ListNamedShadowsForThing	720
Rubriques MQTT de Device Shadow	721
/get	722

/get/accepted	722
/get/rejected	723
/update	723
/update/delta	724
/update/accepted	725
/update/documents	726
/update/rejected	726
/delete	727
/delete/accepted	728
/delete/rejected	728
Documents du service Device Shadow	729
Exemples de documents shadow	729
Propriétés du document	734
État Delta	734
Documents shadow de gestion des versions	736
Jetons clients dans les documents shadow	736
Propriétés de document shadow vides	736
Valeurs de tableau dans les documents shadow	737
Messages d'erreur de Device Shadow	738
Tâches	739
Accès aux AWS IoT offres d'emploi	739
AWS IoT Emplois Régions et points de terminaison	739
Qu'est-ce que AWS IoT Jobs ?	739
Concepts clés relatifs aux tâches	740
Tâches et états d'exécution des tâches	743
Gestion des tâches	747
Signature de code pour les tâches	747
Document de Job	747
URL présignées	747
Création et gestion de tâches à l'aide de la console	749
Créez et gérez des tâches à l'aide de l'interface de ligne de commande	751
Modèles de tâche	759
Modèles personnalisés et AWS gérés	759
Utiliser des modèles AWS gérés	759
Créez des modèles de travail personnalisés	773
Configurations de Job	778
Comment fonctionnent les configurations de tâches	779
Spécifiez des configurations supplémentaires	789
Appareils et tâches	795
Programmation des appareils pour une utilisation avec Jobs	797
Flux de travail des appareils	797
Flux de travail	799
Notifications Jobs	802
AWS IoT Tâches, opérations d'API	808
API de gestion et de contrôle des tâches et types de données	809
Tâches, opérations de l'appareil, API MQTT et HTTPS et types de données	823
Sécurisation des utilisateurs et des appareils pour Jobs	833
Type de politique requis pour AWS IoT Jobs	833
Autorisation des utilisateurs de Jobs et des services cloud	834
Autoriser les appareils à utiliser des tâches	842
Limites des tâches	845
Limites de tâches actives et simultanées	845
Tunneling sécurisé AWS IoT	848
Qu'est-ce que le tunneling sécurisé ?	848
Concepts de tunneling sécurisés	848
Comment fonctionne le tunneling sécurisé	849
Cycle de vie des tunnels sécurisés	850

AWS IoT tutoriels de tunneling sécurisé	850
Didacticiels dans cette section	851
Ouvrez un tunnel et démarrez une session SSH vers un appareil distant	851
Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur	865
Proxy local	868
Comment utiliser le proxy local	868
Configuration du proxy local pour les appareils utilisant un proxy Web	872
Multiplexage et connexions TCP simultanées	878
Multiplexage de plusieurs flux de données	878
Utilisation de connexions TCP simultanées	881
Configuration d'un appareil distant et utilisation d'un agent IoT	883
Extrait de l'agent IoT	883
Contrôle de l'accès aux tunnels	885
Conditions préalables à l'accès au tunnel	885
stratégies d'accès au tunnel	885
Résolution des problèmes de connectivité par tunneling sécurisé	890
Erreur de jeton d'accès au client non valide	890
Erreur de non-correspondance du jeton client	891
Problèmes de connectivité de l'appareil à distance	892
Mise en service des appareils	894
Provisionner des appareils dans AWS IoT	895
API de mise en service de flotte	895
Mise en service d'appareils qui ne disposent pas de certificats d'appareils à l'aide de la mise en service de flotte	896
Allocation par revendication	896
Allocation par utilisateur approuvé	898
Utilisation des hooks de pré-provisionnement avec l'interface de ligne de commande AWS	900
Mise en service d'appareils disposant de certificats d'appareils	902
Mise en service d'un seul objet	903
Just-in-timeApprovisionnement en J	903
Enregistrement en bloc	907
Mise en service des modèles	908
Section Parameters	908
Section Resources	909
Exemple de modèle pour l'enregistrement groupé	912
Exemple de modèle pour le just-in-time provisionnement (JITP)	913
Mise en service de flotte	915
Hooks de mise en service en amont	917
Entrée du hook de pré-provisionnement	918
Valeur de retour du hook de pré-provisionnement	918
Exemple de crochet de pré-provisionnement Lambda	918
Création de politiques et de rôles IAM pour un utilisateur installant un appareil	920
Création d'une politique IAM pour l'utilisateur qui installera un appareil	920
Création d'un rôle IAM pour l'utilisateur qui installera un appareil	921
Mettre à jour une politique existante pour autoriser un nouveau modèle	922
API MQTT de mise en service des appareils	923
CreateCertificateFromCsr	923
CreateKeysAndCertificate	925
RegisterThing	926
Indexation de la flotte	929
Gestion de l'indexation de la flotte	930
Indexation des objets	930
Indexation de groupes d'objets	931
Champs gérés	931
Champs personnalisés	932
Gérer l'indexation des objets	933
Gérer l'indexation des groupes d'objets	943

Interrogation des données agrégées	945
GetStatistics	945
GetCardinality	947
GetPercentiles	948
GetBucketsAggregation	949
Autorisation	950
Syntaxe de requête	950
Fonctionnalités prises en charge	950
Fonctions non prises en charge	951
Remarques	951
Exemples de requêtes sur des objets	951
Exemples de requêtes sur des groupes d'objets	954
Métriques du parc	955
Didacticiel de démarrage	956
Gestion des métriques de flotte	961
Livraison de fichiers basée sur MQTT	965
Qu'est-ce qu'un flux ?	965
Gérer un flux dans leAWS cloud	966
Accordez des autorisations à vos appareils	966
Connect vos appareils àAWS IoT	967
Utilisation deAWS IoT la livraison de fichiers basée sur MQTT sur les appareils	967
DescribeStream À utiliser pour obtenir des données de flux	967
Obtenir des blocs de données à partir d'un fichier de flux	969
Gestion des erreurs liées à la livraison de fichiersAWS IoT basée sur MQTT	973
Exemple de cas d'utilisation dans FreeRTOS OTA	974
AWS IoT Device Defender	975
Formation et certification AWS	975
Démarrer avec AWS IoT Device Defender	975
Configuration	975
Guide d'audit	976
Guide de ML	989
Personnalisez quand et comment vous affichezAWS IoT Device DefenderRésultats de l'audit	1012
Audit	1023
Gravité du problème	1024
Étapes suivantes	1024
Contrôles d'audit	1024
Commandes d'audit	1055
Vérification de la recherche de suppressions	1081
Détection	1090
Surveillance du comportement des appareils non enregistrés	1092
Cas d'utilisation des	1092
Concepts	1097
Behaviors	1098
Détectez ML	1100
Métriques personnalisées	1105
Métriques côté appareil	1111
Mesures côté cloud	1126
Définition de la portée des métriques dans les profils de sécurité à l'aide de dimensions	1133
Autorisations	1140
Commandes Detect	1141
Utilisation d'AWS IoT Device Defender Detect	1142
Actions d'atténuation	1144
Les mesures d'désynchronisation	1144
Détecter les désynchronisation	1147
Comment définir et gérer des actions d'atténuation	1147
Appliquer des actions d'atténuation	1149
Autorisations	1153

Commandes d'action d'atténuation	1156
Utilisation d'AWS IoT Device Defender avec d'autres services AWS	1157
Utilisation AWS IoT Device Defender avec des appareils en cours d'exécution AWS IoT Greengrass	1157
Utilisation AWS IoT Device Defender avec FreeRTOS et appareils intégrés	1157
Utilisation de AWS IoT Device Defender avec AWS IoT Device Management	1158
Intégration avec Security Hub	1158
Prévention du député confus entre services	1164
Bonnes pratiques de sécurité pour les agents d'appareil	1165
Device Advisor	1167
Configuration	1168
Créez un objet IoT	1168
Créez un rôle IAM à utiliser comme rôle de votre appareil	1168
Création d'une politique gérée personnalisée permettant à un utilisateur IAM d'utiliser Device Advisor	1170
Création d'un utilisateur IAM pour utiliser Device Advisor	1170
Configurer votre appareil	1172
Commencer à utiliser Device Advisor dans la console	1173
Flux de Device Advisor	1180
Prérequis	1180
Création d'une définition de suite de test	1180
Obtenir une définition de la suite de tests	1182
Obtenir un point de terminaison de test	1182
Lancer l'exécution d'une suite de tests	1182
Lancer une suite de tests	1183
Arrêter l'exécution d'une suite de tests	1183
Obtenez un rapport de qualification pour une exécution réussie de la suite de tests de qualification	1184
Flux de travail détaillé de Device Advisor sur console	1184
Prérequis	1184
Création d'une définition de suite de test	1185
Lancer l'exécution d'une suite de tests	1190
Arrêter l'exécution d'une suite de tests (facultatif)	1191
Afficher les détails et les journaux d'exécution de la suite de tests	1192
Télécharger un rapport AWS IoT de qualification	1193
Flux de travail de console de tests de longue durée	1194
Points de terminaison VPC Device Advisor () AWS PrivateLink	1200
Considérations relatives aux points de terminaison de VPC AWS IoT Core Device Advisor	1201
Créer un point de terminaison de VPC d'interface pour AWS IoT Core Device Advisor	1202
Contrôle de l'accès aux points de AWS IoT Core Device Advisor terminaison d'un VPC	1202
Scénarios de test de Device Advisor	1203
Device Advisor teste des scénarios afin de les qualifier pour le programme de qualification des AWS appareils	1203
TLS	1204
MQTT	1208
Shadow	1217
Exécution de Job	1219
Autorisations et politiques	1220
Tests de longue durée	1221
AWS IoT CoreEmplacement de l'appareil	1234
Types de mesures et solveurs	1234
Comment fonctionne la localisation de l'AWS IoT Coreappareil	1235
Comment utiliser la localisation de AWS IoT Core l'appareil	1236
Résolution de la localisation des appareils IoT	1237
Résolution de la localisation de l'appareil (console)	1237
Résolution de la localisation de l'appareil (API)	1239
Résolution des erreurs lors de la résolution de l'emplacement	1240

Résolution de la localisation des appareils à l'aide des rubriques MQTT	1241
Format de localisation de l'appareil (rubriques MQTT)	1241
Politique relative à la localisation des appareils (sujets MQTT)	1242
Rubriques relatives à la localisation des appareils et charge utile	1243
Solveurs de localisation et charge utile de l'appareil	1246
Solveur basé sur Wi-Fi	1247
Solveur basé sur la technologie cellulaire	1247
Solveur de recherche inversée IP	1251
solveur GNSS	1251
Messages d'événements	1253
Comment les messages d'événements sont générés	1253
Politique de réception des messages relatifs aux événements	1253
Activer les événements pour AWS IoT	1253
Événements de registre	1257
Évènements d'objets	1257
Événements de type objet	1258
Événements liés à des groupes d'objets	1260
Événements Jobs	1264
Événements du cycle de vie	1268
Événements de connexion/déconnexion	1268
Événements d'abonnement/désabonnement	1270
AWS IoT Corepour LoRa WAN	1272
Introduction	1272
Comment utiliser AWS IoT Core pour le LoRa WAN	1272
AWS IoT Corepour les régions LoRa WAN et les points de terminaison	1273
AWS IoT Corepour la tarification du LoRa WAN	1273
Qu'est-ce que AWS IoT Core le LoRa WAN ?	1273
Qu'est-ce que le LoRa WAN ?	1274
Comment AWS IoT Core fonctionne le LoRa WAN	1275
Connexion de passerelles et d'appareils à AWS IoT Core for LoRaWAN	1276
Conventions de dénomination pour vos appareils, passerelles, profils et destinations	1276
Mappage des données de l'appareil vers les données de service	1277
Utilisation de la console pour intégrer votre appareil et votre passerelle vers AWS IoT Core for LoRaWAN	1277
Décrivez vos ressources AWS IoT Core pour le LoRa WAN	1278
Intégrez vos passerelles vers AWS IoT Core for LoRaWAN	1279
Intégrez vos appareils à AWS IoT Core for LoRaWAN	1288
Configurer les informations de position pour les périphériques et les passerelles LoRa WAN	1300
Comment fonctionne le positionnement pour les appareils LoRa WAN	1300
Présentation des flux de travail de positionnement	1301
Configuration de votre position de ressource	1302
Configuration de la position des passerelles LoRa WAN	1302
Configuration de la position des périphériques LoRa WAN	1304
Connexion AWS IoT Core for LoRaWAN via un point de terminaison d'interface VPC	1308
Point de terminaison d'API du plan de AWS IoT Core for LoRaWAN contrôle intégré	1310
Points de terminaison de l'API du plan de AWS IoT Core for LoRaWAN données	1313
Gestion des passerelles avec AWS IoT Core for LoRa WAN	1319
LoRa Configuration logicielle requise pour Basics Station	1319
Utilisation de passerelles qualifiées figurant dans le catalogue d'appareils AWS partenaires	1319
Utilisation des protocoles CUPS et LNS	1320
Configurez les fonctionnalités de balisage et de filtrage de vos passerelles LoRa WAN	1320
Mettez à jour le firmware de la passerelle à l'aide du service CUPS AWS IoT Core for LoRaWAN ..	1324
Choix des passerelles pour recevoir le trafic de données de liaison descendante du LoRa WAN ..	1335
Gestion des appareils avec AWS IoT Core for LoRaWAN	1337
Considérations sur les appareils	1337
Utilisation d'appareils dotés de passerelles qualifiées pour AWS IoT Core for LoRaWAN	1337
LoRaVersion WAN	1338

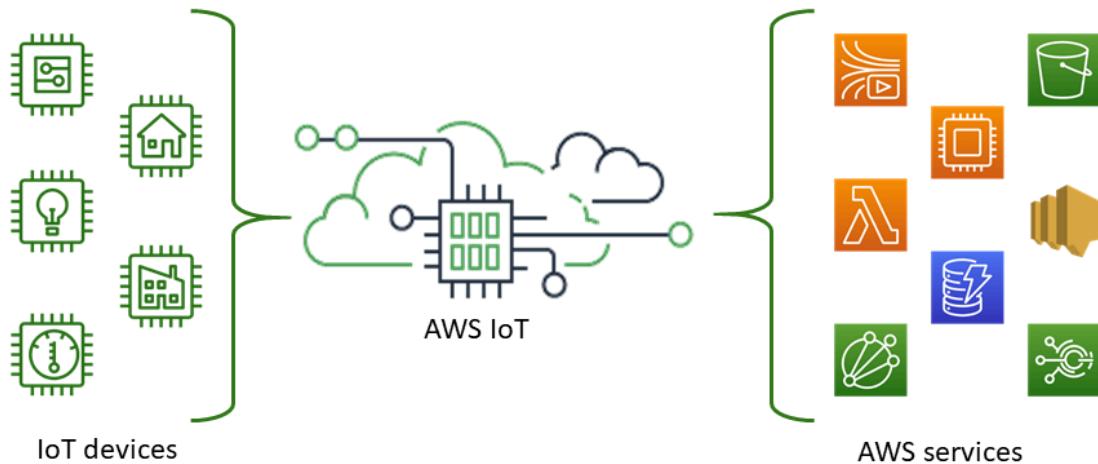
Modes d'activation	1338
Classes d'appareil	1338
Gérez la communication entre vos appareils LoRa WAN et AWS IoT	1338
Gérez le trafic d'itinérance provenant d'appareils LoRa WAN situés hors du réseau domestique ..	1345
Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils	1352
Mises à jour du micrologiciel en direct (FUOTA) pour les AWS IoT Core for LoRaWAN appareils ..	1363
Surveillance de votre parc de ressources sans fil en temps réel à l'aide d'un analyseur de réseau	1374
Ajouter le rôle IAM nécessaire pour l'analyseur de réseau	1376
Créer une configuration d'analyseur de réseau et ajouter des ressources	1377
Diffusez les messages de suivi de l'analyseur de réseau avec WebSockets	1384
Afficher et surveiller les journaux des messages de suivi de l'analyseur de réseau en temps réel ..	1393
Déboguez vos groupes de multidiffusion et vos tâches FUOTA à l'aide d'un analyseur de réseau ..	1396
Sécurité des données avec AWS IoT Core for LoRa WAN	1398
Comment les données sont sécurisées dans l'ensemble du système	1398
LoRaSécurité du transport des périphériques et des passerelles WAN	1399
AWS IoT Core pour Amazon Sidewalk	1400
Comment l'utiliser AWS IoT Core pour Amazon Sidewalk ?	1400
AWS IoT Core pour Amazon SidewalkRégions et points de terminaison	1400
Tarification d'AWS IoT Core pour Amazon Sidewalk	1401
Qu'est-ce que AWS IoT Core pour Amazon Sidewalk ?	1401
Fonctionnalités d'AWS IoT Core pour Amazon Sidewalk	1401
Qu'est-ce qu'Amazon Sidewalk ?	1401
Fonctionnement d'AWS IoT Core pour Amazon Sidewalk	1402
Démarrer avec AWS IoT Core pour Amazon Sidewalk	1404
Essayez le didacticiel de surveillance des capteurs	1404
Configuration	1405
Décrire vos ressources Sidewalk	1407
Présentation de l'intégration de vos appareils Sidewalk	1408
Connexion d'appareils Sidewalk à AWS IoT Core pour Amazon Sidewalk	1411
Prérequis	1411
Décrire vos ressources Sidewalk	1411
Ajoutez votre appareil Sidewalk	1412
Ajouter une destination pour l'appareil Sidewalk	1419
Connect votre appareil Sidewalk	1424
Approvisionnement en masse d'appareils avec AWS IoT Core pour Amazon Sidewalk	1426
Flux de travail de provisionnement en masse d'Amazon Sidewalk	1426
Création de profils d'appareils avec prise en charge en usine	1429
Provisionnement d'appareils Sidewalk à l'aide de tâches d'importation	1432
AWS IoT Core pour Amazon SidewalkOpérations d'API	1441
Opérations d'API pour les profils d'appareils Sidewalk	1441
Opérations d'API pour les terminaux Sidewalk	1442
Opérations d'API pour les destinations pour les appareils Sidewalk	1444
Opérations d'API pour le provisionnement en masse	1446
Surveillance et événements pour AWS IoT Core pour Amazon Sidewalk	1449
Événements pour les appareils Sidewalk	1449
Surveillance des dispositifs de trottoir	1450
Surveillance et journalisation pour AWS IoT Wireless l'utilisation d'Amazon CloudWatch	1452
Configurer la journalisation pour AWS IoT Wireless	1453
Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless	1453
Configurer la journalisation des AWS IoT Wireless ressources	1455
Surveiller AWS IoT Wireless à l'aide de CloudWatch journaux	1463
Afficher les entrées du CloudWatch AWS IoT Wireless journal	1464
Utilisez CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless	1470
Notifications d'événements pour AWS IoT Wireless	1474
Comment vos ressources peuvent être informées des événements	1474
Événements et types de ressources	1474

Politique de réception de notifications d'événements sans fil	1475
Format des rubriques MQTT pour les événements sans fil	1475
Tarification des événements sans fil	1478
Activer les événements pour les ressources sans fil	1478
Configurations d'événements	1478
Prérequis	1478
Activez les notifications à l'aide du AWS Management Console	1478
Activez les notifications à l'aide du AWS CLI	1480
Notifications d'événements pour les ressources LoRa WAN	1481
Types d'événements pour les ressources LoRa WAN	1481
LoRaParticiper à des événements WAN	1482
Événements d'état de connexion	1484
Notifications d'événements pour les ressources Sidewalk	1486
Types d'événements pour les ressources Sidewalk	1486
Événements relatifs à l'état d'enregistrement des appareils	1486
Événements de proximité	1489
Événements d'état de distribution de message	1491
AWS IoTSDK pour appareils, kits SDK mobiles et client deAWS IoT l'appareil	1494
Kits SDK pour les appareils AWS IoT	1494
Kit SDK des appareils AWS IoT pour Embedded C	1495
Versions antérieures des SDK pourAWS IoT appareils	1496
Kits SDK AWS Mobile	1496
AWS IoTAppareil client	1497
Résolution des problèmes	1498
Diagnostic des problèmes de connectivité	1498
Connexion	1498
Authentification	1499
Autorisation	1500
Sécurité et identité	1500
Diagnostic des problèmes de règles	1501
Configuration CloudWatch Journaux pour dépannage	1501
Diagnostic de services externes	1502
Diagnostic de problèmes SQL	1502
Diagnostic des problèmes de shadows	1502
Diagnostic des problèmes liés aux actions Salesforce	1503
Trace d'exécution	1503
Succès et échec d'une action	1504
Guide de dépannage de flotte	1504
Dépannage des requêtes d'agrégation pour le service d'indexation de parc	1504
Résolution des métriques de flotte	1505
Dépannage « Limite de flux dépassée pourAWScompte »	1506
Guide de dépannage AWS IoT Device Defender	1506
AWS IoTGuide de dépannage Device Advisor	1510
Résolution des déconnexions de flotte d'appareils	1512
Erreurs AWS IoT	1512
AWS IoTQuotas	1514
Tarification d'AWS IoT Core	1515

Qu'est-ce que AWS IoT ?

AWS IoT fournit les services cloud qui connectent vos appareils IoT à d'autres appareils et aux services cloud AWS. AWS IoT fournit des logiciels pour appareils qui peuvent vous aider à intégrer vos appareils IoT dans des solutions basées sur AWS IoT. Si vos appareils ont accès à AWS IoT, AWS IoT peut les connecter aux services cloud fournis par AWS.

Pour une introduction pratique à AWS IoT, rendez-vous sur [Démarrer avec AWS IoT Core \(p. 18\)](#).



AWS IoT vous permet de sélectionner les up-to-date technologies et les technologies les plus adaptées à votre solution. Pour vous aider à gérer et à prendre en charge vos appareils IoT sur le terrain, AWS IoT Core prend en charge les protocoles suivants :

- [MQTT \(mise en file d'attente des messages et transport de télémétrie\) \(p. 92\)](#)
- [MQTT sur WSS \(Websockets Secure\) \(p. 92\)](#)
- [HTTPS \(protocole de transfert hypertexte - sécurisé\) \(p. 112\)](#)
- [LoRaWAN \(réseau étendu à longue portée\) \(p. 1272\)](#)

Le courtier de AWS IoT Core messages prend en charge les appareils et les clients qui utilisent les protocoles MQTT et MQTT sur WSS pour publier des messages et s'y abonner. Il prend également en charge les appareils et les clients qui utilisent le protocole HTTPS pour publier des messages.

AWS IoT Core for LoRa WAN vous permet de connecter et de gérer des appareils LoRa WAN sans fil (réseau étendu longue portée à faible consommation). AWS IoT Core for LoRa WAN vous évite de devoir développer et exploiter un serveur réseau LoRa WAN (LNS).

Si vous n'avez pas besoin de AWS IoT fonctionnalités telles que les communications entre appareils, [les règles \(p. 524\)](#) ou les [tâches \(p. 739\)](#), consultez la section [AWS Messagerie](#) pour plus d'informations sur les autres services de AWS IoT messagerie susceptibles de mieux répondre à vos besoins.

Comment accéder à vos appareils et applications AWS IoT

AWS IoT fournit les interfaces suivantes pour [Didacticiels AWS IoT \(p. 143\)](#) :

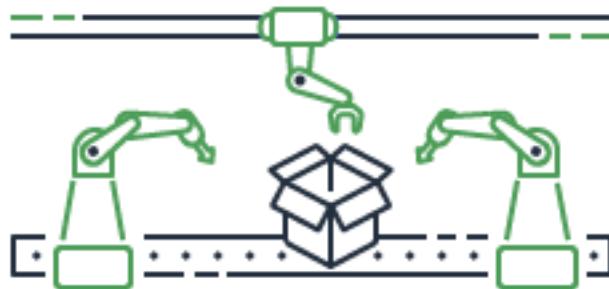
- AWS IoT SDK pour appareils : créez sur vos appareils des applications qui envoient des messages et en reçoivent des messages AWS IoT. Pour plus d'informations, veuillez consulter [AWS IoT SDK pour appareils, kits SDK mobiles et client de AWS IoT l'appareil \(p. 1494\)](#).
- AWS IoT Core pour le LoRa WAN : Connectez et gérez vos périphériques et passerelles LoRa WAN longue portée (WAN) à l'aide de [AWS IoT Core for LoRa WAN \(p. 1272\)](#).
- AWS Command Line Interface (AWS CLI) – Exécutez des commandes pour AWS IoT sur Windows, macOS et Linux. Ces commandes vous permettent de créer et de gérer des objets, des certificats, des règles, des tâches et des politiques. Consultez le [AWS Command Line Interface Guide de l'utilisateur](#) pour démarrer. Pour plus d'informations sur les commandes de AWS IoT, consultez [iot](#) dans la référence des AWS CLI commandes.
- API AWS IoT – Créez vos applications IoT à l'aide de requêtes HTTP ou HTTPS. Ces actions d'API vous permettent de créer et de gérer par programmation des objets, des certificats, des règles et des politiques. Pour plus d'informations sur les actions d'API pour AWS IoT, consultez [Actions](#) dans la AWS IoT Référence d'API.
- AWSSDK : créez vos applications IoT à l'aide d'API spécifiques à chaque langue. 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 y accéder AWS IoT via la [AWS IoT console](#), qui fournit une interface utilisateur graphique (GUI) grâce à laquelle vous pouvez configurer et gérer les objets, les certificats, les règles, les tâches, les politiques et les autres éléments de vos solutions IoT.

QueAWS IoT puis-je faire

Cette rubrique décrit certaines des solutions dont vous pourriez avoir besoin pour prendre AWS IoT en charge

Lemoteur



Voici quelques exemples de AWS IoT solutions pour des [cas d'utilisation industriels](#) qui appliquent les technologies IoT pour améliorer les performances et la productivité des processus industriels.

Solutions pour les cas d'utilisation industriels

- [AWS IoT à utiliser pour créer des modèles de qualité prédictifs dans les opérations industrielles](#)

Découvrez AWS IoT comment collecter et analyser les données des opérations industrielles pour créer des modèles de qualité prédictifs. [En savoir plus](#)

- [AWS IoT à utiliser pour soutenir la maintenance prédictive dans les opérations industrielles](#)

Découvrez AWS IoT comment planifier la maintenance préventive afin de réduire les temps d'arrêt imprévus. [En savoir plus](#)

L'IoT dans la domotique



Voici quelques exemples de AWS IoT solutions pour des [cas d'utilisation de la domotique](#) qui 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

- [AWS IoT à utiliser dans votre maison connectée](#)

Découvrez comment AWS IoT vous pouvez fournir des solutions domotiques intégrées.

- [AWS IoT à utiliser pour assurer la sécurité et la surveillance de la maison](#)

Découvrez AWS IoT comment appliquer l'apprentissage automatique et l'informatique de pointe à votre solution domotique.

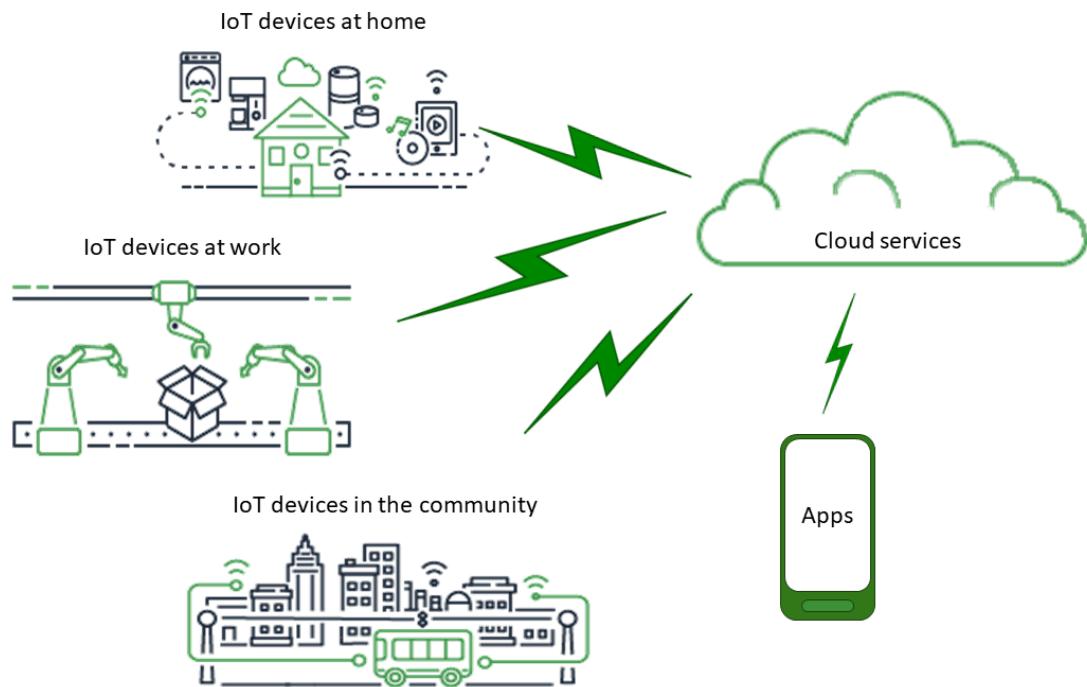
Pour obtenir une liste de solutions destinées à des applications industrielles, grand public et commerciales, consultez le [référentiel de AWS IoT solutions](#).

Fonctionnement d'AWS IoT

AWS IoT fournit des services cloud et une assistance aux appareils que vous pouvez utiliser pour mettre en œuvre des solutions IoT. AWS fournit de nombreux services cloud pour prendre en charge les applications basées sur l'IoT. Pour vous aider à comprendre par où commencer, cette section fournit un schéma et une définition des concepts essentiels pour vous présenter l'univers de l'IoT.

IoT univers

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



Applications

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 connectés à Internet. En voici quelques exemples :

- Services de connexion et de gestion IoT
 - AWS IoT est un exemple de service de connexion et de gestion IoT.
- Des services de calcul, tels qu'Amazon Elastic Compute Cloud et AWS 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 :

- Wi-Fi/Internet haut débit
- Données cellulaires haut débit
- Données cellulaires à bande étroite
- Réseau étendu (LoRaWAN) longue portée
- Communications RF propriétaires

Appareils

Un appareil 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, un processeur, de la mémoire. En voici quelques exemples :

- Raspberry
- Arduino
- Assistants d'interface vocale
- LoRaWAN et appareils
- Appareils Amazon Sidewalk
- Appareils IoT personnalisés

Interfaces

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

- Interfaces utilisateur

Des composants qui permettent aux appareils et utilisateurs de communiquer entre eux

- Interfaces d'entrée

Permettre à un utilisateur de communiquer avec un appareil

Exemples : clavier, bouton

- Interfaces de sortie

Permettre à un appareil de communiquer avec un utilisateur

Exemples : affichage alphanumérique, affichage graphique, voyant, sonnette d'alarme

- Capteurs

Composants d'entrée qui mesurent ou détectent quelque chose dans le monde extérieur d'une manière compréhensible par un appareil. En voici quelques exemples :

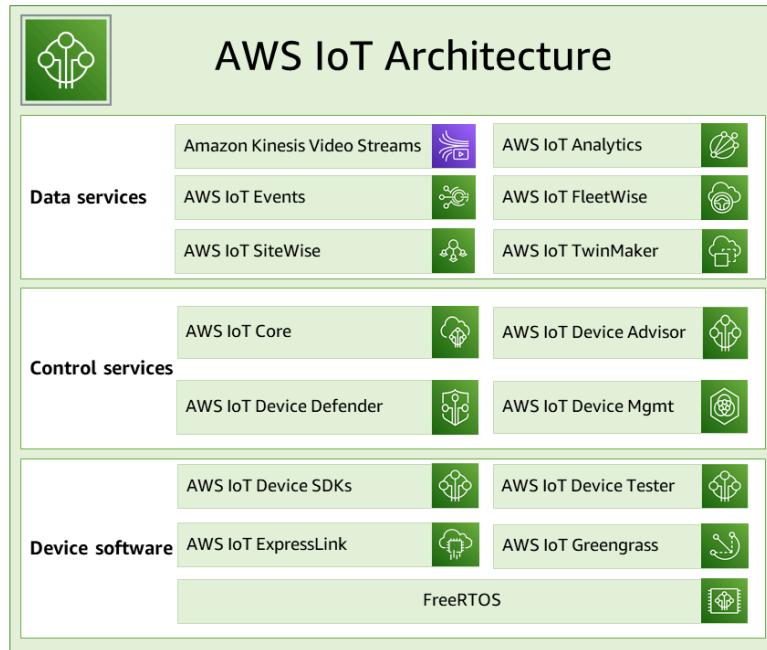
- Capteur de température (convertit la température en signal analogique ou numérique)
- Capteur d'humidité (convertit l'humidité relative en signal analogique ou numérique)
- Convertisseur analogique-numérique (convertit une tension analogique en valeur numérique)
- Unité de mesure de distance à ultrasons (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 (convertissent les signaux électriques en mouvement)
- Relais (contrôle des tensions et des courants électriques élevés)

AWS IoT Architecture

Dans l'univers de l'IoT, AWS IoT fournit les services qui prennent en charge les appareils qui interagissent avec le monde et les données qui transitent entre eux et AWS IoT. AWS IoT est composé des services présentés dans cette illustration pour soutenir votre solution IoT.



AWS IoT Logiciel de l'appareil

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

Kits SDK pour les appareils AWS IoT

Les [SDK pour AWS IoT appareils et mobiles \(p. 1494\)](#) vous aident à connecter efficacement vos appareils à AWS 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.

AWS IoT Device Tester

[AWS IoT Device Tester](#) pour FreeRTOS et AWS IoT Greengrass est un outil d'automatisation des tests pour les microcontrôleurs. AWS IoT Device Test teste votre appareil pour déterminer s'il fonctionnera avec FreeRTOS ou AWS IoT Greengrass s'il interagira avec AWS IoT des services.

AWS IoT ExpressLink

AWS IoT ExpressLink alimente une gamme de modules matériels développés et proposés par [AWS des partenaires](#). Les modules de connectivité incluent AWS un logiciel validé, ce qui vous permet de connecter plus rapidement et plus facilement des appareils au cloud en toute sécurité et de les intégrer facilement à une gamme de AWS services. Pour plus d'informations, visitez la page de [AWS IoT ExpressLink](#) présentation ou consultez le [guide du AWS IoT ExpressLink programmeur](#).

AWS IoT Greengrass

[AWS IoT Greengrass](#) vous permet AWS IoT d'étendre les services aux appareils en périphérie pour une exploitation en local des données qu'ils génèrent et utiliser les ressources du cloud pour la gestion, l'analyse et le stockage de longue durée. Avec AWS IoT Greengrass, les appareils connectés peuvent

exécuter des [AWS Lambda](#)fonctions, des conteneurs Docker, ou les deux, exécuter des prédictions basées sur des modèles d'apprentissage automatique, synchroniser les données des appareils et communiquer avec d'autres appareils en toute sécurité, même lorsqu'ils ne sont pas connectés à Internet.

FreeRTOS

[FreeRTOS](#) est un système d'exploitation open source en temps réel pour les microcontrôleurs qui vous permet d'inclure de petits appareils de périphérie à faible consommation d'énergie 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 d'énergie à des appareils de périphérie plus puissants [AWS IoT](#)et les prendre en charge [AWS IoT Greengrass](#).

AWS IoTservices de contrôle

Connect auxAWS IoT services suivants pour gérer les appareils de votre solution IoT.

AWS IoT Core

[AWS IoT Core](#)est un service cloud géré qui permet aux appareils connectés d'interagir en toute sécurité avec des applications cloud et d'autres appareils. AWS IoT Corepeut prendre en charge de nombreux appareils et messages, et il peut traiter et acheminer ces messages vers desAWS IoT points de terminaison et d'autres appareils. AvecAWS IoT Core, vos applications peuvent interagir avec tous vos appareils, même lorsqu'ils ne sont pas connectés.

AWS IoT Core Device Advisor

[AWS IoT CoreDevice Advisor](#) est une fonctionnalité de test entièrement gérée sur le cloud qui permet de valider les appareils IoT lors du développement logiciel des appareils. Device Advisor propose des tests prédéfinis que vous pouvez utiliser pour valider la fiabilité et la sécurité des appareils IoTAWS IoT Core, avant de les déployer en production.

Device Defender AWS IoT

[AWS IoTDevice Defender](#) vous aide à sécuriser votre parc d'appareils IoT. AWS IoT Device Defender audite 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, comme le partage de certificats d'identité sur plusieurs appareils ou la tentative de connexion d'un appareil dont le certificat d'identité a été révoqué [AWS IoT Core](#).

AWS IoTSemoteur

AWS IoTLes services [de gestion des appareils](#) vous aident à suivre, à surveiller et à gérer la pléthore d'appareils connectés qui constituent votre parc d'appareils. AWS IoT Les services de gestion des appareils vous aident à garantir que vos appareils IoT fonctionnent correctement et en toute sécurité après leur déploiement. Ils fournissent également un tunneling sécurisé pour accéder à vos appareils, surveiller leur état de santé, détecter et résoudre les problèmes à distance, ainsi que des services de gestion des mises à jour des logiciels et des microprogrammes des appareils.

Services de données AWS IoT

Analysez les données des appareils de votre solution IoT et prenez les mesures appropriées en utilisant lesAWS IoT services suivants.

Amazon Kinesis Video Streams

[Amazon Kinesis Video Streams](#) vous permet de diffuser des vidéos en direct depuis des appareils vers leAWS cloud, où elles sont stockées, cryptées et indexées de manière durable, ce qui vous

permet d'accéder à vos données via des easy-to-use API. Vous pouvez utiliser Amazon Kinesis Video Streams pour capturer d'énormes quantités de données vidéo en direct provenant de millions de sources, notamment des smartphones, des caméras de sécurité, des webcams, des caméras embarquées dans des voitures, des drones et d'autres sources. Amazon Kinesis Video Streams vous permet de lire des vidéos pour les visionner en direct et à la demande, et de créer rapidement des applications qui tirent parti de la vision par ordinateur et de l'analyse vidéo grâce à l'intégration à Amazon Rekognition Video et à des bibliothèques pour les frameworks de machine learning. Vous pouvez également envoyer données non vidéo sérialisées, par exemple des données audio, de l'imagerie thermique, de données de profondeur, des données RADAR, et bien plus encore.

Amazon Kinesis Video Streams avec WebRTC

[Amazon Kinesis Video Streams avec WebRTC](#) fournit une implémentation WebRTC conforme aux normes en tant que fonctionnalité entièrement gérée. Vous pouvez utiliser Amazon Kinesis Video Streams avec WebRTC pour diffuser du contenu multimédia en direct en toute sécurité ou effectuer une interaction audio ou vidéo bidirectionnelle entre n'importe quel appareil photo, appareil IoT et lecteurs mobiles ou Web compatibles WebRTC. En tant que fonctionnalité entièrement gérée, vous n'avez pas besoin de créer, d'exploiter ou de dimensionner une infrastructure cloud liée au WebRTC, telle que des serveurs de signalisation ou de relais multimédia pour diffuser du contenu multimédia en toute sécurité entre des applications et des appareils. En utilisant Amazon Kinesis Video Streams avec WebRTC, vous pouvez facilement créer des applications destinées à la diffusion peer-to-peer multimédia en direct ou à l'interactivité audio ou vidéo en temps réel entre les appareils photo, les appareils IoT, les navigateurs Web et les appareils mobiles pour divers cas d'utilisation.

Analyse AWS IoT

[AWS IoT L'analytique](#) vous permet d'exécuter et d'opérationnaliser efficacement des analyses sophistiquées sur d'énormes volumes de données IoT non structurées. AWS IoT L'analytique automatise chaque étape difficile nécessaire à l'analyse des données provenant d'appareils IoT. AWS IoT Analytics filtre, transforme et enrichit les données IoT avant de les stocker dans un magasin de données chronologiques à des fins d'analyse. Vous pouvez analyser vos données en exécutant des requêtes ponctuelles ou planifiées à l'aide du moteur de requêtes SQL intégré ou de l'apprentissage automatique.

Événements AWS IoT

[AWS IoT Events](#) détecte et répond aux événements provenant de capteurs et d'applications IoT. Les événements sont des modèles de données qui identifient des circonstances plus complexes que prévu, comme les détecteurs de mouvement utilisant des signaux de mouvement pour activer les lumières et les caméras de sécurité. AWS IoT Events surveille en permanence les données provenant de plusieurs capteurs et applications IoT et s'intègre à d'autres services AWS IoT Core, tels que l'IoT SiteWise, DynamoDB, etc., pour permettre une détection précoce et des informations uniques.

AWS IoT FleetWise

[AWS IoT FleetWise](#) est un service géré que vous pouvez utiliser pour collecter et transférer les données des véhicules vers le cloud en temps quasi réel. Avec AWS IoT FleetWise, vous pouvez facilement collecter et organiser les données provenant de véhicules utilisant différents protocoles et formats de données. AWS IoT FleetWise permet de transformer les messages de bas niveau en valeurs lisibles par l'homme et de normaliser le format des données dans le cloud pour les analyses de données. Vous pouvez également définir des schémas de collecte de données pour contrôler les données à collecter dans les véhicules et à quel moment les transférer vers le cloud.

AWS IoT SiteWise

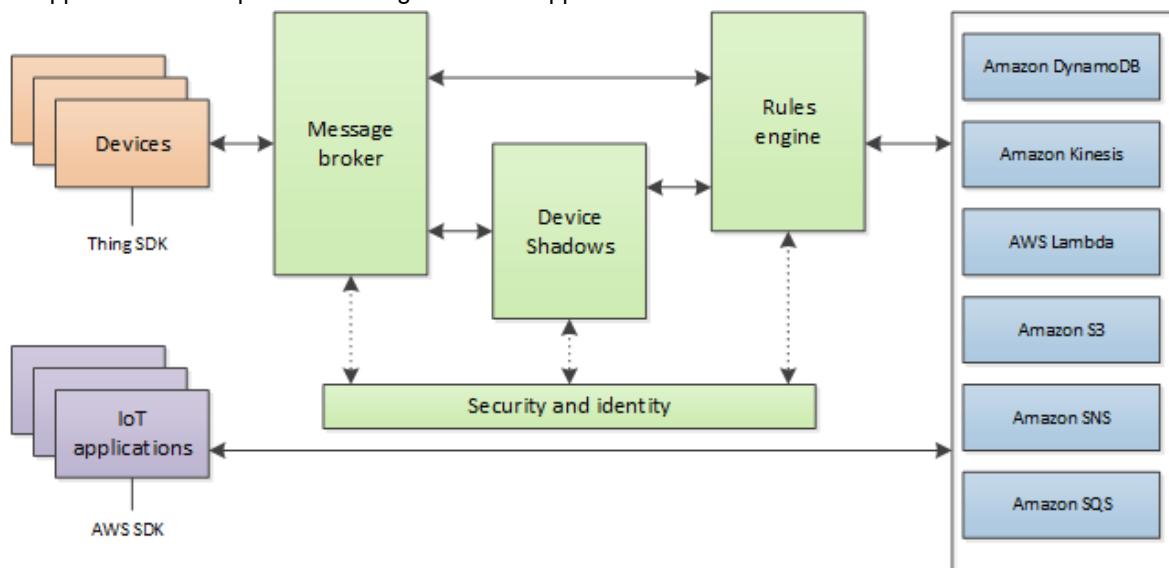
[AWS IoT SiteWise](#) collecte, stocke, organise et surveille les données transmises par des équipements industriels par des messages MQTT ou des API à grande échelle en fournissant un logiciel qui s'exécute sur une passerelle dans vos installations. La passerelle se connecte en toute sécurité à vos serveurs de données locaux et automatise le processus de collecte et d'organisation des données et de leur envoi vers le AWS cloud.

AWS IoT TwinMaker

[AWS IoT TwinMaker](#) construit des jumeaux numériques opérationnels de systèmes physiques et numériques. AWS IoT TwinMaker crée des visualisations numériques à l'aide de mesures et d'analyses provenant de divers capteurs, caméras et applications d'entreprise réels pour vous aider à suivre l'évolution de votre usine, de votre bâtiment ou de votre installation industrielle. Vous pouvez utiliser des données réelles pour surveiller les opérations, diagnostiquer et corriger les erreurs et optimiser les opérations.

AWS IoT Coreservices

AWS IoT Core fournit les services qui connectent vos appareils IoT au AWS cloud 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 services présentés dans l'illustration.

AWS IoT Coreservices de messagerie

Les services de AWS IoT Core connectivité fournissent une communication sécurisée avec les appareils IoT et gèrent les messages qui passent entre eux et AWS IoT.

Passerelle pour les appareils

Permet aux appareils de communiquer de manière efficace et en toute sécurité avec AWS IoT. La communication entre les appareils est sécurisée par des protocoles sécurisés utilisant des 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 over WebSocket pour publier et vous abonner. Pour de plus amples informations sur les protocoles pris en AWS IoT charge, veuillez consulter [the section called “Protocoles de communication des appareils” \(p. 89\)](#). Les appareils et les clients peuvent également utiliser l'interface REST HTTP pour publier des données sur le courtier de messages.

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

AWS IoT Corepour LoRa WAN

AWS IoT Corefor LoRa WAN permet de configurer un réseau LoRa WAN privé en connectant vos périphériques et passerelles LoRa WANAWS sans avoir à développer et à exploiter un serveur réseau LoRa WAN (LNS). Les messages reçus des périphériques LoRa WAN sont envoyés au moteur de règles où ils peuvent être formatés et envoyés à d'autresAWS IoT services.

Moteur de règles

Le moteur Rules connecte les données du courtier de messages à d'autresAWS IoT services à des fins de stockage et de traitement supplémentaire. Par exemple, vous pouvez 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, et les envoyer à d'autres services tels que Amazon Simple Storage Service (Amazon S3) Amazon DynamoDB etAWS Lambda. Vous pouvez également créer des règles qui republient les messages vers le courtier de messages et vers d'autres abonnés. Pour plus d'informations, veuillez consulter [Règles pour AWS IoT \(p. 524\)](#).

AWS IoT Coreservices de contrôle

Les servicesAWS IoT Core de contrôle fournissent des fonctionnalités de sécurité, de gestion et d'enregistrement des appareils.

Service d'authentification personnalisé

Vous pouvez définir des autorisations personnalisées 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 autorisateurs personnalisés peuvent implémenter différentes stratégies d'authentification, par exemple la vérification du jeton Web JSON ou l'appel du fournisseur OAuth. Ils doivent renvoyer les documents de politique utilisés par la passerelle de l'appareil pour autoriser les opérations MQTT. Pour plus d'informations, veuillez consulter [Authentification et autorisation personnalisées \(p. 343\)](#).

Service de mise en service d'appareils

Vous permet de provisionner des appareils à l'aide d'un modèle qui décrit les ressources requises pour votre appareil : un objet, un certificat et une ou plusieurs politiques. Un objet est une entrée du registre qui contient des attributs décrivant un périphérique. 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, veuillez consulter [Mise en service des appareils \(p. 894\)](#).

Registre de groupe

Les groupes vous permettent de gérer plusieurs appareils simultanément en les classant dans des groupes. Les groupes peuvent également contenir des groupes : vous pouvez créer une hiérarchie de groupes. Toute action que vous effectuez sur un groupe parent s'appliquera à ses groupes enfants. La même action s'applique également à tous les appareils du groupe parent et à tous les appareils des groupes enfants. Les autorisations accordées à un groupe s'appliqueront à tous les appareils du groupe et à tous ses groupes enfants. Pour plus d'informations, veuillez consulter [Gestion des appareils avec AWS IoT \(p. 287\)](#).

Service Jobs

Vous permet de définir un ensemble d'opérations à distance qui sont envoyées et exécutées sur un ou plusieurs appareils connectés AWS 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, veuillez consulter [Tâches \(p. 739\)](#).

Registre

Organise les ressources associées à chaque appareil dans le AWS Cloud. 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, veuillez consulter [Gestion des appareils avec AWS IoT \(p. 287\)](#).

Service de sécurité et d'identité

Fournit une responsabilité partagée en matière de sécurité dans le AWS cloud. Vos appareils doivent conserver leurs informations d'identification en toute sécurité pour envoyer des données au courtier de messages en toute sécurité. Le courtier de messages et le moteur de règles utilisent des fonctionnalités AWS de sécurité pour envoyer des données en toute sécurité à des appareils ou à d'autres AWS services. Pour plus d'informations, veuillez consulter [Authentification \(p. 316\)](#).

Services de données AWS IoT Core

Les services de AWS IoT Core données aident vos solutions IoT à fournir 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 appareil afin que les applications puissent communiquer avec celui-ci, que celui-ci soit en ligne ou non. Lorsqu'un appareil est hors ligne, le service Device Shadow gère ses données pour les applications connectées. Lorsque l'appareil se reconnecte, il synchronise son état avec celui de son ombre dans le service Device Shadow. Vos appareils peuvent également publier leur état actuel sur un shadow afin qu'ils puissent être utilisés par des applications ou d'autres appareils qui ne sont peut-être pas connectés en permanence. Pour plus d'informations, veuillez consulter [Service AWS IoT Device Shadow \(p. 690\)](#).

AWS IoT Coreservice d'assistance

Intégration à Amazon Sidewalk pour AWS IoT Core

[Amazon Sidewalk](#) est un réseau partagé qui améliore les options de connectivité pour aider les appareils à mieux fonctionner ensemble. Amazon Sidewalk est compatible avec un large éventail d'appareils clients, tels que ceux qui permettent de localiser les animaux domestiques ou les objets de valeur, ceux qui fournissent une sécurité domestique intelligente et un contrôle de l'éclairage, et ceux qui fournissent des diagnostics à distance pour les appareils et les outils. L'intégration d'Amazon Sidewalk AWS IoT Core permet aux fabricants d'appareils d'ajouter leur parc d'appareils Sidewalk au AWS IoT cloud.

Pour de plus amples informations, consultez [AWS IoT Core pour Amazon Sidewalk \(p. 1400\)](#).

En savoir plus sur AWS IoT

Cette rubrique vous permet de vous familiariser avec le monde deAWS IoT. Vous pouvez obtenir des informations générales sur la manière dont les solutions IoT sont appliquées dans différents cas d'utilisation, des ressources de formation, des liens vers les réseaux sociauxAWS IoT et tous les autresAWS services, ainsi qu'une liste des services et des protocoles de communicationAWS IoT utilisés.

Ressources de formation pourAWS IoT

Nous proposons ces cours de formation pour vous aider à en savoir plusAWS IoT et à les appliquer à la conception de votre solution.

- [Introduction à AWS IoT](#)

Une vidéo de présentationAWS IoT et de ses principaux services.

- [Exploration approfondie deAWS IoT l'authentification et de l'autorisation](#)

Un cours avancé qui explore les concepts d'AWS IoT authentification et d'autorisation. Vous apprendrez comment authentifier et autoriser les clients à accéder aux API du planAWS IoT de contrôle et du plan de données.

- [Série Internet of Things Foundation](#)

Un parcours d'apprentissage composé de modules d'apprentissage en ligne sur les différentes technologies et fonctionnalités de l'IoT.

AWS IoTressources et guides

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

- [IoT Lens :AWS IoT un cadre Well-Architected](#)

Un document qui décrit les meilleures pratiques pour l'architecture de vos applications IoT surAWS.

- [Conception de rubriques MQTT pourAWS IoT Core](#)

Un livre blanc qui décrit les meilleures pratiques pour concevoir des rubriques MQTTAWS IoT Core et tirer parti desAWS IoT Core fonctionnalités de MQTT.

- <https://docs.aws.amazon.com/https://docs.aws.amazon.com/whitepapers/latest/device-manufacturing-provisioning/device-manufacturing-provisioning.html>

Un document PDF qui décrit les différentes méthodesAWS IoT permettant de provisionner de grandes flottes d'appareils.

- [AWS IoT Core Device Advisor](#)

AWS IoT CoreDevice Advisor propose des tests prédéfinis que vous pouvez utiliser pour valider les meilleures pratiques en matière de connectivité fiable et sécurisée sur les appareils IoTAWS IoT Core, avant de les déployer en production.

- [Ressources AWS IoT](#)

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

- [Atlas de l'IoT](#)

Aperçus sur la manière de résoudre les problèmes de conception courants de l'IoT. L'Atlas de l'IoT fournit 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 sur AWS IoT d'autres AWS technologies.

AWS IoT sur les réseaux sociaux

Ces réseaux sociaux fournissent des informations sur AWS IoT des sujets AWS connexes.

- [L'Internet des objets sur AWS IoT — Blog officiel](#)
- [AWS IoT Vidéos de la chaîne Amazon Web Services sur YouTube](#)

Ces comptes de réseaux sociaux couvrent tous les AWS services, y compris AWS IoT

- [La chaîne Amazon Web Services sur YouTube](#)
- [Amazon Web Services sur Twitter](#)
- [Amazon Web Services sur Facebook](#)
- [Amazon Web Services sur Instagram](#)
- [Amazon Web Services sur LinkedIn](#)

AWS services utilisés par le moteur de AWS IoT Core règles

Le moteur de AWS IoT Core règles peut se connecter à ces AWS services.

- [Amazon DynamoDB](#)

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

- [Amazon Kinesis](#)

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

- [AWS Lambda](#)

AWS Lambda vous permet d'exécuter du code sans avoir à louer ou gérer des serveurs. Vous pouvez configurer votre code pour qu'il se déclenche automatiquement à partir de AWS IoT données et d'événements ou pour l'appeler directement depuis une application Web ou mobile.

- [Amazon Simple Storage Service](#)

Amazon Simple Storage Service (Amazon S3) peut stocker et récupérer n'importe quelle quantité de données à tout moment, de n'importe où sur le web. AWS IoT les règles peuvent envoyer des données à Amazon S3 pour stockage.

- [Amazon Simple Notification Service](#)

Amazon Simple Notification Service (Amazon SNS) est un service web qui permet aux applications, utilisateurs finaux et appareils d'envoyer et de recevoir des notifications au cloud ou d'en recevoir.

- [Amazon Simple Queue Service](#)

Amazon Simple Queue Service (Amazon SQS) est un service de mise en file d'attente de messages qui dissocie et adapte les microservices, les systèmes distribués et les applications sans serveur.

- [OpenSearch Service Amazon](#)

Amazon OpenSearch Service (Service) est unOpenSearch service géré qui facilite le déploiement, l'utilisation et le dimensionnement d'un moteur d'analyse et de recherche open source couramment utilisé utilisé couramment utilisé pour effectuer des recherches et des services d'analyse open source couramment utilisé utilisé pour le déploiement OpenSearch, l'utilisation et le dimensionnement.

- [Amazon SageMaker](#)

Amazon SageMaker peut créer des modèles de machine learning (ML) en repérant des modèles dans vos données de l'SSSSSiteSSSSpot. 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 pour vous aider à configurer, gérer et dimensionner vos propres systèmes et infrastructures de surveillance.

Protocoles de communication pris en charge parAWS IoT Core

Ces rubriques fournissent plus d'informations sur les protocoles de communication utilisés parAWS IoT. Pour plus d'informations sur les protocoles utilisés par les appareilsAWS IoT et les services auxquels ils connectentAWS IoT, consultez[Connexion à AWS IoT Core \(p. 76\)](#).

- [MQTT \(transport de télémétrie par file d'attente de messages\)](#)

La page d'accueil du site MQTT.org où vous pouvez trouver les spécifications du protocole MQTT. Pour plus d'informations sur la prise enAWS IoT charge du format MQTT, consultez[MQTT \(p. 92\)](#).

- [HTTPS \(protocole de transfert hypertexte - sécurisé\)](#)

Les appareils et les applications peuvent accéder auxAWS IoT services via HTTPS.

- [LoRaWAN \(réseau étendu à longue portée\)](#)

LoRaLes périphériques et passerelles WAN peuvent se connecter àAWS IoT Core l'aideAWS IoT Core de for LoRa WAN.

- [TLS \(Transport Layer Security\)](#)

Spécification du protocole TLS v1.2 (RFC 5246). AWS IoTUtilise le protocole TLS v1.2 pour établir des connexions sécurisées entre les appareils etAWS IoT.

Nouveautés dans la nouvelleAWS IoT clé de

Nous sommes en train de mettre à jour l'interface utilisateur de laAWS IoT console pour une nouvelle expérience. Nous mettons à jour l'interface utilisateur par étapes. Ainsi, certaines pages de la console proposeront une nouvelle expérience, d'autres proposeront à la fois l'expérience d'origine et la nouvelle, et d'autres proposeront uniquement l'expérience d'origine.

Ce tableau affiche l'état des différentes zones de l'interface utilisateur de la AWS IoT console au 27 janvier 2022.

AWS IoT état de l'interface utilisateur de la console

Page de console	de de de de de	de de de de de	Commentaires
Moniteur	Non disponible	Disponible	
Activité	Non disponible	Disponible	
À bord - Commencez	Non disponible	Disponible	Non disponible dans les régions du CN
Onboard - Modèles de provisionnement de flotte	Disponible	Disponible	
Gérer - Objets	Disponible	Disponible	
Gérer - Types	Disponible	Disponible	
Gérer : groupes d'objets	Disponible	Disponible	
Gérer - Groupes de facturation	Disponible	Disponible	
Gérer - Offres d'emploi	Disponible	Disponible	
Gérer - Modèles de Job	Non disponible	Disponible	
Gérer - Tunnels	Non disponible	Disponible	
Fleet Hub - Commencez	Non disponible	Disponible	Non disponible dans tousRégions AWS
Fleet Hub - Applications	Non disponible	Disponible	Non disponible dans tousRégions AWS
Greengrass - Pour commencer	Non disponible	Disponible	Non disponible dans tousRégions AWS
Greengrass - Appareils principaux	Non disponible	Disponible	Non disponible dans tousRégions AWS
Greengrass - Composants	Non disponible	Disponible	Non disponible dans tousRégions AWS
Greengrass - Déploiements	Non disponible	Disponible	Non disponible dans tousRégions AWS
Greengrass - Classique (V1)	Disponible	Disponible	
Connectivité sans fil - Intro	Non disponible	Disponible	Non disponible dans tousRégions AWS
Connectivité sans fil - Passerelles	Non disponible	Disponible	Non disponible dans tousRégions AWS
Connectivité sans fil - Appareils	Non disponible	Disponible	Non disponible dans tousRégions AWS

Page de console	de de de de de	de de de de	Commentaires
Connectivité sans fil - Profils	Non disponible	Disponible	Non disponible dans tousRégions AWS
Connectivité sans fil - Destinations	Non disponible	Disponible	Non disponible dans tousRégions AWS
Sécurisé - Certificats	Disponible	Disponible	
Sécurisé - Politiques	Disponible	Disponible	
Sécurisé - CA	Disponible	Disponible	
Secure - Alias de rôle	Disponible	Disponible	
Sécurisé - Autorisateurs	Disponible	Disponible	
Defend - Intro	Non disponible	Disponible	
Défendez - Audit	Non disponible	Disponible	
Défendez - Déetectez	Non disponible	Disponible	
Défendez - Actions d'atténuation	Non disponible	Disponible	
Defend - Paramètres	Non disponible	Disponible	
Loi - Règles	Disponible	Disponible	
Loi - Destinations	Disponible	Disponible	
Test - Device Advisor	Disponible	Disponible	Non disponible dans tousRégions AWS
Test : client de test MQTT	Disponible	Disponible	
Logiciels	Disponible	Disponible	
Paramètres	Non disponible	Disponible	
Apprenez	Disponible	Pas encore disponible	

Légende

Valeurs de statut de

- Disponible

Cette expérience d'interface utilisateur peut être utilisée.

- Non disponible

Cette interface utilisateur ne peut pas être utilisée.

- Pas encore disponible

La nouvelle interface utilisateur est en cours d'élaboration, mais elle n'est pas encore prête.

- En cours

La demande de est en cours d'exécution. Certaines pages peuvent toutefois conserver l'expérience utilisateur d'origine.

Démarrer avec AWS IoT Core

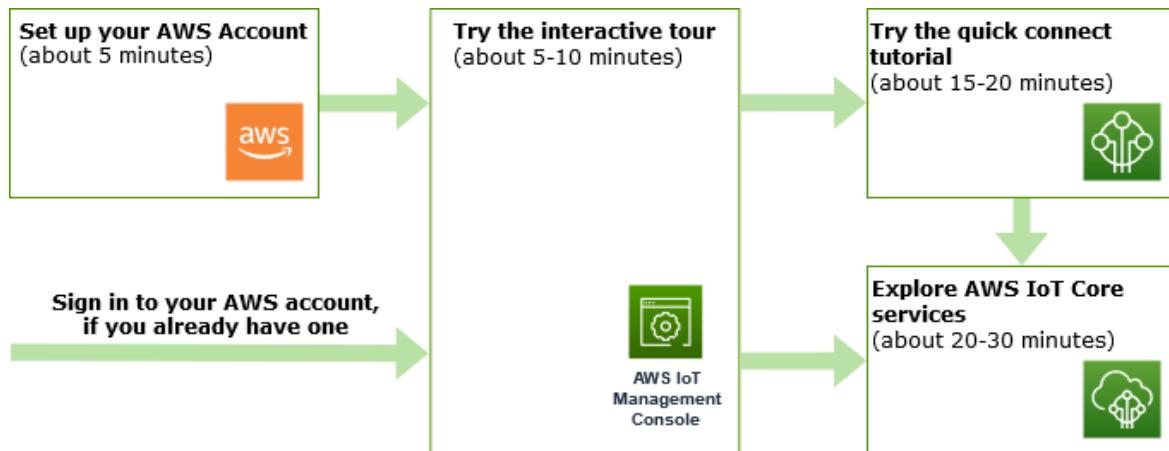
Que vous soyez novice en matière d'IoT ou que vous ayez des années d'expérience, ces ressources présentent les AWS IoT concepts et les termes qui vous aideront à commencer à l'utiliser AWS IoT.

- Regardez à l'intérieur AWS IoT et ses composants à l'intérieur [Fonctionnement d'AWS IoT \(p. 3\)](#).
- [Pour en savoir plus \(p. 12\)](#), AWS IoT consultez notre collection de supports de formation et de vidéos. Cette rubrique inclut également une liste de services auxquels il est AWS IoT possible de se connecter, des liens vers les réseaux sociaux et des liens vers les spécifications du protocole de communication.
- [the section called "Connect votre premier appareil à AWS IoT Core" \(p. 18\)](#).
- Développez vos solutions IoT [Connexion à AWS IoT Core \(p. 76\)](#) en explorant le [Didacticiels AWS IoT \(p. 143\)](#).
- Testez et validez vos appareils IoT pour une communication sécurisée et fiable en utilisant le [Device Advisor \(p. 1167\)](#).
- Gérez votre solution à l'aide AWS IoT Core de services de gestion tels que [Indexation de la flotte \(p. 929\)](#) [Tâches \(p. 739\)](#), et [AWS IoT Device Defender \(p. 975\)](#).
- Analysez les données de vos appareils à l'aide du [Services de données AWS IoT \(p. 7\)](#).

Connect votre premier appareil à AWS IoT Core

AWS IoT Core les services connectent les appareils IoT à AWS IoT des services et à d'autres AWS services. AWS IoT Core inclut la passerelle pour appareils et le courtier de messages, qui connectent et traitent les messages entre vos appareils IoT et le cloud.

Voici comment commencer avec AWS IoT Core et AWS IoT.



Cette section présente ses principaux services et fournit plusieurs exemples de connexion d'un appareil à un appareil AWS IoT Core et de transmission de messages entre eux. AWS IoT Core La transmission de messages entre les appareils et le cloud est essentielle à toute solution IoT et permet à vos appareils d'interagir avec d'autres AWS services.

- [Configurez votre Compte AWS \(p. 19\)](#)

Avant de pouvoir utiliser AWS IoT les services, vous devez configurer un Compte AWS. Si vous avez déjà un utilisateur Compte AWS et un utilisateur IAM pour vous-même, vous pouvez les utiliser et ignorer cette étape.

- [Essayez le didacticiel interactif \(p. 21\)](#)

Cette démo est idéale si vous souhaitez découvrir ce qu'une AWS IoT solution de base peut faire sans connecter un appareil ni télécharger de logiciel. Le didacticiel interactif présente une solution simulée basée sur AWS IoT Core des services qui illustre la façon dont ils interagissent.

- [Essayez le didacticiel de connexion rapide \(p. 23\)](#)

Ce didacticiel est idéal si vous souhaitez démarrer rapidement AWS IoT et voir comment il fonctionne dans un scénario limité. Dans ce didacticiel, vous aurez besoin d'un appareil sur lequel vous installerez des AWS IoT logiciels. Si vous ne possédez pas d'appareil IoT, vous pouvez utiliser votre ordinateur personnel Windows, Linux ou macOS comme appareil pour ce didacticiel. Si vous voulez essayer AWS IoT, mais que vous ne possédez pas d'appareil, essayez l'option suivante.

- [Découvrez les AWS IoT Core services grâce à un didacticiel pratique \(p. 38\)](#)

Ce didacticiel est idéal pour les développeurs qui AWS IoT souhaitent commencer à explorer d'autres AWS IoT Core fonctionnalités 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 afin de faciliter la transition vers les didacticiels plus avancés.

- [Afficher les messages MQTT avec le client AWS IoT MQTT \(p. 71\)](#)

Découvrez comment utiliser le client de test MQTT pour regarder votre premier appareil publier des messages MQTT. AWS IoT Le client de test MQTT est un outil utile pour surveiller et résoudre les problèmes de connexion des appareils.

Note

Si vous souhaitez essayer plusieurs de ces didacticiels de mise en route ou répéter le même didacticiel, vous devez supprimer l'objet que vous avez créé à partir d'un didacticiel précédent avant d'en commencer un autre. Si vous ne supprimez pas l'objet d'un didacticiel précédent, vous devrez utiliser un nom d'objet différent pour les didacticiels suivants. En effet, le nom de l'objet doit être unique dans votre compte etRégion AWS.

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

Configurez votre Compte AWS

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

Rubriques

- [S'inscrire à un Compte AWS \(p. 19\)](#)
- [Création d'un utilisateur administratif \(p. 20\)](#)
- [Ouvrez la AWS IoT console. \(p. 20\)](#)

S'inscrire à un Compte AWS

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

Pour s'inscrire à 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 saisierez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous souscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur root a accès à l'ensemble des Services AWS et des ressources de ce compte. En tant que bonne pratique de sécurité, [attribuer un accès administratif à un utilisateur administratif](#), et utilisez uniquement l'utilisateur root pour effectuer [tâches nécessitant un accès utilisateur root](#).

AWS vous envoie un e-mail de confirmation lorsque le processus d'inscription est terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à <https://aws.amazon.com/> et en cliquant sur Mon compte.

Création d'un utilisateur administratif

Une fois que vous êtes inscrit à un Compte AWS, créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisation de votre Utilisateur racine d'un compte AWS

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

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

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

Pour obtenir des instructions, consultez [Activation d'un dispositif MFA virtuel pour l'utilisateur root de votre Compte AWS \(console\)](#) dans le Guide de l'utilisateur IAM.

Création d'un utilisateur administratif

- Pour vos tâches administratives quotidiennes, octroyez un accès administratif à un utilisateur administratif dans AWS IAM Identity Center (successor to AWS Single Sign-On).

Pour plus d'informations, consultez [Mise en route](#) dans le Guide de l'utilisateur AWS IAM Identity Center (successor to AWS Single Sign-On).

Connexion en tant qu'utilisateur administratif

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

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

- [Ouvrez la AWS IoT console. \(p. 20\)](#)

Si vous avez déjà un Compte AWS et un utilisateur pour vous, vous pouvez les utiliser et passer directement à [the section called "Ouvrez la AWS IoT console." \(p. 20\)](#).

Ouvrez la AWS IoT console.

[La plupart des rubriques de cette section relatives à la console partent de la AWS IoT console.](#) Si vous n'êtes pas encore connecté à votre Compte AWS, connectez-vous, puis ouvrez la [AWS IoT console](#) et passez à la section suivante pour continuer à démarrer AWS IoT.

Essayez le didacticiel AWS IoT Core interactif

Le didacticiel interactif présente les composants d'une solution IoT simple basée sur AWS IoT. Les animations du didacticiel montrent comment les appareils IoT interagissent avec AWS IoT Core et les services. Cette rubrique fournit un aperçu du didacticiel AWS IoT Core interactif. Les images de la console incluent des animations qui n'apparaissent pas dans les images de ce didacticiel.

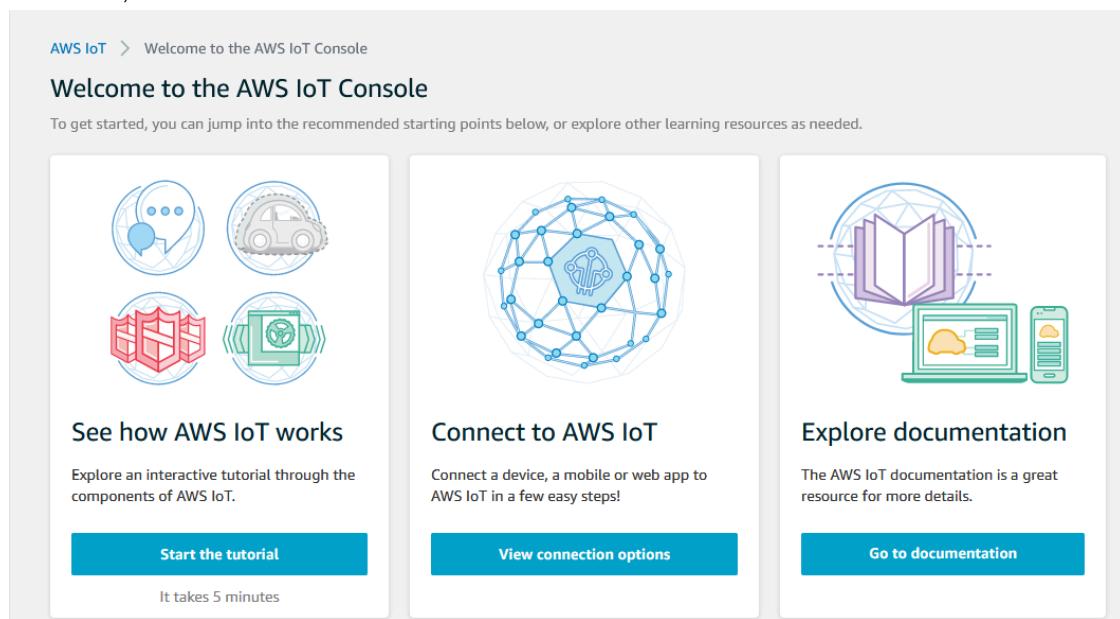
Pour lancer la démo, vous devez d'abord [the section called “Configurez votre Compte AWS” \(p. 19\)](#). Le didacticiel ne nécessite toutefois aucune AWS IoT ressource, aucun logiciel supplémentaire ni aucun codage.

Attendez-vous à consacrer environ 5 à 10 minutes à cette démo. Si vous accordez 10 minutes, vous aurez plus de temps pour réfléchir à chacune des étapes.

Pour exécuter le didacticiel AWS IoT Core interactif

1. Ouvrez le [Learning Hub](#) dans la AWS IoT console.

Sur la page Bienvenue sur la AWS IoT console, dans la vignette Découvrez comment AWS IoT fonctionne, choisissez Démarrer le didacticiel.



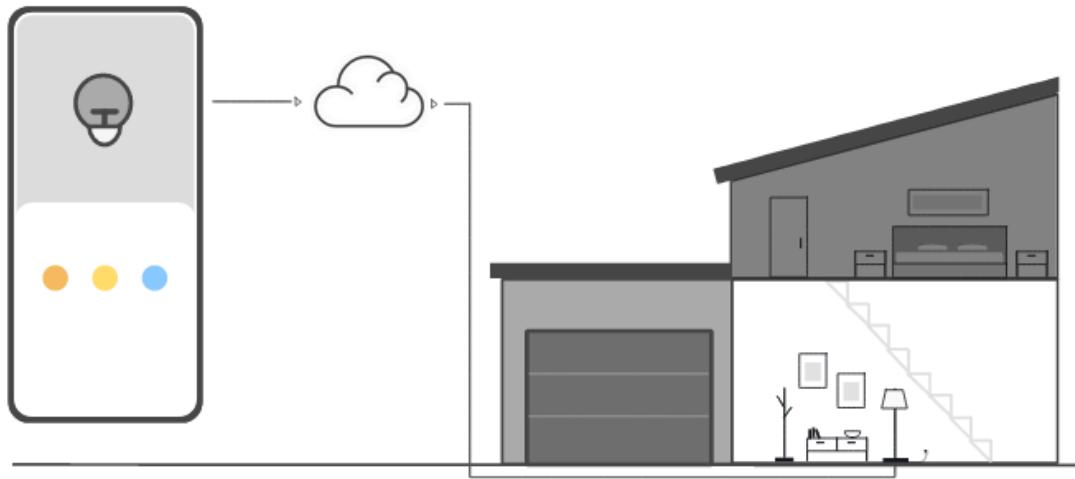
2. Sur la page du didacticiel AWS IoT interactif, passez en revue les différentes parties du didacticiel, puis choisissez Démarrer le didacticiel lorsque vous êtes prêt à continuer.

Les sections suivantes décrivent comment le didacticiel AWS IoT interactif présente ces AWS IoT Core fonctionnalités :

- [Connexion des appareils IoT \(p. 21\)](#)
- [Enregistrer l'état de l'appareil hors ligne \(p. 22\)](#)
- [Routage des données de l'appareil vers les services \(p. 23\)](#)

Connexion des appareils IoT

Découvrez comment les appareils IoT communiquent avec AWS IoT Core.

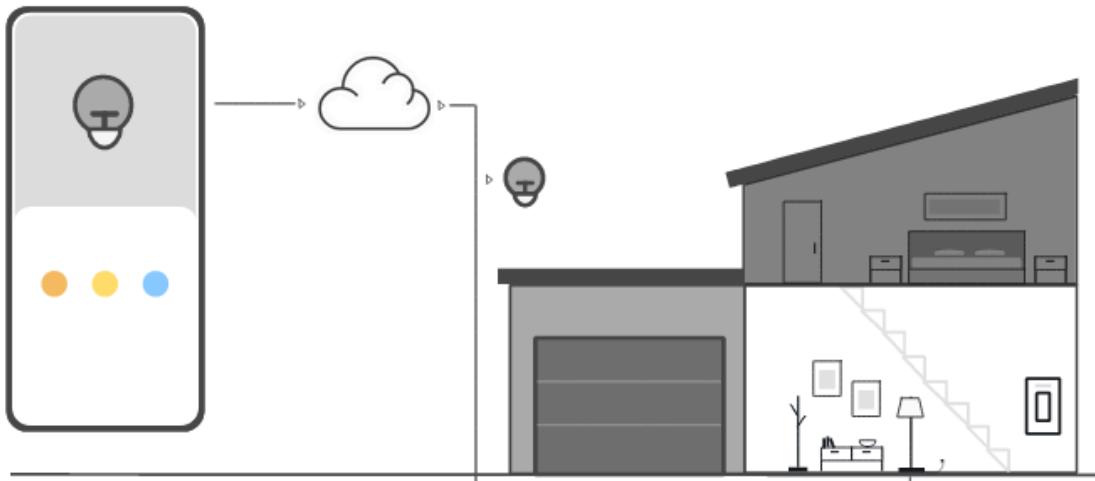


L'animation de cette étape montre comment deux appareils, l'appareil de commande sur la gauche et une lampe intelligente dans la maison sur la droite, se connectent et communiquent avec eux AWS IoT Core dans le cloud. L'animation montre les appareils communiquant avec AWS IoT Core les messages qu'ils reçoivent et y réagissent.

Pour plus d'informations sur la connexion de périphériques àAWS IoT Core, consultez[Connexion à AWS IoT Core \(p. 76\)](#).

Enregistrer l'état de l'appareil hors ligne

Découvrez comment AWS IoT Core enregistrer l'état de l'appareil lorsqu'un appareil ou une application est hors ligne.



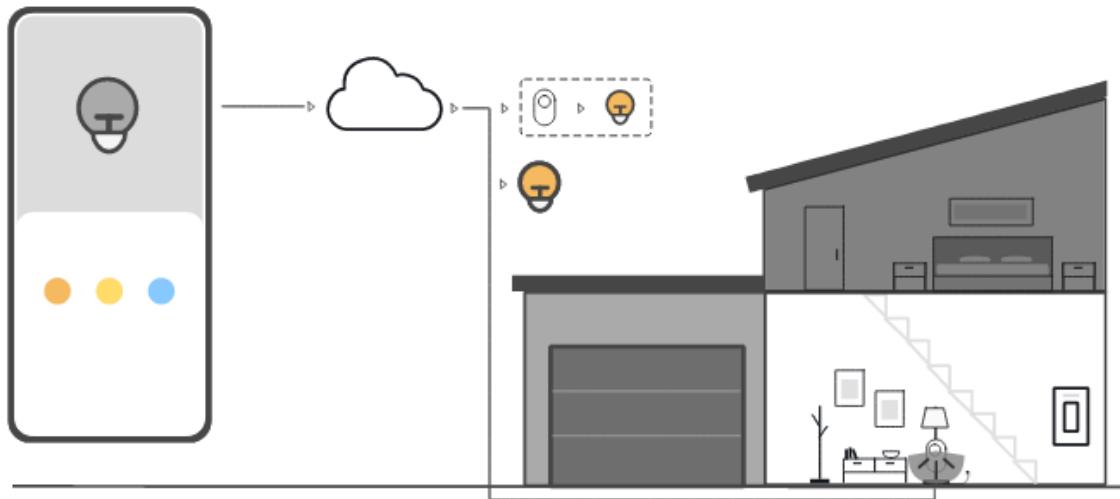
L'animation de cette étape montre comment le service Device Shadow AWS IoT Core enregistre les informations sur l'état de l'appareil pour le dispositif de contrôle et la lampe intelligente. Lorsque la lampe intelligente est hors ligne, le Device Shadow enregistre les commandes de l'appareil de contrôle.

Lorsque la lampe intelligente se reconnecte à AWS IoT Core, elle récupère ces commandes. Lorsque l'appareil de commande est hors ligne, Device Shadow enregistre les informations d'état de la lampe intelligente. Lorsque le dispositif de commande se reconnecte, il récupère l'état actuel de la lampe intelligente pour mettre à jour son affichage.

Pour d'autre information sur AWS IoT Device Shadow, consultez [Service AWS IoT Device Shadow \(p. 690\)](#).

Routage des données de l'appareil vers les services

Découvrez comment transmettre AWS IoT Core l'état de l'appareil à d'autres AWS services.



L'animation de cette étape montre comment AWS IoT Core envoie des données depuis les appareils vers d'autres AWS services à l'aide de AWS IoT règles. AWS IoT règles s'abonnent à des messages spécifiques provenant des appareils, interprètent les données contenues dans ces messages et acheminent les données interprétées vers d'autres services. Dans cet exemple, une AWS IoT règle interprète les données d'un capteur de mouvement et envoie des commandes à Device Shadow, qui les envoie ensuite à l'ampoule intelligente. Comme dans l'exemple précédent, Device Shadow enregistre les informations relatives à l'état du périphérique de contrôle.

Pour plus d'informations sur les règles du AWS IoT, consultez [Règles pour AWS IoT \(p. 524\)](#).

Essayez la connexion AWS IoT 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 consacrer 15 à 20 minutes à ce didacticiel.

Ce didacticiel est idéal pour les personnes qui souhaitent démarrer rapidement AWS IoT pour voir comment il fonctionne dans un scénario limité. Si vous recherchez un exemple qui vous permettra de démarrer et d'explorer davantage de fonctionnalités et de services, essayez [Découvrez les AWS IoT Core services dans le cadre d'un didacticiel pratique \(p. 38\)](#).

Dans ce didacticiel, vous allez télécharger et exécuter un logiciel sur un appareil qui se connecte à une ressource objet dans AWS IoT Core le cadre d'une toute petite solution IoT. L'appareil peut être un appareil

IoT, tel qu'un Raspberry Pi, ou il peut également s'agir d'un ordinateur exécutant Linux, OS et OSX, ou Windows. Si vous souhaitez connecter un périphérique WAN longue portée (LoRaWAN) àAWS IoT, consultez le didacticiel [Connecter des appareils et des passerelles à AWS IoT Core un LoRa réseau étendu \(p. 1272\)](#).

Si votre appareil est compatible avec un navigateur capable d'exécuter la [AWS IoTconsole](#), nous vous recommandons de suivre ce didacticiel sur cet appareil.

Note

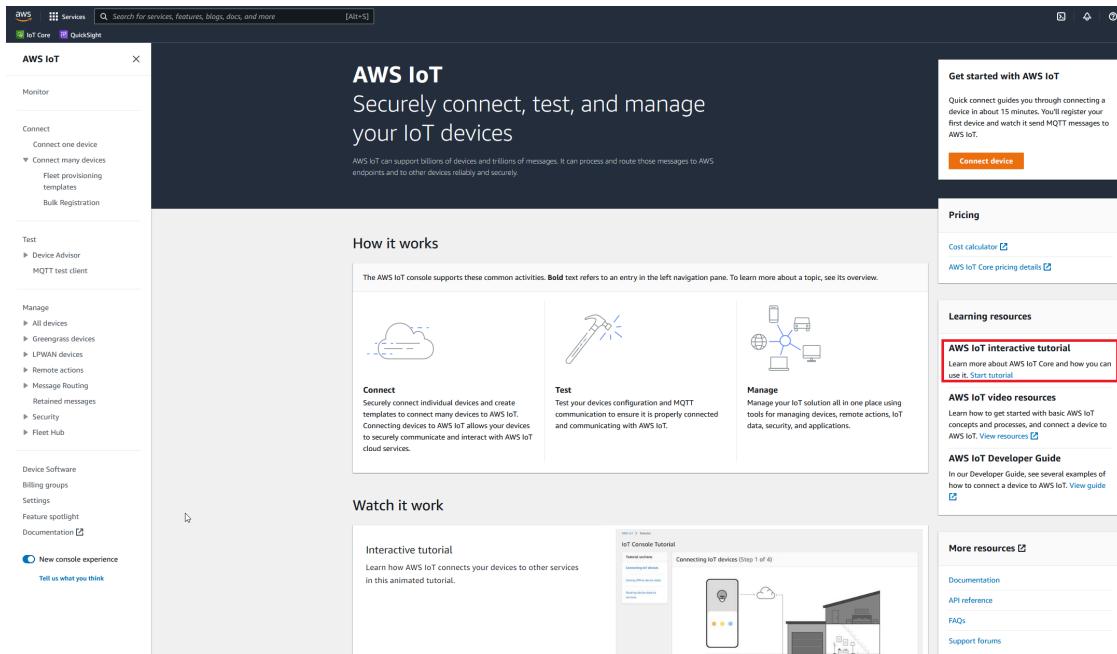
Si votre appareil ne dispose pas d'un navigateur compatible, suivez ce didacticiel sur un ordinateur. Lorsque la procédure vous demande de télécharger le fichier, téléchargez-le sur votre ordinateur, puis transférez le fichier téléchargé sur votre appareil à l'aide de Secure Copy (SCP) ou d'un processus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

Le didacticiel nécessite que votre appareil IoT communique avec le port 8443 du point de terminaison Compte AWS de données de votre appareil. Pour vérifier s'il peut accéder à ce port, essayez les procédures décrites dans[Tester la connectivité avec le point de terminaison de données de votre appareil \(p. 34\)](#).

Étape 1. Démarrez le didacticiel

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

1. Connectez-vous à la [console AWS IoT](#). Sur la page d'accueil de la AWS IoT console, dans Ressources pédagogiques sur la droite, choisissez Démarrer le didacticiel dans le didacticiel AWS IoT interactif.



2. Dans la AWS IoT vignette Connect à, choisissez Afficher les options de connexion.

The screenshot shows the AWS IoT Console's welcome screen. On the left is a sidebar with navigation links: Monitor, Connect (with options for one or many devices), Test (Device Advisor and MQTT test client), Manage (All devices, Greengrass devices, LPWAN devices, Remote actions, Message Routing, Retained messages, Security, Fleet Hub), Device Software, Billing groups, Settings, Feature spotlight, Documentation, and a New console experience toggle. Below the sidebar is a "Tell us what you think" link. The main content area is titled "Welcome to the AWS IoT Console" and includes a message: "To get started, you can jump into the recommended starting points below, or explore other learning resources as needed." It features three main sections: "See how AWS IoT works" (with icons for a speech bubble, a car, a server, and a device), "Connect to AWS IoT" (with an icon of a network graph), and "Explore documentation" (with icons for a book, a laptop, and a smartphone). A red box highlights the "View connection options" button under the "Connect to AWS IoT" section.

Étape 2. Création d'un objet

1. Dans la section Préparez votre appareil, suivez les instructions qui s'affichent à l'écran pour préparer votre appareil en vue de la connexion àAWS IoT.

2. Dans la section Enregistrer et sécuriser votre appareil, choisissez Créer un nouvel objet ou Choisir un objet existant. Dans le champ Nom de l'objet, entrez le nom de votre objet. Le nom de l'objet utilisé dans cet exemple est **TutorialTestThing**

Important

Vérifiez le nom de votre objet avant de continuer.

Le nom d'un objet ne peut pas être modifié après la création de l'objet. Si vous souhaitez modifier le nom d'un objet, vous devez créer un objet d'objet avec le nom d'un objet, puis supprimer celui qui porte le nom d'un objet approprié, puis supprimer celui qui porte le nom d'un objet approprié, puis supprimer celui qui porte le nom d'un objet, puis supprimer l'objet

Dans la section Configurations supplémentaires, personnalisez davantage votre ressource d'objets à l'aide des configurations facultatives répertoriées.

Après avoir donné un nom à votre objet et sélectionné d'autres configurations, cliquez sur Suivant.

3. Dans la section Choisir la plateforme et le SDK, choisissez la plateforme et la langue du SDK de l'AWS IoT appareil que vous souhaitez utiliser. Cet exemple utilise la plateforme Linux/OSX et le SDK Python. Assurez-vous que python3 et pip3 sont installés sur votre appareil cible avant de passer à l'étape suivante.

Note

Assurez-vous de consulter la liste des logiciels requis par le SDK que vous avez choisi au bas de la page de la console.

Le logiciel requis doit être installé sur votre ordinateur cible avant de passer à l'étape suivante.

Après avoir choisi la langue du SDK de la plate-forme et de l'appareil, choisissez Suivant.

The screenshot shows the 'Choose platform and SDK' step of the AWS IoT 'Connect' wizard. On the left, a sidebar lists steps: Step 1 (Prepare your device), Step 2 (Register and secure your device), Step 3 (Choose platform and SDK), Step 4 (Download connection kit), and Step 5 (Run connection kit). The main area has a title 'Choose platform and SDK' with an 'Info' link. It contains two sections: 'Choose the software for your device' (with a monitor icon) and 'Platform and SDK'. The 'Platform and SDK' section asks to choose the platform OS and AWS IoT Device SDK. Under 'Device platform operating system', 'Linux / macOS' is selected (radio button checked). Under 'AWS IoT Device SDK', 'Python' is selected (radio button checked). At the bottom right are 'Cancel', 'Previous', and 'Next' buttons, with 'Next' being highlighted with a red border.

Étape 3. Télécharger des fichiers sur votre appareil

Cette page s'affiche après AWS IoT la création du kit de connexion, qui inclut 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 politique avec laquelle autoriser votre objet à interagir avec AWS IoT
- Le script permettant de télécharger le SDK de l'AWS appareil et d'exécuter l'exemple de programme sur votre appareil

1. Lorsque vous êtes prêt à continuer, cliquez sur le bouton Télécharger le kit de connexion pour télécharger le kit de connexion pour la plateforme 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 sur votre appareil.

3. Dans la section Décompresser le kit de connexion sur votre appareil, entrez le répertoire unzip connect_device_package.zip dans lequel se trouvent les fichiers du kit de connexion.

Si vous utilisez une fenêtre de PowerShell Windows et que la unzip commande ne fonctionne pas, remplacez-la unzip par expand-archive et réessayez d'utiliser la ligne de commande.

4. Une fois que le fichier du kit de connexion est installé sur l'appareil, poursuivez le didacticiel en choisissant Suivant.

AWS IoT > Connect > Connect one device

Step 1 Prepare your device

Step 2 Register and secure your device

Step 3 Choose platform and SDK

Step 4 Download connection kit

Step 5 Run connection kit

Download connection kit [Info](#)

Install the software on your device

 → 

We created the AWS IoT resources that your device needs to connect to AWS IoT. We also created a connection kit that includes the resources in a zipped file that you need to install on your device. The resources in the connection kit are listed below. In this step, you'll install them on your device.

Connection kit

Certificate TutorialTestThing.cert.pem	Private key TutorialTestThing.private.key	AWS IoT Device SDK Python
Script to send and receive messages start.sh	Policy TutorialTestThing-Policy View policy	

Download

If you are running this from a browser on the device, after you download the connection kit, it will be in the browser's download folder.

If you are not running this from a browser on your device, you'll need to transfer the connection kit from your browser's download folder to your device using the method you tested when you prepared your device in step 1.

[Download connection kit](#)

Unzip connection kit on your device

 After the connection kit is on your device, unzip it using this command:

`unzip connect_device_package.zip`

[Copy](#)

[Cancel](#) [Previous](#) [Next](#)

Étape 4. Exécutez l'exemple

Vous effectuez cette procédure dans un terminal ou dans la fenêtre de commande de votre appareil tout en suivant les instructions affichées dans la console. Les commandes affichées dans la console concernent le système d'exploitation que vous avez choisi [the section called "Étape 2. Création d'un objet" \(p. 25\)](#). Celles présentées ici concernent les systèmes d'exploitation Linux/OSX.

1. Dans un terminal ou une fenêtre de commande de votre appareil, dans le répertoire contenant le fichier du kit de connexion, effectuez les étapes indiquées dans la AWS IoT console.

AWS IoT > Connect > Connect one device

Step 1 Prepare your device

Step 2 Register and secure your device

Step 3 Choose platform and SDK

Step 4 Download connection kit

Step 5 Run connection kit

Run connection kit Info

How to display messages from your device

Step 1: Add execution permissions
On the device, launch a terminal window to copy and paste the command to add execution permissions.

`chmod +x start.sh`



Step 2: Run the start script
On the device, copy and paste the command to the terminal window and run the start script.

`./start.sh`

Step 3: Return to this screen to view your device's messages
After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.

Subscriptions	sdk/test/Python	Pause	Clear
sdk/test/Python	Waiting for messages		

- Après avoir saisi la commande de l'étape 2 dans la console, vous devriez voir une sortie similaire à la suivante dans le terminal ou la fenêtre de commande de l'appareil. Cette sortie provient des messages auxquels le programme envoie puis reçoit en retour AWS IoT Core.

```
Running pub/sub sample application...
Connecting to a13hikvzkye6lx-ats.iot.us-east-1.amazonaws.com with client ID 'basicPubSub'...
Connected!
Subscribing to topic 'sdk/test/Python'...
Subscribed with QoS.AT LEAST_ONCE
Sending messages until program killed
Publishing message to topic 'sdk/test/Python': Hello World! [1]
Received message from topic 'sdk/test/Python': b'"Hello World! [1]"'
Publishing message to topic 'sdk/test/Python': Hello World! [2]
Received message from topic 'sdk/test/Python': b'"Hello World! [2]"'
Publishing message to topic 'sdk/test/Python': Hello World! [3]
Received message from topic 'sdk/test/Python': b'"Hello World! [3]"'
```

Pendant que l'exemple de programme est en cours d'exécution, le message de test s'Hello World ! affiche également. Le message de test s'affiche dans le terminal ou dans la fenêtre de commande de votre appareil.

Note

Pour plus d'informations sur l'abonnement et la publication d'une rubrique, consultez l'exemple de code du SDK que vous avez choisi.

3. Pour réexécuter le programme d'exemple, vous pouvez répéter les commandes de l'étape 2 dans la console de cette procédure.
4. (Facultatif) Si vous souhaitez voir les messages de votre client IoT dans la [AWS IoTconsole](#), ouvrez le [client de test MQTT](#) sur la page Test de la AWS IoT console. Si vous avez choisi le SDK Python, dans le client de test MQTT, dans le filtre Rubrique, entrez la rubrique, par exemple **sdk/test/python** pour vous abonner aux messages depuis votre appareil. Les filtres thématiques distinguent les majuscules et minuscules et dépendent du langage de programmation du SDK que vous avez choisi à l'étape 1. Pour plus d'informations sur l'abonnement et la publication d'une rubrique, consultez l'exemple de code du SDK que vous avez choisi.
5. Une fois que vous êtes abonné au sujet de test, exécutez-le `./start.sh` sur votre appareil. Pour plus d'informations, veuillez consulter [the section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” \(p. 71\)](#).

Après l'exécution `./start.sh`, des messages similaires aux suivants apparaissent dans le client MQTT :

```
{  
    "message": "Hello World!" [1]  
}
```

Le sequence nombre est [] incrémenté d'une unité chaque fois qu'un nouveau Hello World! message est reçu et s'arrête lorsque vous mettez fin au programme.

6. Pour terminer le didacticiel et voir un résumé, dans la AWS IoT console, choisissez Continuer.

AWS IoT Core Guide du développeur

Étape 4. Exécutez l'exemple

AWS IoT > Connect > Connect one device

Step 1
Prepare your device

Step 2
Register and secure your device

Step 3
Choose platform and SDK

Step 4
Download connection kit

Step 5
Run connection kit

Run connection kit Info

How to display messages from your device

Step 1: Add execution permissions
On the device, launch a terminal window to copy and paste the command to add execution permissions.

```
chmod +x start.sh
```

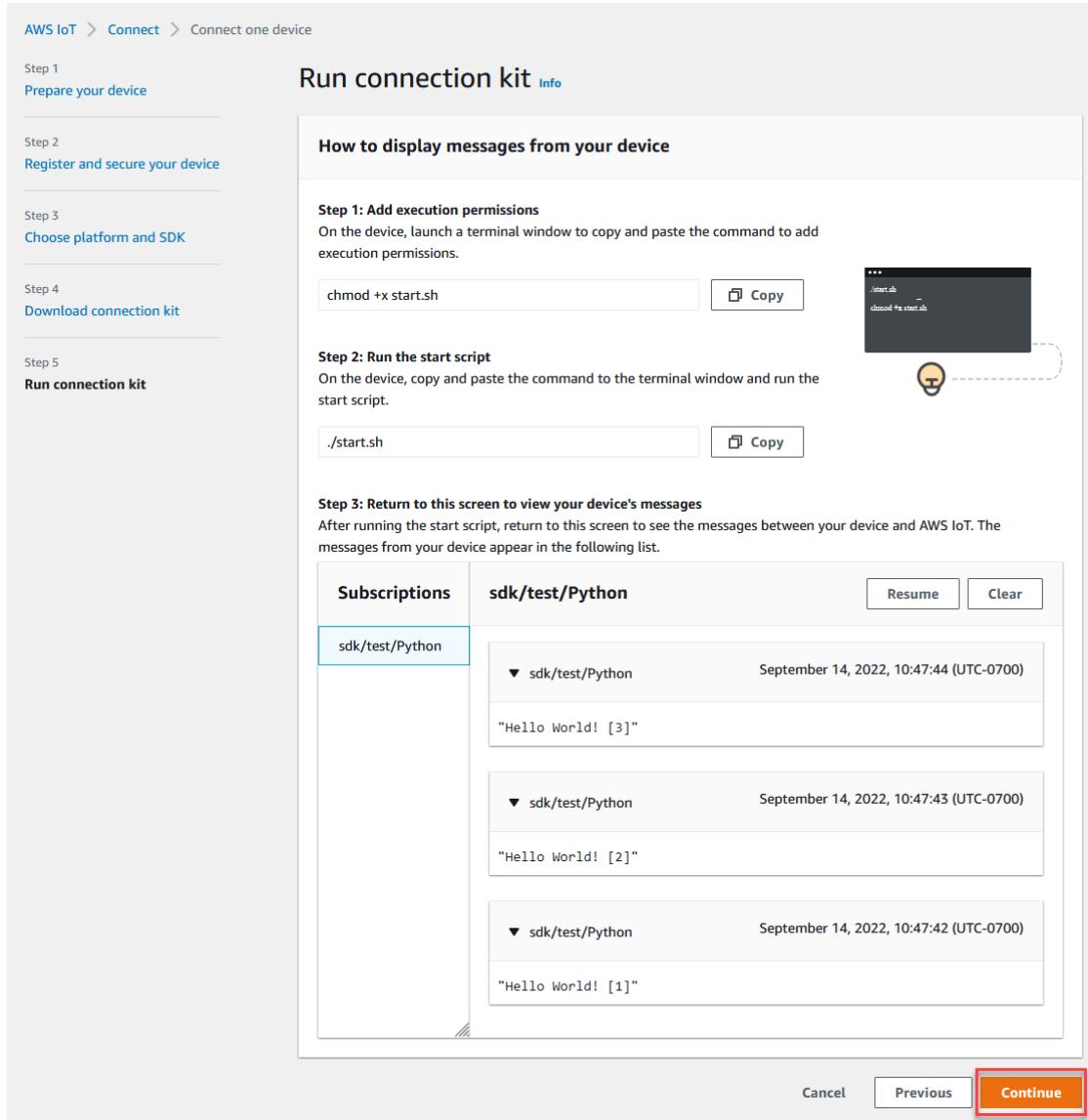
Step 2: Run the start script
On the device, copy and paste the command to the terminal window and run the start script.

```
./start.sh
```

Step 3: Return to this screen to view your device's messages
After running the start script, return to this screen to see the messages between your device and AWS IoT. The messages from your device appear in the following list.

Subscriptions	sdk/test/Python	Resume	Clear
sdk/test/Python	<p>▼ sdk/test/Python September 14, 2022, 10:47:44 (UTC-0700)</p> <p>"Hello World! [3]"</p> <p>▼ sdk/test/Python September 14, 2022, 10:47:43 (UTC-0700)</p> <p>"Hello World! [2]"</p> <p>▼ sdk/test/Python September 14, 2022, 10:47:42 (UTC-0700)</p> <p>"Hello World! [1]"</p>		

Cancel Previous Continue



7. Un résumé de votre didacticiel de connexion AWS IoT rapide s'affiche désormais.

AWS IoT > Connect > Connect one device > Next steps

Device is connected

Your device is now connected. There are many services you can explore.

Device connected to AWS IoT Explore AWS services

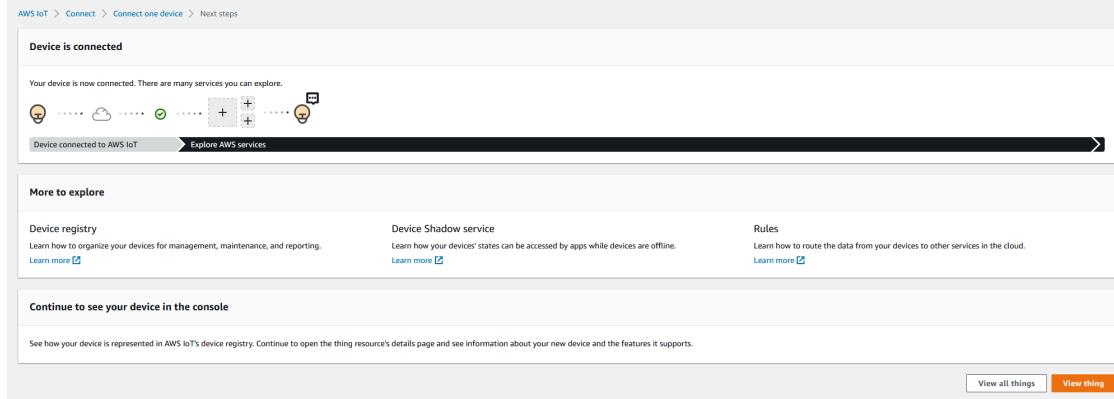
More to explore

Device registry	Device Shadow service	Rules
Learn how to organize your devices for management, maintenance, and reporting. Learn more	Learn how your devices' states can be accessed by apps while devices are offline. Learn more	Learn how to route the data from your devices to other services in the cloud. Learn more

Continue to see your device in the console

See how your device is represented in AWS IoT's device registry. Continue to open the thing resource's details page and see information about your new device and the features it supports.

View all things View thing



Étape 5. Explorez plus loin

Voici quelques idées à AWS IoT approfondir une fois que vous aurez terminé le démarrage rapide.

- [Afficher les messages MQTT dans le client de test MQTT](#)

Depuis la [AWS IoTconsole](#), vous pouvez ouvrir le [client MQTT](#) sur la page Test de la AWS IoT console. Dans le client de test MQTT#, abonnez-vous à, puis exécutez le programme sur votre appareil ./start.sh comme décrit à l'étape précédente. Pour plus d'informations, veuillez consulter [the section called "Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT" \(p. 71\)](#).

- Exécutez des tests sur vos appareils avec [Device Advisor](#)

Utilisez Device Advisor pour vérifier si vos appareils peuvent se connecter et interagir avec AWS IoT.

- [the section called "Essayez le didacticiel AWS IoT Core interactif" \(p. 21\)](#)

Pour démarrer le didacticiel interactif, sur la page Apprendre de la AWS IoT console, dans la vignette Voir comment AWS IoT fonctionne, choisissez Démarrer le didacticiel.

- [Préparez-vous à découvrir d'autres didacticiels \(p. 38\)](#)

Ce démarrage rapide ne vous donne qu'un échantillon de AWS IoT. Si vous souhaitez en savoir AWS IoT plus et découvrir les fonctionnalités qui en font une puissante plateforme de solutions IoT, commencez à préparer votre plateforme de développement en[Découvrez les AWS IoT Core services dans le cadre d'un didacticiel pratique \(p. 38\)](#).

Tester la connectivité avec le point de terminaison de données de votre appareil

Cette rubrique explique comment tester la connexion d'un appareil avec le point de terminaison de données de votre compte, le point de terminaison auquel vos appareils IoT se connectent AWS IoT.

Effectuez ces procédures sur l'appareil que vous souhaitez tester ou en utilisant une session de terminal SSH connectée à l'appareil que vous souhaitez tester.

Pour tester la connectivité d'un appareil avec le point de terminaison de données de votre appareil.

- [Trouvez le point de terminaison des données de votre appareil \(p. 34\)](#)
- [Tester la connexion rapidement \(p. 35\)](#)
- [Téléchargez l'application pour tester la connexion au point de terminaison de données et au port de votre appareil \(p. 35\)](#)
- [Testez la connexion au terminal de données et au port de votre appareil \(p. 38\)](#)

Trouvez le point de terminaison des données de votre appareil

Pour rechercher le point de terminaison de votre appareil

1. Dans la [AWS IoTconsole](#), en bas du volet de navigation, choisissez Paramètres.
2. Sur la page Paramètres, dans le conteneur de points de terminaison de données de l'appareil, localisez la valeur du point de terminaison et copiez-la. La valeur de votre point de terminaison est unique Compte AWS et est similaire à cet exemple :a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com.
3. Enregistrez le point de terminaison des données de votre appareil pour l'utiliser dans les procédures suivantes.

Tester la connexion rapidement

Cette procédure teste la connectivité générale avec le point de terminaison de données de votre appareil, mais elle ne teste pas le port spécifique que vos appareils utiliseront. Ce test utilise un programme courant et est généralement suffisant pour déterminer si vos appareils peuvent se connecter à AWS IoT.

Si vous souhaitez tester la connectivité avec le port spécifique que vos appareils utiliseront, ignorez cette procédure et continuez [Téléchargez l'application pour tester la connexion au point de terminaison de données et au port de votre appareil \(p. 35\)](#).

Pour tester rapidement le point de terminaison des données de l'appareil

1. Dans une fenêtre de terminal ou de ligne de commande de votre appareil, remplacez l'exemple de point de terminaison de données de l'appareil (`a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com`) par le point de terminaison de données de l'appareil de votre compte, puis entrez cette commande.

Linux

```
ping -c 5 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
```

Windows

```
ping -n 5 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
```

2. S'il ping affiche une sortie similaire à la suivante, il s'est connecté correctement au point de terminaison de données de votre appareil. Bien qu'il n'ait pas communiqué AWS IoT directement avec le serveur, il a trouvé le serveur et il AWS IoT est probable qu'il soit disponible via ce point de terminaison.

```
PING a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (xx.xx.xxx.xxx) 56(84) bytes of data.  
64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):  
icmp_seq=1 ttl=231 time=127 ms  
64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):  
icmp_seq=2 ttl=231 time=127 ms  
64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):  
icmp_seq=3 ttl=231 time=127 ms  
64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):  
icmp_seq=4 ttl=231 time=127 ms  
64 bytes from ec2-EXAMPLE-218.eu-west-1.compute.amazonaws.com (xx.xx.xxx.xxx):  
icmp_seq=5 ttl=231 time=127 ms
```

Si vous êtes satisfait de ce résultat, vous pouvez arrêter les tests ici.

Si vous souhaitez tester la connectivité avec le port spécifique utilisé par AWS IoT, passez à [Téléchargez l'application pour tester la connexion au point de terminaison de données et au port de votre appareil \(p. 35\)](#).

3. Si ping aucun résultat n'a été renvoyé, vérifiez la valeur du point de terminaison pour vous assurer que vous disposez du bon point de terminaison et vérifiez la connexion de l'appareil à Internet.

Téléchargez l'application pour tester la connexion au point de terminaison de données et au port de votre appareil

Un test de connectivité plus approfondi peut être effectué à l'aide de nmap. Cette procédure permet de vérifier s'il nmap est installé sur votre appareil.

À vérifier nmap sur l'appareil

1. Dans un terminal ou une fenêtre de ligne de commande de l'appareil que vous souhaitez tester, entrez cette commande pour voir si elle nmap est installée.

```
nmap --version
```

2. Si vous voyez une sortie similaire à ce qui suit, nmap est installé et vous pouvez continuer à [the section called “Testez la connexion au terminal de données et au port de votre appareil” \(p. 38\)](#).

```
Nmap version 6.40 ( http://nmap.org )
Platform: x86_64-koji-linux-gnu
Compiled with: nmap-liblua-5.2.2 openssl-1.0.2k libpcre-8.32 libpcap-1.5.3 nmap-
libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

3. Si vous ne voyez pas de réponse similaire à celle indiquée à l'étape précédente, vous devez procéder à l'installation sur l'appareil, vous devez procéder à l'installation nmap sur l'appareil. Choisissez la procédure pour le système d'exploitation de votre appareil.

Linux

Cette procédure requiert que vous ayez l'autorisation d'installer le logiciel sur l'ordinateur.

Pour installer nmap sur votre ordinateur Linux

1. Dans une fenêtre de terminal ou de ligne de commande de votre appareil, entrez la commande correspondant à la version de Linux qu'il exécute.

- a. Debian ou Ubuntu :

```
sudo apt install nmap
```

- b. CentOS ou RHEL :

```
sudo yum install nmap
```

2. Testez l'installation à l'aide de cette commande :

```
nmap --version
```

3. Si vous voyez une sortie similaire à ce qui suit, nmap est installé et vous pouvez continuer à [the section called “Testez la connexion au terminal de données et au port de votre appareil” \(p. 38\)](#).

```
Nmap version 6.40 ( http://nmap.org )
Platform: x86_64-koji-linux-gnu
Compiled with: nmap-liblua-5.2.2 openssl-1.0.2k libpcre-8.32 libpcap-1.5.3 nmap-
libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

macOS

Cette procédure requiert que vous ayez l'autorisation d'installer le logiciel sur l'ordinateur.

Pour installer nmap sur votre ordinateur macOS

1. Dans un navigateur, ouvrez <https://nmap.org/download#macosx> et téléchargez le dernier programme d'installation stable.

Lorsque vous y êtes invité, sélectionnez Ouvrir avec DiskImageInstaller.

2. Dans la fenêtre d'installation, déplacez le package vers le dossier Applications.
3. Dans le Finder, recherchez le nmap-xxxx-mpkg package dans le dossier Applications. Ctrl-cliquez package activé et sélectionnez Ouvrir pour ouvrir le package.
4. Vérifiez la boîte de dialogue de sécurité. Si vous êtes prêt à procéder à l'installation nmap, choisissez Ouvrir pour installernmap.
5. Dans Terminal, testez l'installation à l'aide de cette commande.

```
nmap --version
```

6. Si vous voyez une sortie similaire à ce qui suit, nmap est installé et vous pouvez continuer à [the section called "Testez la connexion au terminal de données et au port de votre appareil" \(p. 38\)](#).

```
Nmap version 7.92 ( https://nmap.org )
Platform: x86_64-apple-darwin17.7.0
Compiled with: nmap-liblua-5.3.5 openssl-1.1.1k nmap-libssh2-1.9.0 libz-1.2.11
    nmap-libpcre-7.6 nmap-libpcap-1.9.1 nmap-libdnet-1.12 ipv6 Compiled without:
Available nsock engines: kqueue poll select
```

Windows

Cette procédure requiert que vous ayez l'autorisation d'installer le logiciel sur l'ordinateur.

Pour installer nmap sur votre ordinateur Windows

1. Dans un navigateur, ouvrez <https://nmap.org/download#windows> et téléchargez la dernière version stable du programme d'installation.

Si vous y êtes invité, choisissez Enregistrer le fichier. Une fois le fichier téléchargé, ouvrez-le depuis le dossier des téléchargements.

2. Une fois le téléchargement du fichier d'installation terminé, ouvrez la nmap-xxxx-setup.exe fenêtre téléchargée pour installer l'application.
3. Acceptez les paramètres par défaut lors de l'installation du programme.

Vous n'avez pas besoin de l'application Npcap pour ce test. Vous pouvez désélectionner cette option si vous ne souhaitez pas l'installer.

4. Dans Command, testez l'installation à l'aide de cette commande.

```
nmap --version
```

5. Si vous voyez une sortie similaire à ce qui suit, nmap est installé et vous pouvez continuer à [the section called "Testez la connexion au terminal de données et au port de votre appareil" \(p. 38\)](#).

```
Nmap version 7.92 ( https://nmap.org )
Platform: i686-pc-windows-windows
Compiled with: nmap-liblua-5.3.5 openssl-1.1.1k nmap-libssh2-1.9.0 nmap-libz-1.2.11
    nmap-libpcre-7.6 Npcap-1.50 nmap-libdnet-1.12 ipv6
Compiled without:
```

Available nsock engines: iocp poll select

Testez la connexion au terminal de données et au port de votre appareil

Pour tester le point de terminaison et le port de données de votre appareil

1. Dans une fenêtre de terminal ou de ligne de commande de votre appareil, remplacez l'exemple de point de terminaison de données de l'appareil (a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com) par le point de terminaison de données de l'appareil de votre compte, puis entrez cette commande.

```
nmap -p 8443 a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com
```

2. Il nmap affiche une sortie similaire à la suivante, nmap a réussi à se connecter au point de terminaison de données de votre appareil sur le port sélectionné.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-18 16:23 Pacific Standard Time
Nmap scan report for a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (xx.xxx.147.160)
Host is up (0.036s latency).
Other addresses for a3qEXAMPLEsffp-ats.iot.eu-west-1.amazonaws.com (not scanned):
  xx.xxx.134.144 xx.xxx.55.139 xx.xxx.110.235 xx.xxx.174.233 xx.xxx.74.65 xx.xxx.122.179
  xx.xxx.127.126
rDNS record for xx.xxx.147.160: ec2-EXAMPLE-160.eu-west-1.compute.amazonaws.com

PORT      STATE SERVICE
8443/tcp  open  https-alt
MAC Address: 00:11:22:33:44:55 (Cimsys)

Nmap done: 1 IP address (1 host up) scanned in 0.91 seconds
```

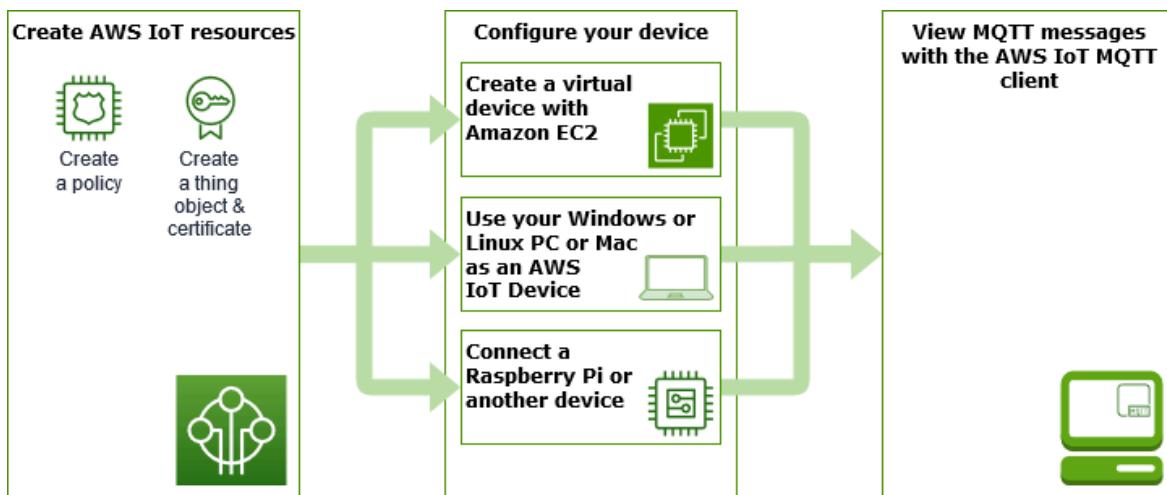
3. Si nmap aucun résultat n'a été renvoyé, vérifiez la valeur du point de terminaison pour vous assurer que vous disposez du bon point de terminaison et vérifiez la connexion de votre appareil à Internet.

Vous pouvez tester d'autres ports sur le point de terminaison de données de votre appareil, tels que le port 443, le port HTTPS principal, **8443** en remplaçant le port utilisé à l'étape 1 par le port que vous souhaitez tester.

Découvrez les AWS IoT Core services dans le cadre d'un didacticiel pratique

Dans ce didacticiel, vous allez installer le logiciel et créer les AWS IoT ressources nécessaires pour connecter un appareil AWS IoT Core afin qu'il puisse envoyer et recevoir des messages MQTT avec AWS IoT Core. Vous verrez les messages du client MQTT dans la AWS IoT console.

Vous pouvez vous attendre à consacrer 20 à 30 minutes à ce didacticiel. Si vous utilisez un appareil 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 didacticiel est idéal pour les développeurs qui AWS IoT Core souhaitent commencer à explorer des fonctionnalités plus avancées, telles que le [moteur de règles](#) et les [ombres](#). Ce didacticiel vous permet de continuer à en apprendre davantage sur AWS IoT Core les autres AWS services et sur la manière dont ils interagissent avec eux en expliquant les étapes de manière plus détaillée que dans [le didacticiel de démarrage rapide \(p. 23\)](#). Si vous recherchez une expérience rapide de Hello World, essayez le [Essayez la connexion AWS IoT rapide \(p. 23\)](#).

Après avoir configuré votre AWS IoT console Compte AWS et, vous allez suivre ces étapes pour savoir comment connecter un appareil et lui faire envoyer des messagesAWS IoT Core.

Étapes suivantes

- [Choisissez l'option d'appareil qui vous convient le mieux \(p. 39\)](#)
- [the section called “Création de AWS IoT ressources” \(p. 40\)](#) si vous n'avez pas l'intention de créer un appareil virtuel avec Amazon EC2
- [the section called “Configurer votre appareil” \(p. 43\)](#)
- [the section called “Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT” \(p. 71\)](#)

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 peut être une bonne option si :	Ce n'est peut-être pas une bonne option si :
the section called “Création d'un appareil virtuel avec Amazon EC2” (p. 44)	<ul style="list-style-type: none"> Vous n'avez pas votre propre appareil à tester. Vous ne voulez installer aucun logiciel sur votre propre système. Vous souhaitez effectuer un test sur un système d'exploitation Linux. 	<ul style="list-style-type: none"> Vous n'êtes pas à l'aise avec les commandes de ligne de commande. Vous ne souhaitez pas exposer votre à des AWS frais supplémentaires. Vous ne souhaitez pas effectuer de test sur un système d'exploitation Linux.

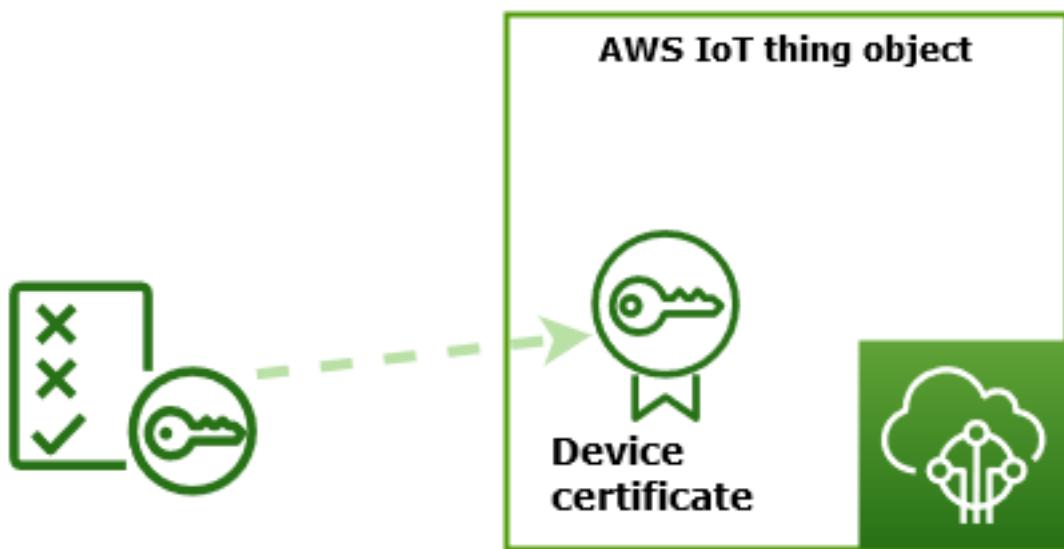
Option	Cela peut être une bonne option si :	Ce n'est peut-être pas une bonne option si :
<u>the section called "Utilisez votre PC ou Mac Windows ou Linux comme AWS IoT appareil" (p. 52)</u>	<ul style="list-style-type: none"> • Vous ne souhaitez pas exposer votre à des AWS frais supplémentaires. • Vous ne souhaitez pas configurer d'appareils supplémentaires. 	<ul style="list-style-type: none"> • Vous ne souhaitez pas installer de logiciel sur votre ordinateur personnel. • Vous souhaitez une plateforme de test plus représentative.
<u>the section called "Connect un Raspberry Pi ou un autre appareil" (p. 61)</u>	<ul style="list-style-type: none"> • Vous souhaitez effectuer un test AWS IoT avec un appareil réel. • Vous disposez déjà d'un appareil à tester. • Vous avez de l'expérience en matière d'intégration de matériel dans des systèmes. 	<ul style="list-style-type: none"> • Vous ne voulez pas acheter ou configurer un appareil juste pour l'essayer. • Vous souhaitez tester AWS IoT le plus simplement possible, pour l'instant.

Création de AWS IoT ressources

Dans ce didacticiel, vous allez créer les AWS IoT ressources dont un appareil a besoin pour se connecter à des messages AWS IoT Core et les échanger.

Create an AWS IoT Core policy

Create a thing and its certificate



1. Créez un document AWS IoT de politique qui autorisera votre appareil à interagir avec les AWS IoT services.
2. Créez un objet AWS IoT et son certificat de périphérique X.509, puis joignez le document de politique. L'objet objet est la représentation virtuelle de votre appareil dans le AWS IoT registre. Le certificat authentifie votre appareil et le document de politique autorise votre appareil à interagir avec AWS IoT Core AWS IoT

Note

Si vous avez l'intention de [the section called “Création d'un appareil virtuel avec Amazon EC2” \(p. 44\)](#) le faire, vous pouvez ignorer cette page et continuer [the section called “Configurer votre appareil” \(p. 43\)](#). Vous créerez ces ressources lorsque vous créerez votre objet virtuel.

Ce didacticiel utilise la AWS IoT console pour créer les AWS IoT ressources. 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 devrez copier les fichiers de certificat sur votre appareil avant de pouvoir les utiliser par l'exemple d'application.

Création d'une stratégie AWS IoT

Les appareils utilisent un certificat X.509 pour s'authentifier auprès d'. AWS IoT Core Des AWS IoT politiques sont associées au certificat. Ces politiques déterminent quelles AWS IoT opérations, telles que l'abonnement ou la publication à des sujets MQTT, l'appareil est autorisé à effectuer. Votre appareil présente son certificat lorsqu'il se connecte et envoie des messages àAWS IoT Core.

Suivez les étapes pour créer une politique qui autorise votre appareil à effectuer les AWS IoT opérations nécessaires à l'exécution du programme d'exemple. Vous devez créer la AWS IoT politique avant de pouvoir l'associer au certificat de l'appareil, que vous créerez ultérieurement.

Pour créer une stratégie AWS IoT

1. Dans le menu de gauche, choisissez Sécurité, puis Politiques. Sur la page Vous n'avez pas encore de politique, choisissez Créer une politique.

Si des politiques existent déjà sur votre compte, choisissez Créeer.

2. Sur la page Créeer une politique :

1. Dans la section Propriétés de la stratégie, dans le champ Nom de la stratégie, entrez le nom de la stratégie (par exemple, **My_Iot_Policy**). N'utilisez pas d'informations personnelles identifiables dans les noms de vos politiques.
2. Dans la section Document de politique, créez les déclarations de politique qui accordent ou refusent aux ressources l'accès aux AWS IoT Core opérations. Pour créer une déclaration de politique autorisant tous les clients à exécuter des tâches **iot:Connect**, procédez comme suit :
 - Dans le champ Effet de la politique, sélectionnez Autoriser. Cela permet à tous les clients dont cette politique est associée à leur certificat d'effectuer l'action répertoriée dans le champ Action de stratégie.
 - Dans le champ Action de stratégie, choisissez une action de stratégie telle que **iot:Connect**. Les actions de politique sont les actions que votre appareil doit être autorisé à effectuer lorsqu'il exécute le programme d'exemple à partir du SDK de l'appareil.
 - Dans le champ Ressource de politique, entrez une ressource Amazon Resource Name (ARN) ou*. A * pour sélectionner n'importe quel client (appareil).

Pour créer les déclarations de politique pour **iot:Receive**, **iot:Publish**, et **iot:Subscribe**, choisissez Ajouter une nouvelle déclaration et répétez les étapes.

Policy effect	Policy action	Policy resource	
Allow	iot:Connect	*	Remove
Allow	iot:Receive	*	Remove
Allow	iot:Publish	*	Remove
Allow	iot:Subscribe	*	Remove

Note

Dans ce démarrage rapide, le caractère générique (*) est utilisé pour des raisons de simplicité. Pour une sécurité accrue, vous devez limiter les clients (appareils) autorisés à se connecter et à publier des messages en spécifiant un ARN client au lieu du caractère générique comme ressource. Les ARN des clients suivent le format suivant :arn:aws:iot:*your-region*:*your-aws-account*:client/*my-client-id*. Toutefois, vous devez d'abord créer la ressource (comme un appareil client ou un objet fantôme) avant de pouvoir attribuer son ARN à une politique. Pour d'd'd'd'd'd'd'[AWS IoT Core](#)d'd'd'd'd'd'd'

- Après avoir saisi les informations relatives à votre politique, choisissez Créer.

Pour plus d'informations, veuillez consulter [Fonctionnement de AWS IoT avec IAM \(p. 419\)](#).

Création d'un objet

Les appareils auxquels AWS IoT Core ils sont connectés sont représentés par des objets dans le AWS IoT registre. Un objet représente un périphérique ou une entité logique spécifique. Il peut s'agir d'un appareil physique ou d'un capteur (par exemple, une ampoule ou un interrupteur mural). Il peut également s'agir d'une entité logique, telle qu'une instance d'une application ou d'une entité physique qui ne se connecte pas à d'autres appareils qui y sont connectés (par exemple, une voiture équipée de capteurs moteur ou d'un panneau de commande). AWS IoT

Pour créer un objet dans la AWS IoT console

- Dans la [AWS IoTconsole](#), dans le menu de gauche, choisissez Gérer, puis Objets.
- Sur la page Objets, choisissez Créer des objets.
- Sur la page Crée des objets, choisissez Crée un objet unique, puis choisissez Suivant.
- Sur la page Spécifier les propriétés de l'objet, dans Nom de l'objet, entrez le nom de votre objet, tel que **MyIoTThing**.

Choisissez les noms des objets avec soin, car vous ne pourrez pas les modifier ultérieurement.

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 le nom de votre objet. Le nom de l'objet peut apparaître dans des communications et des rapports non chiffrés.

- Laissez les autres champs de cette page vides. Choisissez Suivant.
- Sur la page Configurer le certificat de l'appareil - facultatif, choisissez Générer automatiquement un nouveau certificat (recommandé). Choisissez Suivant.
- Sur la page facultative Joindre des politiques au certificat, sélectionnez la politique que vous avez créée dans la section précédente. Dans cette section, la politique a été nommée, **My_Iot_Policy**. Choisissez Crée un objet.

8. Sur la page Télécharger les certificats et les clés :

1. Téléchargez chaque certificat et chaque fichier clé et enregistrez-les pour plus tard. Vous devez installer ces fichiers sur votre appareil.

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

Noms des fichiers de certificat

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

2. Pour télécharger le fichier d'autorité de certification racine correspondant à ces fichiers, cliquez sur le lien de téléchargement du fichier de certificat de l'autorité de certification racine correspondant au type de point de terminaison de données et de suite de chiffrement que vous utilisez. Dans ce didacticiel, choisissez Télécharger à droite de la clé RSA 2048 bits : Amazon Root CA 1 et téléchargez la clé RSA 2048 bits : fichier de certificat Amazon Root CA 1.

Important

Vous devez enregistrer les fichiers de certificat avant de quitter cette page. Une fois que vous aurez 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 que vous avez créé, puis recommencer cette procédure depuis le début.

3. Sélectionnez Done (Exécuté).

Une fois cette procédure terminée, vous devriez voir apparaître le nouvel objet dans votre liste d'objets.

Configurer votre appareil

Cette section décrit comment configurer votre appareil pour vous connecter à AWS IoT Core. Si vous souhaitez commencer AWS IoT Core mais que vous ne possédez pas encore d'appareil, vous pouvez créer un appareil virtuel à l'aide d'Amazon EC2 ou vous pouvez utiliser votre PC Windows ou Mac comme appareil IoT.

Sélectionnez l'option d'appareil qui vous convient le mieux AWS IoT Core. Bien sûr, vous pouvez tous les essayer, mais n'en essayez qu'un à la fois. Si vous ne savez pas quelle option d'appareil vous convient le mieux, découvrez comment choisir l'[option d'appareil qui vous convient le mieux \(p. 39\)](#), puis revenez à cette page.

Options de l'appareil

- [Création d'un appareil virtuel avec Amazon EC2 \(p. 44\)](#)
- [Utilisez votre PC ou Mac Windows ou Linux comme AWS IoT appareil \(p. 52\)](#)
- [Connect un Raspberry Pi ou un autre appareil \(p. 61\)](#)

Création d'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 effectuer ce didacticiel, vous avez besoin d'un Compte AWS. Si vous n'en avez pas, suivez les étapes décrites dans [Configurez votre Compte AWS \(p. 19\)](#) avant de continuer.

Dans le cadre de ce didacticiel, vous effectuerez les tâches suivantes :

- [Configuration d'une instance Amazon EC2 \(p. 44\)](#)
- [Installez Git, Node.js et configurez le AWS CLI \(p. 45\)](#)
- [Créez AWS IoT des ressources pour votre appareil virtuel \(p. 46\)](#)
- [Installez le SDK de AWS IoT l'appareil pour JavaScript \(p. 50\)](#)
- [Exécution de l'exemple d'application \(p. 50\)](#)
- [Afficher les messages de l'exemple d'application dans la AWS IoT console \(p. 51\)](#)

Configuration d'une instance Amazon EC2

Les étapes suivantes vous montrent comment créer une instance Amazon EC2 qui agira comme votre appareil virtuel à la place d'un appareil physique.

Si c'est la première fois que vous créez une instance Amazon EC2, les instructions de la section [Commencer avec les instances Amazon EC2Linux](#) peuvent vous être plus utiles.

Pour lancer une instance

1. Ouvrez la console Amazon EC2 à l'adresse <https://console.aws.amazon.com/ec2/>.
2. Dans le menu de la console sur la gauche, développez la section Instances et choisissez Instances. Dans le tableau de bord des instances, choisissez Lancer les instances sur la droite pour afficher la liste des configurations de base.
3. Dans la section Nom et balises, entrez le nom de l'instance et ajoutez éventuellement des balises.
4. Dans la section Application et images (Image du système d'exploitation (Amazon Machine)), choisissez un modèle d'AMI pour votre instance, par exemple, par exemple Amazon Linux 2 (HVM). Notez que cette AMI est indiquée comme « Éligible à l'offre gratuite ».
5. Dans la section 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.
6. Dans la section Key (Nom), choisissez un nom de la paire (Nom), choisissez un nom de la key pair (Nom), choisissez un nom de la paire (Nom), choisissez un nom de la paire (Nom) pour en créer une. Lorsque vous créez une nouvelle key pair, veillez à télécharger le fichier de clé privée et à l'enregistrer en lieu sûr, car c'est votre seule chance de le télécharger et de l'enregistrer. 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 choisissez pas l'option Poursuivre sans paire de la key pair de paire. Si vous lancez votre instance sans une paire de clés, vous ne pourrez pas vous y connecter.

7. Dans la section Paramètres réseau et la section Configurer le stockage, vous pouvez conserver les paramètres par défaut. Une fois que vous êtes prêt, choisissez Lancer 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 avec succès ses contrôles 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, choisissez Instances.
3. Sélectionnez l'instance, puis sélectionnez Connect (Connexion).
4. Choisissez Amazon EC2 Instance Connect, Connect.

Vous devriez maintenant avoir une fenêtre Amazon EC2 Instance Connect connectée à votre nouvelle instance Amazon EC2.

Installez Git, Node.js et configurez le AWS CLI

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

Pour installer Git

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

```
sudo yum update -y
```

2. Dans votre fenêtre Amazon EC2 Instance Connect, installez Git à l'aide de la commande suivante.

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

3. Pour vérifier si Git a été installé ainsi que la version actuelle de Git, exécutez la commande suivante :

```
git --version
```

Pour installer Node.js

1. Dans votre fenêtre Amazon EC2 Instance Connect, installez le gestionnaire de versions des nœuds (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 votre fenêtre Amazon EC2 Instance Connect, activez nvm à l'aide de cette commande.

```
. ~/.nvm/nvm.sh
```

3. Dans votre fenêtre Amazon EC2 Instance Connect, utilisez nvm pour installer la dernière version de Node.js à l'aide de cette commande.

```
nvm install 16
```

Note

Cela installe la dernière version LTS de Node.js.

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 votre fenêtre Amazon EC2 Instance Connect, vérifiez que Node.js est installé et fonctionne correctement à l'aide de cette commande.

```
node -e "console.log('Running Node.js ' + process.version)"
```

Ce didacticiel nécessite Node v10.0 ou une version ultérieure. Pour plus d'informations, consultez le [didacticiel : Configuration de Node.js sur une instance Amazon EC2](#).

Pour configurer AWS CLI

Votre instance Amazon EC2 est préchargée avec le AWS CLI Cependant, vous devez compléter votre AWS CLI profil. Pour plus d'informations sur la configuration de votre interface de ligne de commande, consultez la section [Configuration du AWS CLI](#).

1. L'exemple suivant montre des exemples de valeurs. Remplacez les par vos propres valeurs. Vous pouvez trouver ces valeurs sur votre [AWSconsole, dans les informations de votre compte sous Informations de sécurité](#).

Dans votre fenêtre Amazon EC2 Instance Connect, entrez cette commande :

```
aws configure
```

Entrez ensuite les valeurs de votre compte en suivant les instructions qui s'affichent.

```
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 votre AWS CLI configuration à l'aide de cette commande :

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

Si votre AWS CLI est correctement configuré, la commande doit renvoyer une adresse de point de terminaison provenant de votreCompte AWS.

Créez AWS IoT des ressources pour votre appareil virtuel

Cette section explique comment utiliser le AWS CLI pour créer l'objet objet et ses fichiers de certificat directement sur le périphérique virtuel. Cela se fait directement sur l'appareil pour éviter les complications

potentielles qui pourraient découler de leur copie sur l'appareil à partir d'un autre ordinateur. Dans cette section, vous allez créer les ressources suivantes pour votre appareil virtuel :

- Un objet dans lequel représenter votre appareil virtuel AWS IoT.
- Un certificat pour authentifier votre appareil virtuel.
- Un document de politique autorisant votre appareil virtuel à se connecter à des messages AWS IoT, à les publier, à les recevoir et à s'y abonner.

Pour créer un AWS IoT objet dans votre instance Linux

Les appareils auxquels AWS IoT ils sont connectés sont représentés par des objets dans le AWS IoT registre. Un objet représente un périphérique ou une entité logique spécifique. Dans ce cas, votre objet représentera votre appareil virtuel, cette instance Amazon EC2.

1. Dans votre fenêtre Amazon EC2 Instance Connect, exécutez la commande suivante pour créer votre objet d'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 joindre AWS IoT des clés et des certificats dans votre instance Linux

La [create-keys-and-certificate](#) commande crée 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 votre fenêtre Amazon EC2 Instance Connect, créez un répertoire pour stocker vos fichiers de certificat et de clé.

```
mkdir ~/certs
```

2. Dans votre fenêtre Amazon EC2 Instance Connect, téléchargez une copie du certificat de l'autorité de certification Amazon (CA) à l'aide de cette commande.

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

3. Dans votre fenêtre Amazon EC2 Instance Connect, exécutez la commande suivante pour créer votre clé privée, votre clé publique et vos fichiers de certificat X.509. Cette commande enregistre et active également le certificat avec AWS 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 le `certificateArn` pour pouvoir l'utiliser dans les commandes suivantes. Vous en aurez besoin pour joindre votre certificat à votre appareil et pour joindre la politique au certificat lors des étapes ultérieures.

```
{
    "certificateArn": "arn:aws:iot:us-west-2:123456789012:cert/9894ba17925e663f1d29c23af4582b8e3b7619c31f3fb93adcb51ae54b83dc2",
    "certificateId": "9894ba17925e663f1d29c23af4582b8e3b7619c31f3fb93adcb51ae54b83dc2",
    "certificatePem": "",
    -----BEGIN CERTIFICATE-----
MIICiTCCEXAMPLE6m7oRw0uX0jANBgkqhkiG9w0BAQUFADCBiDELMAkGA1UEBhMCVVMxCzAJBgNVBAgEXAMPLEAwDgYDVQQHEwdTZWFOdGxLMQ8wDQYDVQQKEwZBbWF6b24xFDASBgNVBAsTC0LBTSExAMPLE2xLMRIwEAYDVQDDEwLUZXN0Q2lsYWmxHzAdBgkqhkiG9w0BCQEWEg5vb25lQGFtYExAMPLEb20wHhcNMTEwNDI1MjA0NTIxWhCNMTIwNDI0MjA0NTIxWjCBiDELMAkGA1UEBhMCEXAMPLEJBgNVBAgTAldBMRAwDgYDQQQHEwdTZWFOdGxLMQ8wDQYDVQQKEwZBbWF6b24xFDAExAMPLEsTC0LBTSBDb25zb2xLMRIwEAYDVQDDEwLUZXN0Q2lsYWmxHzAdBgkqhkiG9w0BCQEExAMPLE25lQGFtYXpvbi5jb20wgZ8wDQYJKoZIhvCNQAEBBQADgY0AMIGJAOGBAMk0dn+aExAMPLEExAMPLEfEvySwtC2XADZ4nB+BLYgVIk60CpiwsZ3G93vUEI03iyNoH/f0wYK8m9TrDHudUZExAMPLELG5M43q7Wgc/MbQITx0USQv7c7ugFFDzQGBzZswY6786m86gpEIbb30hjZnzcVQAXExAMPLEWIm2nrAgMBAEwDQYJKoZIhvCNQEFBQADgYEAtCu4nUhVVxYUntneD9+h8Mg9qExAMPLEyExzyLwaxlAoo7TJHidbtS4J51NmZgXL0FkbFFBjvSfpJILJ00zbhNYS5f6GuoEDExAMPLEBHjJnyp3780D8uTs7fLvjx79LjSTbNYiytVbZPQUQ5Yaxu2jXnimvw3rrszlaExAMPLE
    -----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkExAMPLEQEFAA0CAQ8AMIIIBCgKCAQEAExAMPLE1nnnyJwKSMHw4h\nnMMExAMPLEeuN/dMAS3fyce8DW/4+ExAMPLEyjmoF/YVF/\nghr99VEExAMPLE5VF13\nn59VK7cExAMPLE67GK+y+jikqX0gHh/xJTtwo\n+sGpWExAMPLEdZ18x0d2ka4tCzuWExAMPLEahJbYkCPUBSU8opVkr7qkExAMPLE1DR6sx2Hocli00Lu6Fkw91swQWExAMPLE\n\\GB3ZPrNh0PzQYvjUStZeccyNx2ExAMPLEvp9mQ0UXP6plfgxwKRX2fExAMPLEd\n\\nhJLXkX3rHU2xbxJSq7D\n+XExAMPLEcw+LyFhI5mgFRl88eGdsAExAMPLElnI9EesG\\nFQIDAQAB\\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----\nkey omitted for security\nreasons\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

- Dans votre fenêtre Amazon EC2 Instance Connect, attachez votre objet au certificat que vous venez de créer à l'aide de la commande suivante et du *CertificateArn* dans la réponse de la commande précédente.

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

En cas de succès, cette commande n'affiche aucun résultat.

Pour créer et attacher une stratégie

- Dans votre fenêtre Amazon EC2 Instance Connect, créez le fichier de politique en copiant et en collant ce document de politique dans un fichier nommé. *~/policy.json*

Si vous n'avez pas d'éditeur Linux préféré, vous pouvez l'ouvrir nano à l'aide de cette commande.

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

Collez-y le document *policy.json* de politique. Entrez **ctrl-x** pour quitter l'nanoeéditeur et enregistrer le fichier.

Contenu du document de politique pour *policy.json*.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish",  
                "iot:Subscribe",  
                "iot:Receive",  
                "iot:Connect"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

2. Dans votre fenêtre Amazon EC2 Instance Connect, créez votre politique à l'aide de la commande suivante.

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

Sortie :

```
{  
    "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 votre fenêtre Amazon EC2 Instance Connect, associez la politique au certificat de votre appareil virtuel à l'aide de la commande suivante.

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

En cas de succès, cette commande n'affiche aucun résultat.

Installez le SDK de AWS IoT l'appareil pour JavaScript

Dans cette section, vous allez installer le SDK AWS IoT Device pour JavaScript, qui contient le code que les applications peuvent utiliser pour communiquer ainsi que AWS IoT des exemples de programmes. Pour plus d'informations, consultez le [SDK AWS IoT Device pour le JavaScript GitHub référentiel](#).

Pour installer le AWS IoT Device SDK pour JavaScript sur votre instance Linux

1. Dans votre fenêtre Amazon EC2 Instance Connect, clonez le AWS IoT Device SDK pour le JavaScript référentiel dans le `aws-iot-device-sdk-js-v2` répertoire 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 au `aws-iot-device-sdk-js-v2` répertoire que vous avez créé à l'étape précédente.

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

3. Utilisez npm pour installer le SDK.

```
npm install
```

Exécution de l'exemple d'application

Les commandes des 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	<code>~/certs/private.pem.key</code>
Certificat de l'appareil	<code>~/certs/device.pem.crt</code>
Certificat racine de l'autorité de certification	<code>~/certs/Amazon-root-CA-1.pem</code>

Dans cette section, vous allez installer et exécuter l'`pub-sub.js` exemple d'application qui se trouve dans le `aws-iot-device-sdk-js-v2/samples/node` répertoire du AWS IoT Device SDK pour JavaScript. Cette application montre comment un appareil, votre instance Amazon EC2, utilise la bibliothèque MQTT pour publier des messages MQTT et s'y abonner. L'`pub-sub.js` exemple d'application s'abonne à une rubrique `topic_1`, publie 10 messages sur cette rubrique et affiche les messages tels qu'ils sont reçus de la part du courtier de messages.

Pour installer et exécuter l'exemple d'application

1. Dans votre fenêtre Amazon EC2 Instance Connect, accédez au `aws-iot-device-sdk-js-v2/samples/node/pub_sub` répertoire créé par le SDK 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 votre fenêtre Amazon EC2 Instance Connect, accédez à *votre point de terminaison IoT à l'aide de AWS IoT cette commande*.

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

3. Dans votre fenêtre Amazon EC2 Instance Connect, insérez *votre point de terminaison IoT* comme indiqué et exécutez cette commande.

```
node dist/index.js --topic topic_1 --ca_file ~/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 AWS IoT Core à votre compte.
2. S'abonne à la rubrique du message, topic_1, et affiche les messages qu'il reçoit à ce sujet.
3. Publie 10 messages sur le sujet, topic_1.
4. Affiche un résultat semblable à ce qui suit :

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

Si vous rencontrez des difficultés en exécutant l'exemple d'application, consultez[the section called "Résolution des problèmes liés à l'exemple d'application" \(p. 69\)](#).

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

Afficher les messages de l'exemple d'application dans la AWS IoT console

Vous pouvez voir les messages de l'exemple d'application lorsqu'ils passent par le courtier de messages en utilisant le client de test MQTT dans la AWS 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 \(p. 71\)](#). Cela vous permet d'apprendre à utiliser le client de test MQTT dans la AWS IoTconsole pour afficher les messages MQTT lorsqu'ils passent par le courtier de messages.
2. Ouvrez le client de test MQTT dans la AWS IoTconsole.
3. Dans S'abonner à une rubrique, S'abonner à la rubrique, topic_1.

4. Dans votre fenêtre Amazon EC2 Instance Connect, réexécutez l'exemple d'application et regardez les messages du client de test MQTT sur la AWS IoT console.

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

Pour plus d'informations sur MQTT et sur la prise en AWS IoT Core charge du protocole, consultez [MQTT](#).

Utilisez votre PC ou Mac Windows ou Linux comme AWS IoT appareil

Dans ce tutoriel, vous allez configurer un ordinateur personnel à utiliser avec AWS IoT. Ces instructions sont compatibles avec les PC et Mac Windows et Linux. Pour ce faire, vous devez installer certains logiciels sur votre ordinateur. Si vous ne souhaitez pas installer de logiciel sur votre ordinateur, vous pouvez essayer [Création d'un appareil virtuel avec Amazon EC2 \(p. 44\)](#), qui installe tous les logiciels sur une machine virtuelle.

Dans le cadre de ce didacticiel, vous effectuerez les tâches suivantes :

- [Configuration de votre ordinateur personnel \(p. 52\)](#)
- [Installation de Git, Python et du SDK AWS IoT Device pour Python \(p. 52\)](#)
- [Configurez la politique et exécutez l'exemple d'application \(p. 55\)](#)
- [Afficher les messages de l'exemple d'application dans la AWS IoT console \(p. 58\)](#)
- [Exécutez l'exemple d'abonnement partagé en Python \(p. 58\)](#)

Configuration de votre ordinateur personnel

Pour terminer ce didacticiel, vous avez besoin d'un PC Windows ou Linux ou d'un Mac connecté à Internet.

Avant de passer à l'étape suivante, assurez-vous de pouvoir ouvrir une fenêtre de ligne de commande sur votre ordinateur. À utiliser cmd.exe sur un PC Windows. Sur un PC Linux ou un Mac, utilisez Terminal.

Installation de Git, Python et du SDK AWS IoT Device pour Python

Dans cette section, vous allez installer Python et le AWS IoT Device SDK pour Python sur votre ordinateur.

Installation de 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, cela signifie que Git est installé et vous pouvez passer à l'étape suivante.

Si la commande affiche une erreur, ouvrez <https://git-scm.com/download> et installez Git sur votre ordinateur.

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

```
python -V
```

Note

Si cette commande génère 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 par python3 et poursuivez le reste de ce didacticiel.

Si la commande affiche la version de Python, cela signifie que Python est déjà installé. Ce didacticiel nécessite Python v3.7 ou une version ultérieure.

3. Si Python est installé, vous pouvez ignorer le reste des étapes de cette section. Si ce n'est pas le cas, continuez.
4. Ouvrez <https://www.python.org/downloads/> et téléchargez le programme d'installation pour votre ordinateur.
5. Si l'installation du téléchargement ne démarre pas automatiquement, exécutez le programme téléchargé pour installer Python.
6. Vérifier l'installation de Python.

```
python -V
```

Vérifiez que la commande affiche la version de Python. Si la version de Python ne s'affiche pas, essayez à nouveau de télécharger et d'installer Python.

Installation du SDK de AWS IoT l'appareil pour Python

Pour installer le AWS IoT Device SDK pour Python sur votre ordinateur

1. Installez la version 2 du AWS IoT Device SDK pour Python.

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

2. Clonez le référentiel AWS IoT Device SDK pour 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 répertoire d'*accueil*.

L'emplacement réel du *répertoire* personnel dépend de votre système d'exploitation.

Linux/macOS

Sous macOS et Linux, le *répertoire* personnel est~.

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

Windows

Sous Windows, vous pouvez trouver le chemin du *répertoire* personnel en exécutant cette commande dans la cmd fenêtre.

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

Note

Si vous utilisez Windows PowerShell au lieu de cmd.exe, utilisez la commande suivante.

```
echo $home
```

Pour d'abord d'ouvrir le dossier AWS IoT dans votre dossier GitHub, tapez la commande suivante :

Se préparer à exécuter les exemples d'applications

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

- Créez le `certs` répertoire. Dans le `certs` répertoire, copiez les fichiers de clé privée, de certificat d'appareil et de certificat CA racine que vous avez enregistrés lors de la création et de l'enregistrement de l'objet dans [la section intitulée "Création de AWS IoT ressources" \(p. 40\)](#). Les noms de fichier de chaque fichier du répertoire de destination doivent correspondre à ceux du tableau.

Les commandes de 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 le `certs` sous-répertoire que vous utiliserez lorsque vous exécuterez les exemples d'applications.

```
mkdir ~/certs
```

Dans le nouveau sous-répertoire, copiez les fichiers vers les chemins des fichiers 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 du `certs` répertoire et les comparer à ceux répertoriés dans le tableau.

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

Windows

Exécutez cette commande pour créer le `certs` sous-répertoire que vous utiliserez lorsque vous exécuterez les exemples d'applications.

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

Dans le nouveau sous-répertoire, copiez les fichiers vers les chemins des fichiers 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 du `certs` répertoire et les comparer à ceux répertoriés dans le tableau.

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

Configurez la politique et exécutez l'exemple d'application

Dans cette section, vous allez configurer votre politique et exécuter l'`pubsub.py` exemple de script qui se trouve dans le `aws-iot-device-sdk-python-v2/samples` répertoire du Kit SDK des appareils AWS IoT pour Python. Ce script montre comment votre appareil utilise la bibliothèque MQTT pour publier des messages MQTT et s'y abonner.

L'`pubsub.py` exemple d'application s'abonne à une rubrique `est/topic`, publie 10 messages sur cette rubrique et affiche les messages tels qu'ils sont reçus de la part du courtier de messages.

Pour exécuter l'`pubsub.py` exemple de script, vous avez besoin des informations suivantes :

Valeurs des paramètres de l'application

Paramètre	Où trouver la valeur
<i>votre point de terminaison IoT</i>	1. Dans la AWS IoTconsole , dans le menu de gauche, choisissez Paramètres. 2. Sur la page Paramètres, votre point de terminaison s'affiche dans la section Point de terminaison des données de l'appareil.

La valeur `your-iot-endpoint` est au format :`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 politique de votre appareil autorise l'exemple de script à se connecter, à s'abonner, à publier et à recevoir.

Pour rechercher et consulter le document de politique d'une ressource liée à un objet

1. Dans la [AWS IoTconsole](#), dans la liste des objets, recherchez la ressource d'objets qui représente votre appareil.
2. Cliquez sur le lien Nom de la ressource d'objet qui représente votre appareil pour ouvrir la page de détails de l'objet.

3. Sur la page Détails de l'objet, dans l'onglet Certificats, choisissez le certificat associé à la ressource de l'objet. Il ne doit y avoir qu'un seul certificat dans la liste. S'il y en a plusieurs, choisissez le certificat dont les fichiers sont installés sur votre appareil et qui sera utilisé pour vous connecter AWS IoT Core.

Sur la page Détails du certificat, dans l'onglet Politiques, choisissez la politique associée au certificat. Il ne doit y en avoir qu'un. S'il y en a plusieurs, répétez l'étape suivante pour chacune d'elles afin de vous assurer qu'au moins une politique accorde l'accès requis.

4. Sur la page de présentation de la politique, recherchez l'éditeur JSON et choisissez Modifier le document de politique pour revoir et modifier le document de stratégie selon vos besoins.
5. La politique JSON s'affiche dans l'exemple suivant. Dans l'"Resource" élément, *region:account* remplacez-le par votre Région AWS et Compte AWS dans chacune des Resource valeurs.

```
{  
    "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, accédez au `~/aws-iot-device-sdk-python-v2/samples/node/pub_sub` répertoire 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 *your-iot-endpoint* comme indiqué et exécutez cette commande.

```
python3 pubsub.py --endpoint your-iot-endpoint --ca_file ~/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, accédez au %USERPROFILE%\aws-iot-device-sdk-python-v2\samples répertoire créé par le SDK 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 *your-iot-endpoint* comme indiqué et exécutez cette commande.

```
python3 pubsub.py --endpoint your-iot-endpoint --ca_file %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 AWS IoT Core à votre compte.
2. S'abonne au sujet du message, au test/au sujet et affiche les messages qu'il reçoit sur ce sujet.
3. Publie 10 messages sur le sujet, le test/le sujet.
4. Affiche un résultat semblable à ce qui suit :

```
Connected!
Subscribing to topic 'test/topic'...
Subscribed with QoS.AT\_LEAST\_ONCE
Sending 10 message(s)
Publishing message to topic 'test/topic': Hello World! [1]
Received message from topic 'test/topic': b'"Hello World! [1]"'
Publishing message to topic 'test/topic': Hello World! [2]
Received message from topic 'test/topic': b'"Hello World! [2]"'
Publishing message to topic 'test/topic': Hello World! [3]
Received message from topic 'test/topic': b'"Hello World! [3]"'
Publishing message to topic 'test/topic': Hello World! [4]
Received message from topic 'test/topic': b'"Hello World! [4]"'
Publishing message to topic 'test/topic': Hello World! [5]
Received message from topic 'test/topic': b'"Hello World! [5]"'
Publishing message to topic 'test/topic': Hello World! [6]
Received message from topic 'test/topic': b'"Hello World! [6]"'
Publishing message to topic 'test/topic': Hello World! [7]
Received message from topic 'test/topic': b'"Hello World! [7]"'
Publishing message to topic 'test/topic': Hello World! [8]
Received message from topic 'test/topic': b'"Hello World! [8]"'
Publishing message to topic 'test/topic': Hello World! [9]
Received message from topic 'test/topic': b'"Hello World! [9]"'
Publishing message to topic 'test/topic': Hello World! [10]
Received message from topic 'test/topic': b'"Hello World! [10]"'
10 message(s) received.
Disconnecting...
Disconnected!
```

Si vous rencontrez des difficultés en exécutant l'exemple d'application, consultez[the section called "Résolution des problèmes liés à l'exemple d'application" \(p. 69\)](#).

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

Afficher les messages de l'exemple d'application dans la AWS IoT console

Vous pouvez voir les messages de l'exemple d'application lorsqu'ils passent par le courtier de messages en utilisant le client de test MQTT dans la AWS 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 \(p. 71\)](#). Cela vous permet d'apprendre à utiliser le client de test MQTT dans la AWS IoTconsole pour afficher les messages MQTT lorsqu'ils passent par le courtier de messages.
2. Ouvrez le client de test MQTT dans la AWS IoTconsole.
3. Dans S'abonner à un sujet, abonnez-vous au sujet, test/sujet.
4. Dans votre fenêtre de ligne de commande, exécutez à nouveau l'exemple d'application et regardez les messages du client MQTT sur la AWS IoTconsole.

Linux/macOS

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

Pour plus d'informations sur MQTT et sur la prise en AWS IoT Core charge du protocole, consultez [MQTT](#).

Exécutez l'exemple d'abonnement partagé en Python

AWS IoT Core prend en charge les [abonnements partagés \(p. 103\)](#) pour MQTT 3 et MQTT 5. Les abonnements partagés permettent à plusieurs clients de partager un abonnement à un sujet et un seul client recevra les messages publiés sur ce sujet en utilisant une distribution aléatoire. Pour utiliser les abonnements partagés, les clients s'abonnent au [filtre de sujets](#) d'un abonnement partagé `:$share/{ShareName}/{TopicFilter}`.

Pour configurer la politique et exécuter l'exemple d'abonnement partagé

1. Pour exécuter l'exemple d'abonnement partagé, vous devez configurer la politique de votre appareil comme indiqué dans [MQTT 5 Shared Subscription](#).
2. Pour exécuter l'exemple d'abonnement partagé, exécutez les commandes suivantes.

Linux/macOS

Pour exécuter l'exemple de script sous Linux/macOS

1. Dans votre fenêtre de ligne de commande, accédez au `~/aws-iot-device-sdk-python-v2/samples` répertoire 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 *your-iot-endpoint* comme indiqué et exécutez cette commande.

```
python3 mqtt5_shared_subscription.py --endpoint your-iot-endpoint --ca_file  
~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/  
private.pem.key --group_identifier consumer
```

Windows

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

1. Dans votre fenêtre de ligne de commande, accédez au %USERPROFILE%\aws-iot-device-sdk-python-v2\samples répertoire créé par le SDK 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 *your-iot-endpoint* comme indiqué et exécutez cette commande.

```
python3 mqtt5_shared_subscription.py --endpoint your-iot-endpoint --ca_file  
%USERPROFILE%\certs\Amazon-root-CA-1.pem --cert %USERPROFILE%\certs  
\device.pem.crt --key %USERPROFILE%\certs\private.pem.key --group_identifier  
consumer
```

Note

Vous pouvez éventuellement spécifier un identifiant de groupe en fonction de vos besoins lorsque vous exécutez l'exemple (par exemple, --group_identifier consumer). Si vous n'espécifiez pas d'identifiant, python-sample c'est l'identifiant par défaut pour le groupe.

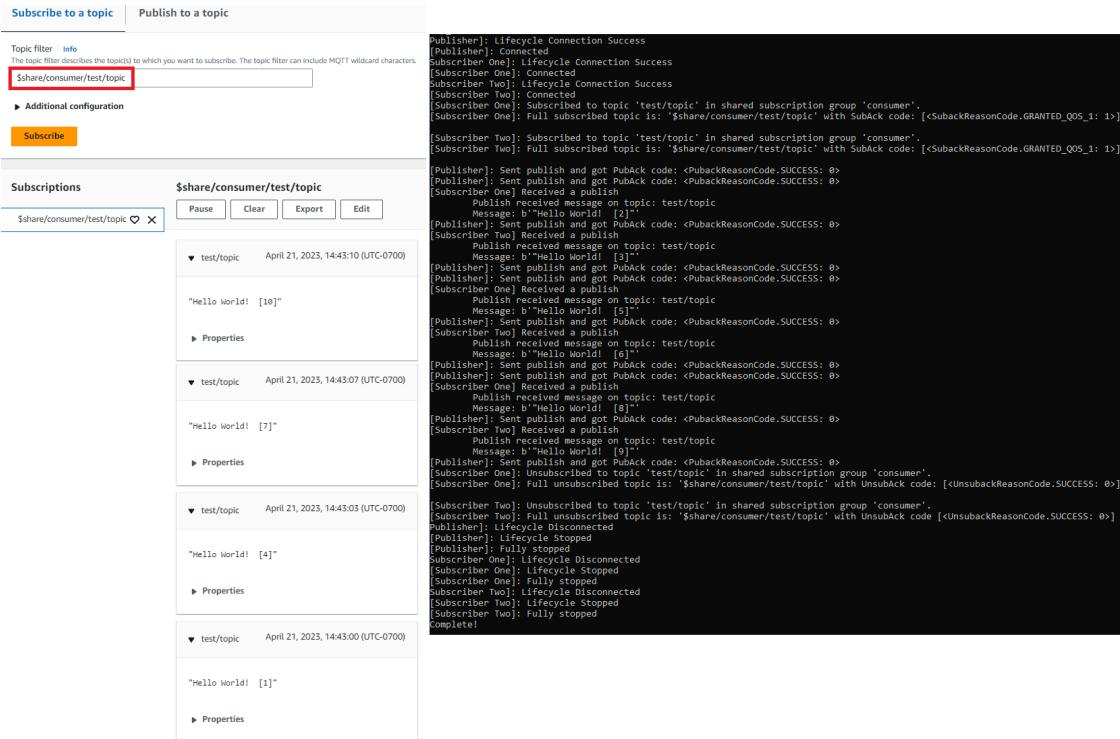
3. Le résultat de votre ligne de commande peut ressembler à ce qui suit :

```
Publisher]: Lifecycle Connection Success
[Publisher]: Connected
Subscriber One]: Lifecycle Connection Success
[Subscriber One]: Connected
Subscriber Two]: Lifecycle Connection Success
[Subscriber Two]: Connected
[Subscriber One]: Subscribed to topic 'test/topic' in shared subscription group
'consumer'.
[Subscriber One]: Full subscribed topic is: '$share/consumer/test/topic' with SubAck
code: [<SubackReasonCode.GRANTED_QOS_1: 1>]
[Subscriber Two]: Subscribed to topic 'test/topic' in shared subscription group
'consumer'.
[Subscriber Two]: Full subscribed topic is: '$share/consumer/test/topic' with SubAck
code: [<SubackReasonCode.GRANTED_QOS_1: 1>]
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber Two] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [1]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber One] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [2]"'
```

```
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber Two] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [3]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber One] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [4]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber Two] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [5]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber One] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [6]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber Two] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [7]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber One] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [8]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber Two] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [9]"'
[Publisher]: Sent publish and got PubAck code: <PubackReasonCode.SUCCESS: 0>
[Subscriber One] Received a publish
    Publish received message on topic: test/topic
    Message: b'"Hello World! [10]"'
[Subscriber One]: Unsubscribed to topic 'test/topic' in shared subscription group
'consumer'.
[Subscriber One]: Full unsubscribed topic is: '$share/consumer/test/topic' with
UnsubAck code: [<UnsubackReasonCode.SUCCESS: 0>]
[Subscriber Two]: Unsubscribed to topic 'test/topic' in shared subscription group
'consumer'.
[Subscriber Two]: Full unsubscribed topic is: '$share/consumer/test/topic' with
UnsubAck code [<UnsubackReasonCode.SUCCESS: 0>]
Publisher]: Lifecycle Disconnected
[Publisher]: Lifecycle Stopped
[Publisher]: Fully stopped
Subscriber One]: Lifecycle Disconnected
[Subscriber One]: Lifecycle Stopped
[Subscriber One]: Fully stopped
Subscriber Two]: Lifecycle Disconnected
[Subscriber Two]: Lifecycle Stopped
[Subscriber Two]: Fully stopped
[Subscriber Two]: Fully stopped
Complete!
```

4. Ouvrez le client de test MQTT dans la AWS IoTconsole. Dans S'abonner à une rubrique, abonnez-vous à la rubrique de l'abonnement partagé telle que :\$share/consumer/test/topic. Vous pouvez spécifier un identifiant de groupe en fonction de vos besoins lorsque vous exécutez l'exemple (par exemple,--group_identifier consumer). Si vous ne spécifiez pas d'identifiant de groupe, la valeur par défaut estpython-sample. Pour plus d'informations, consultez l'[exemple Python d'abonnement partagé MQTT 5](#) et les [abonnements partagés \(p. 103\)](#) du Guide du AWS IoT Core développeur.

Dans votre fenêtre de ligne de commande, exécutez à nouveau l'exemple d'application et observez la distribution des messages dans votre client de test MQTT sur la AWS IoTconsole et sur la ligne de commande.



Connect un Raspberry Pi ou un autre appareil

Dans cette section, nous allons configurer un Raspberry Pi à utiliser avec AWS IoT. Si vous souhaitez connecter un autre appareil, les instructions relatives au 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 devez installer de nombreuses mises à niveau du logiciel système.

Dans le cadre de ce didacticiel, vous effectuerez les tâches suivantes :

- [Configuration de votre appareil \(p. 62\)](#)
- [Installez les outils et bibliothèques requis pour le AWS IoT Device SDK \(p. 62\)](#)
- [Installer le SDK AWS IoT de l'appareil \(p. 63\)](#)
- [Installation et exécution de l'exemple d'application \(p. 65\)](#)
- [Afficher les messages de l'exemple d'application dans la AWS IoT console \(p. 68\)](#)

Important

L'adaptation de ces instructions à d'autres appareils et systèmes d'exploitation peut s'avérer difficile. Vous devez comprendre suffisamment bien 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 pour AWS IoT, vous pouvez essayer l'une des autres options de l'appareil comme alternative, telle que [Création d'un appareil virtuel avec Amazon EC2 \(p. 44\)](#) ou [Utilisez votre PC ou Mac Windows ou Linux comme AWS IoT appareil \(p. 52\)](#).

Configuration de votre appareil

L'objectif de cette étape est de collecter les informations dont vous aurez besoin pour configurer votre appareil afin qu'il puisse démarrer le système d'exploitation (SE), se connecter à Internet et vous permettre d'interagir avec lui via 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 dans [Configurez votre Compte AWS \(p. 19\)](#) avant de continuer.
- Un [Raspberry Pi 3 modèle B](#) ou un modèle plus récent. Cela peut fonctionner sur les versions antérieures du Raspberry Pi, mais elles n'ont pas été testées.
- 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 elles n'ont pas été testées.

Pour exécuter cet exemple, il n'est pas nécessaire d'installer le poste de travail doté de l'interface utilisateur graphique (GUI) ; toutefois, si vous utilisez le Raspberry Pi pour la première fois et que votre matériel Raspberry Pi le prend en charge, il peut être plus facile d'utiliser le bureau avec l'interface graphique.

- Un port Ethernet ou WiFi une connexion.
- Clavier, souris, moniteur, câbles, blocs d'alimentation et autre matériel requis par votre appareil.

Important

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

Si votre Raspberry Pi utilise un système d'exploitation doté d'une interface utilisateur graphique (GUI), ouvrez une fenêtre de terminal sur l'appareil et suivez 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 le AWS IoT Device SDK

Avant d'installer le kit SDK AWS IoT Device et l'exemple de code, assurez-vous que votre système est à jour et dispose des outils et bibliothèques nécessaires à l'installation des kits SDK.

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

Avant d'installer un SDK pour AWS IoT appareil, exécutez ces commandes dans une fenêtre de terminal de 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. Installation de Git

Si Git n'est pas installé sur le système d'exploitation de votre appareil, vous devez l'installer pour installer le AWS IoT Device SDK pour JavaScript.

- a. Vérifiez 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 git commande, 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 le AWS IoT Device SDK pour JavaScript.

Installer le SDK AWS IoT de l'appareil

Installez le SDK de AWS IoT l'appareil.

Python

Dans cette section, vous allez installer Python, ses outils de développement et le AWS IoT Device SDK pour Python sur votre appareil. Ces instructions concernent un Raspberry Pi exécutant le dernier système d'exploitation Raspberry Pi. Si vous avez un autre appareil ou que vous utilisez un autre système d'exploitation, vous devrez peut-être adapter ces instructions à votre appareil.

1. Installation de Python et de ses outils de développement

Le AWS IoT Device SDK pour Python nécessite l'installation de Python v3.5 ou version ultérieure sur votre Raspberry Pi.

Dans une fenêtre de terminal de votre appareil, exécutez ces commandes.

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 supérieure, 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 vérifier que la version correcte de Python est désormais installée.

```
python3 --version
```

2. Test pour pip3

Dans une fenêtre de terminal de votre appareil, exécutez ces commandes.

1. Exécutez cette commande pour voir si elle pip3 est installée.

```
pip3 --version
```

2. Si la commande renvoie un numéro de version, elle pip3 est installée et vous pouvez passer à l'étape 3.

3. Si la commande précédente renvoie une erreur, exécutez cette commande pour procéder à l'installation pip3.

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

4. Exécutez cette commande pour voir si elle pip3 est installée.

```
pip3 --version
```

3. Installation du SDK AWS IoT Device actuel pour Python

Installez le SDK AWS IoT Device 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 packages npm et le AWS IoT Device SDK pour JavaScript sur votre appareil. Ces instructions concernent un Raspberry Pi exécutant le système d'exploitation Raspberry Pi. Si vous avez un autre appareil ou que vous utilisez un autre système d'exploitation, vous devrez peut-être adapter ces instructions à votre appareil.

1. Installer la dernière version de Node.js

Le AWS IoT Device SDK pour JavaScript nécessite que Node.js et le gestionnaire de packages npm soient installés sur votre Raspberry Pi.

- a. Téléchargez la dernière version du référentiel Node 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. Vérifier l'installation de Node.

```
node -v
```

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

- d. Vérifier l'installation de npm.

```
npm -v
```

Vérifiez que la commande affiche la version de npm. Si la version de npm ne s'affiche pas, essayez à nouveau d'installer Node et npm.

- e. Redémarrez le périphérique.

```
sudo shutdown -r 0
```

Continuez après le redémarrage de l'appareil.

2. Installez le SDK de AWS IoT l'appareil pour JavaScript

Installez le AWS IoT Device SDK pour JavaScript votre Raspberry Pi.

- a. Clonez le AWS IoT Device SDK pour le JavaScript dépôt dans le aws-iot-device-sdk-js-v2 répertoire de votre *répertoire* personnel. Sur le Raspberry Pi, le *répertoire* personnel est~/, qui est utilisé comme *répertoire* personnel dans les commandes suivantes. Si votre appareil utilise un chemin différent pour le *répertoire* personnel, vous devez le ~/ remplacer par le chemin correspondant à votre appareil dans les commandes suivantes.

Ces commandes créent le ~/aws-iot-device-sdk-js-v2 répertoire et y copient le code du SDK.

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

- b. Accédez au aws-iot-device-sdk-js-v2 répertoire que vous avez créé à l'étape précédente et exécutez-le npm install pour installer le SDK. La commande npm install invoque la commande aws-crt invoque la commande, qui peut prendre quelques minutes.

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

Installation et exécution de l'exemple d'application

Dans cette section, vous allez installer et exécuter l'pubsubexemple d'application qui se trouve dans le SDK de l'AWS IoT appareil. Cette application montre comment votre appareil utilise la bibliothèque MQTT pour publier des messages MQTT et s'y abonner. L'exemple d'application s'abonne à une rubrique topic_1, publie 10 messages sur cette rubrique et affiche les messages tels qu'ils sont reçus de la part du courtier de messages.

Installation des fichiers de certificat.

L'exemple d'application nécessite que les fichiers de certificat qui authentifient l'appareil soient installés sur celui-ci.

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

1. Créez un `certs` sous-répertoire dans votre *répertoire* personnel en exécutant ces commandes.

```
cd ~  
mkdir certs
```

2. Dans le `~/certs` répertoire, copiez la clé privée, le certificat de l'appareil et le certificat de l'autorité de certification racine que vous avez créés précédemment dans [the section called “Création de AWS IoT ressources” \(p. 40\)](#).

La façon dont vous copiez les fichiers de certificat sur votre appareil dépend de l'appareil 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 exécuter la procédure décrite dans le navigateur Web [the section called “Création de AWS IoT ressources” \(p. 40\)](#) de votre appareil pour télécharger les fichiers obtenus 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	<code>~/certs/Amazon-root-CA-1.pem</code>
Certificat de l'appareil	<code>~/certs/device.pem.crt</code>
Clé privée	<code>~/certs/private.pem.key</code>

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

Valeurs des paramètres de l'application

Paramètre	Où trouver la valeur
<i>votre point de terminaison IoT</i>	Dans la AWS IoT console , choisissez Tous les appareils, puis Objets. Choisissez l'objet IoT que vous avez créé pour votre appareil, MyIoTThings sous le nom utilisé précédemment, puis choisissez Interact. Sur la page des détails de l'objet, votre point de terminaison s'affiche dans la section HTTPS. Si vous utilisez la nouvelle AWS IoT console, choisissez Paramètres AWS IoT dans le menu. Votre point de terminaison s'affiche dans la section Point de terminaison des données de l'appareil.

La valeur `your-iot-endpoint` est au format :`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'exemples d'applications.

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

2. Dans la fenêtre de ligne de commande, remplacez **your-iot-endpoint** comme indiqué et exécutez cette commande.

```
python3 pubsub.py --topic topic_1 --ca_file ~/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 au AWS IoT service associé à votre compte.
2. S'abonne à la rubrique du message, topic_1, et affiche les messages qu'il reçoit à ce sujet.
3. Publie 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 difficultés en exécutant l'exemple d'application, consultez[the section called "Résolution des problèmes liés à l'exemple d'application" \(p. 69\)](#).

Vous pouvez également ajouter le **--verbosity Debug** paramètre à la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'il 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, accédez au `~/aws-iot-device-sdk-js-v2/samples/node/pub_sub` répertoire créé par le SDK et installez l'exemple d'application à l'aide de ces commandes. La commande `npm install` invoque la commande `aws-crt` qui peut prendre quelques minutes.

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

2. Dans la fenêtre de ligne de commande, remplacez `your-iot-endpoint` comme indiqué et exécutez cette commande.

```
node dist/index.js --topic topic_1 --ca_file ~/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 au AWS IoT service associé à votre compte.
2. S'abonne à la rubrique du message, `topic_1`, et affiche les messages qu'il reçoit à ce sujet.
3. Publie 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 difficultés en exécutant l'exemple d'application, consultez[the section called "Résolution des problèmes liés à l'exemple d'application" \(p. 69\).](#)

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

Afficher les messages de l'exemple d'application dans la AWS IoT console

Vous pouvez voir les messages de l'exemple d'application lorsqu'ils passent par le courtier de messages en utilisant le client de test MQTT dans la AWS IoT console.

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 \(p. 71\)](#). Cela vous permet d'apprendre à utiliser le client de test MQTT dans la AWS IoTconsole pour afficher les messages MQTT lorsqu'ils passent par le courtier de messages.
2. Ouvrez le client de test MQTT dans la AWS IoTconsole.
3. Abonnez-vous à la rubrique, topic_1.
4. Dans votre fenêtre de ligne de commande, exécutez à nouveau l'exemple d'application et regardez les messages du client MQTT sur la AWS IoTconsole.

Python

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

Résolution des problèmes liés à l'exemple d'application

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

Vérifiez le certificat

Si le certificat n'est pas actif, n'AWS IoT acceptera aucune tentative de connexion l'utilisant à des fins d'autorisation. Lors de la création de votre certificat, il est facile d'oublier le bouton Activer. Heureusement, vous pouvez activer votre certificat depuis la [AWS IoTconsole](#).

Pour vérifier l'activation de votre certificat

1. Dans la [AWS IoTconsole](#), dans le menu de gauche, choisissez Secure, puis Certificats.
2. Dans la liste des certificats, recherchez le certificat que vous avez créé pour l'exercice et vérifiez son statut dans la colonne État.

Si vous ne vous souvenez pas du nom du certificat, recherchez ceux qui sont inactifs pour voir s'il s'agit peut-être de celui que vous utilisez.

Choisissez le certificat dans la liste pour ouvrir sa page détaillée. Sur la page détaillée, vous pouvez voir sa date de création pour vous aider à identifier le certificat.

3. Pour activer un certificat inactif, sur la page détaillée du certificat, choisissez Actions, puis Activate.

Si vous avez trouvé le bon certificat et qu'il est actif, mais que vous rencontrez toujours des problèmes lors de l'exécution de l'exemple d'application, vérifiez sa politique comme décrit à l'étape suivante.

Vous pouvez également essayer de créer un nouvel objet et un nouveau certificat en suivant les étapes décrites dans[the section called “Création d'un objet” \(p. 42\)](#). Si vous créez un nouvel objet, vous devez lui donner un nouveau nom et télécharger les nouveaux fichiers de certificat sur votre appareil.

Vérifiez la politique associée au certificat

Les politiques autorisent les actions dans AWS IoT. Si le certificat utilisé pour se connecter AWS IoT n'a pas de politique, ou si aucune politique ne l'autorise à se connecter, la connexion sera refusée, même si le certificat est actif.

Pour vérifier les politiques associées à un certificat

1. Recherchez le certificat comme décrit dans la rubrique précédente et ouvrez sa page de détails.
2. Dans le menu de gauche de la page de détails du certificat, choisissez Politiques pour voir les politiques associées au certificat.
3. Si aucune politique n'est associée au certificat, ajoutez-en une en choisissant le menu Actions, puis en choisissant Attacher une politique.

Choisissez la stratégie que vous avez créée précédemment dans [the section called "Création de AWS IoT ressources" \(p. 40\)](#).

4. Si une politique est associée, choisissez la vignette de la politique pour ouvrir sa page de détails.

Sur la page de détails, consultez le document de politique pour vous assurer qu'il contient les mêmes informations que celles que vous avez créées [the section called "Création d'une stratégie AWS IoT" \(p. 41\)](#).

Vérifier la ligne de commande.

Assurez-vous d'avoir utilisé la bonne ligne de commande pour votre système. Les commandes utilisées sur les systèmes Linux et macOS sont souvent différentes de celles utilisées sur les systèmes Windows.

Vérifiez l'adresse du point de terminaison

Vérifiez la commande que vous avez saisie et vérifiez que l'adresse du point de terminaison figurant dans votre commande correspond à celle de votre [AWS IoT console](#).

Vérifiez les noms des fichiers de certificat

Comparez les noms de fichiers contenus dans la commande que vous avez saisie avec les noms des fichiers de certificat du `certs` répertoire.

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

Vérifier 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 suffit de trouver la section du didacticiel intitulée Installer le SDK du AWS IoT périphérique pour le **Langage du SDK** et de suivre à nouveau la procédure.

Si vous utilisez le SDK AWS IoT Device 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 explique comment utiliser le client de test AWS IoT MQTT dans la [AWS IoTconsole](#) pour regarder les messages MQTT envoyés et reçus par AWS IoT. L'exemple utilisé dans cette section concerne les exemples utilisés dans [Démarrez avec AWS IoT Core \(p. 18\)](#); toutefois, vous pouvez remplacer le *topicName* utilisé dans les exemples par n'importe quel [nom de rubrique ou filtre de rubrique \(p. 115\)](#) utilisé par votre solution IoT.

Les appareils publient des messages MQTT identifiés par [des sujets \(p. 115\)](#) auxquels communiquer leur état AWS IoT, et AWS IoT publient des messages MQTT pour informer les appareils et les applications des modifications et des événements. Vous pouvez utiliser le client MQTT pour vous abonner à ces rubriques et regarder les messages au fur et à mesure qu'ils apparaissent. Vous pouvez également utiliser le client de test MQTT pour publier des messages MQTT sur les appareils et services auxquels vous êtes abonné dans votre Compte AWS.

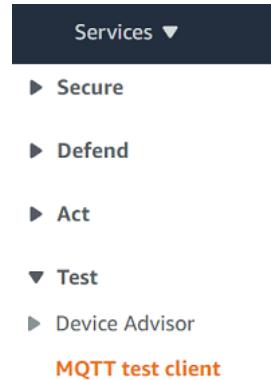
Table des matières

- [Affichage des messages MQTT dans le client MQTT \(p. 71\)](#)
- [Publication de messages MQTT à partir du client MQTT \(p. 73\)](#)
- [Tester les abonnements partagés dans le client MQTT \(p. 74\)](#)

Affichage des messages MQTT dans le client MQTT

Pour afficher les messages MQTT dans le client de test MQTT

1. Dans la [AWS IoTconsole](#), dans le menu de gauche, choisissez Test, puis choisissez le client de test MQTT.



2. Dans l'onglet S'abonner à une rubrique, entrez le *topicName* pour vous abonner à la rubrique sur laquelle votre appareil publie. Pour accéder à l'exemple d'application de mise en route, abonnez-vous à #, qui s'abonne à toutes les rubriques des messages.

Pour continuer avec l'exemple de mise en route, dans l'onglet S'abonner à une rubrique, dans le champ Filtre de rubrique, entrez #, puis choisissez S'abonner.

The screenshot shows the 'Subscribe to a topic' section of the AWS IoT Core developer guide. It features a 'Topic filter' input field with the value '#'. Below it is a link to 'Additional configuration'. At the bottom is a large orange 'Subscribe' button.

La page du journal des messages thématiques, #, s'ouvre et # apparaît dans la liste des abonnements. Si l'appareil que vous avez [the section called “Configurer votre appareil” \(p. 43\)](#) configuré exécute le programme d'exemple, vous devriez voir les messages auxquels il est envoyé AWS IoT dans le journal des messages #. Les entrées du journal des messages apparaîtront sous la section Publier lorsque des messages contenant le sujet auquel vous êtes abonné sont reçus par AWS IoT.

Subscriptions	#	Pause	Clear	Export	Edit
#	♡ X				

- Sur la page # du journal des messages, vous pouvez également publier des messages dans un sujet, mais vous devez spécifier le nom du sujet. Vous ne pouvez pas publier dans la rubrique #.

Les messages publiés sur des sujets auxquels vous êtes abonné apparaissent dans le journal des messages au fur et à mesure de leur réception, avec le message le plus récent en premier.

Dépannage des messages MQTT

Utiliser le filtre de sujet générique

Si vos messages n'apparaissent pas comme prévu dans le journal des messages, essayez de vous abonner à un filtre de sujet générique, comme décrit dans [Filtres de rubrique \(p. 116\)](#). Le filtre de sujet générique à plusieurs niveaux MQTT est le signe dièse ou dièse (#) et peut être utilisé comme filtre de sujet dans le champ Rubrique d'abonnement.

En vous abonnant au filtre de # sujets, vous abonnez à tous les sujets reçus par le courtier de messages. Vous pouvez affiner le filtre en remplaçant les éléments du chemin du filtre de sujet par un caractère générique à # plusieurs niveaux ou par le caractère générique « + » 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 dans le filtre de la catégorie.
- Le chemin du filtre de sujet ne peut comporter qu'un seul caractère générique de niveau par niveau de sujet.

Par exemple :

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

Pour plus d'informations sur les filtres thématiques, consultez[Filtres de rubrique \(p. 116\)](#).

Vérifiez les erreurs de nom de rubrique

Les noms de rubrique et les filtres de rubrique MQTT sont sensibles à la casse. Si, par exemple, votre appareil publie des messages Topic_1 (avec un T majuscule) au lieu du topic_1 sujet auquel vous êtes abonné, ses messages n'apparaîtront pas dans le client de test MQTT. L'abonnement au filtre de rubrique générique indiquerait toutefois que l'appareil publie des messages et que vous pourriez constater qu'il utilise un nom de sujet qui ne correspond pas à celui que vous attendiez.

Publication de messages MQTT à partir du client MQTT

Pour publier un message dans une rubrique MQTT

- Sur la page du client de test MQTT, dans l'onglet Publier dans un sujet, dans le champ **topicName**, **entrez le nom du sujet de votre message**. Pour cet exemple, utilisez **my/topic**.

Note

N'utilisez pas d'informations personnelles identifiables dans les noms des rubriques, que ce soit dans le client de test MQTT ou dans l'implémentation de votre système. Les noms des sujets peuvent apparaître dans les communications et les rapports non chiffrés.

- Dans la fenêtre de charge du message, entrez le JSON suivant :

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

- Choisissez Publier pour publier votre message AWS IoT.

Note

Assurez-vous d'être abonné à la rubrique my/topic avant de publier votre message.

The screenshot shows the 'Publish to a topic' section of the AWS IoT Core console. At the top, there are two tabs: 'Subscribe to a topic' and 'Publish to a topic', with the latter being active. Below the tabs, there is a 'Topic name' field containing 'my/topic'. A message payload editor shows the following JSON:

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

Below the payload editor, there is a link labeled '► Additional configuration'. At the bottom right is a large orange 'Publish' button.

4. Dans la liste des abonnements, choisissez my/topic pour voir le message. Vous devriez voir le message apparaître dans le client de test MQTT en dessous de la fenêtre de chargement du message de publication.

Subscriptions	#	Pause	Clear	Export	Edit
#	Heartbeat				
	▼ my/topic	November 02, 2021, 11:55:22 (UTC-0700)			
	{ "message": "Hello, world", "clientType": "MQTT client" }				

Vous pouvez publier des messages MQTT sur d'autres sujets en modifiant le **topicName dans le champ Nom** du sujet et en cliquant sur le bouton Publier.

Tester les abonnements partagés dans le client MQTT

Cette section explique comment utiliser le client AWS IoT MQTT dans la [AWS IoTconsole](#) pour regarder les messages MQTT envoyés et reçus à AWS IoT l'aide des abonnements partagés. [??? \(p. 103\)](#) autorisez plusieurs clients à partager un abonnement à un sujet, un seul client recevant des messages publiés sur ce sujet en utilisant une distribution aléatoire. Pour simuler plusieurs clients MQTT (dans cet exemple, deux clients MQTT) partageant le même abonnement, vous ouvrez le client AWS IoT MQTT dans la [AWS IoTconsole](#) à partir de plusieurs navigateurs Web. L'exemple utilisé dans cette section ne correspond pas aux exemples utilisés dans [Démarrer avec AWS IoT Core \(p. 18\)](#). Pour d'd'd'd'd'd'd'd'd'd'

Pour partager un abonnement à une rubrique MQTT

1. Dans la [AWS IoTconsole](#), dans le volet de navigation, choisissez Test, puis choisissez le client de test MQTT.

- Dans l'onglet S'abonner à une rubrique, entrez le **topicName** pour vous abonner à la rubrique sur laquelle votre appareil publie. Pour utiliser les abonnements partagés, abonnez-vous au filtre de sujets d'un abonnement partagé comme suit :

```
$share/{ShareName}/{TopicFilter}
```

Un exemple de filtre de sujet peut être **\$share/group1/topic1** celui qui s'abonne à la rubrique **topic1** du message.

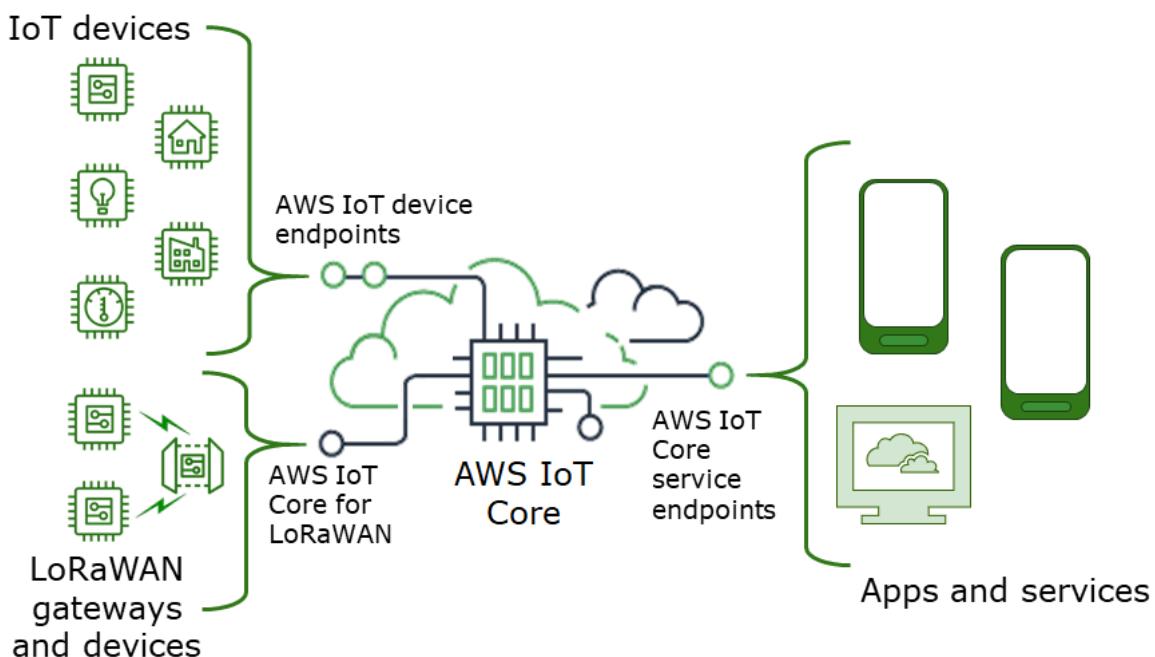
The screenshot shows the 'Subscribe to a topic' section of the AWS IoT Core console. It includes a 'Topic filter' input field where '\$share/group1/topic1' is entered, and a large orange 'Subscribe' button below it.

- Ouvrez un autre navigateur Web et répétez les étapes 1 et 2. De cette façon, vous simulez deux clients MQTT différents qui partagent le même abonnement. **\$share/group1/topic1**
- Choisissez un client MQTT, dans l'onglet Publier dans un sujet, dans le champ **topicName**, entrez **le nom du sujet de votre message**. Pour cet exemple, utilisez **topic1**. Essayez de publier le message plusieurs fois. Dans la liste des abonnements des deux clients MQTT, vous devriez pouvoir voir que les clients reçoivent le message en utilisant une distribution aléatoire. Dans cet exemple, nous publions trois fois le même message « Bonjour depuis AWS IoT la console ». Le client MQTT de gauche a reçu le message deux fois et le client MQTT de droite l'a reçu une fois.

The screenshot displays two separate MQTT publishing sessions. Both are configured with the same topic filter '\$share/group1/topic1'. The left session shows a successful publish attempt with the message payload '{ "message": "Hello from AWS IoT console" }'. The right session shows a pending publish attempt with the same message payload. Both sessions include a large orange 'Publish' button.

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 appareils se connectent AWS IoT Core afin de pouvoir envoyer des données vers des AWS IoT services et d'autres appareils et en recevoir de la part de ces derniers. Des applications et d'autres services se connectent également AWS IoT Core pour contrôler et gérer les appareils IoT et traiter les données de votre solution IoT. Cette section explique comment choisir le meilleur moyen de connexion et de communication AWS IoT Core pour chaque aspect de votre solution IoT.



Il existe plusieurs méthodes pour interagir avec AWS IoT. Les applications et les services peuvent utiliser le [AWS IoT Core- points d'extrémité du plan de contrôle \(p. 76\)](#) et les appareils peuvent se connecter AWS IoT Core à l'aide du [AWS IoT points de terminaison de l'appareil \(p. 77\)](#) ou [AWS IoT Core pour passerelles et appareils LoRa WAN \(p. 78\)](#).

AWS IoT Core- points d'extrémité du plan de contrôle

Les AWS IoT Core points de terminaison du plan de contrôle permettent d'accéder aux fonctions qui contrôlent et gèrent votre AWS IoT solution.

- Points de terminaison

Les points d'extrémité du plan de contrôle AWS IoT Core - et du plan de contrôle AWS IoT Core Device Advisor sont spécifiques à la région et sont répertoriés dans [AWS IoT CorePoints de terminaison et quotas](#). Les formats des points de terminaison sont les suivants.

Objectif du terminaison	Format du point de terminaison	Sert
AWS IoT Core- plan de contrôle	<code>iot.aws-region.amazonaws.com</code>	AWS IoTAPI du plan de contrôle
AWS IoT CoreDevice Advisor - plan de contrôle	<code>api.iotdeviceadvisor.aws-region.amazonaws.com</code>	AWS IoT CoreAPI Device Advisor Control Plane

- Kit SDK et outils

Les [AWSkits SDK](#) fournissent un support spécifique au langage pour les AWS IoT Core API et les API d'autres services. AWS Les [kits SDK AWS mobiles](#) fournissent aux développeurs d'applications un support spécifique à la plate-forme pour l'AWS IoT CoreAPI et d'autres AWS services sur les appareils mobiles.

[AWS CLI](#) Fournit un accès en ligne de commande aux fonctions fournies par les points de terminaison du AWS IoT service. [AWS Tools for PowerShell](#) fournit des outils permettant de gérer les AWS services et les ressources dans l'environnement PowerShell de script.

- Authentification

Les points de terminaison du service utilisent les utilisateurs et les AWS informations d'identification IAM pour authentifier les utilisateurs.

- En savoir plus

Pour plus d'informations et pour des liens vers des références au SDK, consultez[the section called "Connexion aux points de terminaison AWS IoT Core de service" \(p. 79\)](#).

AWS IoTpoints de terminaison de l'appareil

Les points de terminaison des AWS IoT appareils prennent en charge la communication entre vos appareils IoT et AWS IoT.

- Points de terminaison

Les terminaux, le support AWS IoT Core et les AWS IoT Device Management fonctions de l'appareil. Ils sont spécifiques à votre Compte AWS et vous pouvez les voir à l'aide de la [describe-endpoint](#) commande.

Objectif du terminaison	Format du point de terminaison	Sert
AWS IoT Core- plan de données	Consultez ??? (p. 85).	AWS IoTAPI Data Plane
AWS IoT Device Management-données sur les emplois	Consultez ??? (p. 85).	AWS IoTAPI Jobs Data Plane
AWS IoTDevice Advisor - plan de données	Consultez ??? (p. 1172).	
AWS IoT Device Management-Centre de flotte	Ne s'applique pas.	Ne s'applique pas.
AWS IoT Device Management-tunneling sécurisé	<code>api.tunneling.iot.aws-region.amazonaws.com</code>	AWS IoTAPI de tunneling sécurisé

Pour plus d'informations sur ces terminaux et les fonctions qu'ils prennent en charge, consultez [the section called "AWS IoT données de l'appareil et points de terminaison de service" \(p. 85\)](#).

- Kits SDK

Les [kits SDK pour AWS IoT appareils \(p. 87\)](#) fournissent un support spécifique au langage pour les protocoles MQTT (Message Queueing Telemetry Transport) et WebSocket Secure (WSS), que les appareils utilisent pour communiquer avec AWS IoT. Les [Kits SDK AWS Mobile \(p. 84\)](#) fournissent également un support pour les communications entre appareils MQTT, AWS IoT les API et les API d'autres AWS services sur les appareils mobiles.

- Authentification

Les terminaux des appareils utilisent des certificats X.509 ou des utilisateurs AWS IAM dotés d'informations d'identification pour authentifier les utilisateurs.

- En savoir plus

Pour plus d'informations et pour des liens vers des références au SDK, consultez [the section called "Kits SDK pour les appareils AWS IoT" \(p. 87\)](#).

AWS IoT Core pour passerelles et appareils LoRa WAN

AWS IoT Core pour le LoRa WAN connecte les passerelles et les appareils sans fil à AWS IoT Core.

- Points de terminaison

AWS IoT Core pour LoRa WAN gère les connexions de passerelle aux points de terminaison spécifiques au compte et à la région AWS IoT Core. Les passerelles peuvent se connecter au point de terminaison du serveur de configuration et de mise à jour (CUPS) de votre compte fourni AWS IoT Core pour le LoRa WAN.

Objectif du terminaison	Format du point de terminaison	Sert
Serveur de configuration et de mise à jour (CUPS)	<i>account-specific-prefix.cups.lorawan.aws-region.amazonaws.com:443</i>	Communication par passerelle avec le serveur de configuration et de mise à jour fourni par AWS IoT Core for LoRa WAN
LoRaServeur réseau WAN (LNS)	<i>account-specific-prefix.gateway.lorawan.aws-region.amazonaws.com:443</i>	Communication par passerelle avec le serveur réseau LoRa WAN fourni par AWS IoT Core for LoRa WAN

- Kits SDK

L'API AWS IoT sans fil sur laquelle AWS IoT Core repose le LoRa WAN est prise en charge par le AWS SDK. Pour de plus amples informations, veuillez consulter [Kits AWSSDK et boîtes à outils](#).

- Authentification

AWS IoT Core pour les communications entre appareils LoRa WAN, utilisez des certificats X.509 pour sécuriser les communications avec AWS IoT.

- En savoir plus

Pour de plus amples informations sur la configuration et la connexion des appareils sans fil, consultez [AWS IoT pour LoRa WAN \(p. 1272\)](#).

Connexion aux points de terminaison AWS IoT Core de service

Vous pouvez accéder aux fonctionnalités du AWS IoT Core plan de contrôle en utilisant le AWS CLI AWS SDK correspondant à votre langue préférée ou en appelant directement l'API REST. Nous vous recommandons d'utiliser le SDK AWS CLI ou un AWS SDK avec lequel interagir, AWS IoT Core car ils intègrent les meilleures pratiques en matière de AWS services d'appel. Il est possible d'appeler directement les API REST, mais vous devez fournir [les informations de sécurité nécessaires](#) pour accéder à l'API.

Note

Les appareils IoT doivent utiliser [Kits SDK pour les appareils AWS IoT \(p. 87\)](#). Les kits SDK pour appareils sont optimisés pour une utilisation sur des appareils, prennent en charge la communication MQTT avec AWS IoT les appareils et prennent en charge les AWS IoT API les plus utilisées par les appareils. Pour de plus amples informations sur les kits SDK for the Device et les fonctions qu'ils leur fournissent, consultez [Kits SDK pour les appareils AWS IoT \(p. 87\)](#).

Les appareils mobiles doivent utiliser [Kits SDK AWS Mobile \(p. 84\)](#). Les kits SDK mobiles prennent en charge les AWS IoT API, les communications entre appareils MQTT et les API d'autres AWS services sur les appareils mobiles. Pour de plus amples informations sur les kits SDK Mobile et les fonctions qu'ils leur fournissent, consultez [Kits SDK AWS Mobile \(p. 84\)](#).

Vous pouvez utiliser AWS Amplify les outils et les ressources des applications Web et mobiles pour vous connecter plus facilement à AWS IoT Core. Pour plus d'informations sur la connexion à l'aide AWS IoT Core d'Amplify, consultez [Pub Sub Getting Started](#) dans la documentation d'Amplify.

Les sections suivantes décrivent les outils et les kits SDK que vous pouvez utiliser pour développer et interagir avec AWS IoT d'autres AWS services. Pour obtenir la liste complète des AWS outils et des kits de développement disponibles pour créer et gérer des applications AWS, consultez la section [Outils de création AWS](#).

AWS CLI pour AWS IoT Core

AWS CLIFournit un accès en ligne de commande aux API. AWS

- Installation

Pour plus d'informations sur l'installation du AWS CLI, consultez la section [Installation du AWS CLI](#).

- Authentification

Les AWS CLI utilisateurs utilisent les informations d'identification de votre Compte AWS.

- Référence

Pour plus d'informations sur les AWS CLI commandes de ces AWS IoT Core services, consultez :

- [AWS CLIRéférence de commande pour l'IoT](#)
- [AWS CLIRéférence de commande pour les données IoT](#)
- [AWS CLIRéférence de commande pour les données des tâches liées à l'IoT](#)
- [AWS CLIRéférence de commande pour le tunneling sécurisé de l'IoT](#)

Pour les outils permettant de gérer les AWS services et les ressources dans l'environnement PowerShell de script, consultez la section [AWSOutils pour PowerShell](#).

Kits de développement logiciel (SDK) AWS

Avec AWS les kits SDK, vos applications et appareils compatibles peuvent appeler AWS IoT des API et les API d'autres AWS services. Cette section fournit des liens vers les AWS kits SDK et vers la documentation de référence des API pour les API des AWS IoT Core services.

Les AWS SDK prennent en charge ces API AWS IoT Core

- [AWS IoT](#)
- [AWS IoTPlan de données](#)
- [AWS IoTPlan de données sur les emplois](#)
- [AWS IoTTunneling sécurisé](#)
- [AWS IoTsans fil](#)

C++

Pour installer le [AWS SDK for C++](#)et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la section [Mise en route avec le AWS SDK for C++](#)

Ces instructions décrivent comment :

- Installation et génération du SDK à partir des fichiers source
- Fournissez les informations d'identification nécessaires pour utiliser le SDK avec votre Compte AWS
- Initialisez et arrêtez le SDK de votre application ou service
- Créez un projet CMake pour créer votre application ou votre service

2. Créez et exécutez un exemple d'application. Pour des exemples d'applications utilisant le AWS SDK for C++, consultez la section [Exemples de AWS SDK for C++ code](#).

Documentation relative aux AWS IoT Core services pris en AWS SDK for C++ charge

- [AWSDocumentation de référence : :IoTClient »](#)
- [Documentation de référence Aws : :IoT DataPlane : :IoT DataPlaneClient](#)
- [Documentation de référence Aws : :IoT JobsDataPlane : :IoT JobsDataPlaneClient](#)
- [Documentation de référence Aws : :IoT SecureTunneling : :IoT SecureTunnelingClient](#)

Go

Pour installer le [AWS SDK for Go](#)et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la [section Mise en route du AWS SDK for Go](#)

Ces instructions décrivent comment :

- Installation de l'AWS SDK for Go
- Obtenez les clés d'accès pour que le SDK accède à votre Compte AWS
- Importer des packages dans le code source de nos applications ou services

2. Créez et exécutez un exemple d'application. Pour des exemples d'applications utilisant leAWS SDK for Go, voir [Exemples de AWS SDK for Go code](#).

Documentation relative aux AWS IoT Core services pris en AWS SDK for Go charge

- [Documentation de référence sur l'IoT](#)
- [Documentation de DataPlane référence sur l'IoT](#)
- [Documentation de JobsDataPlane référence sur l'IoT](#)
- [Documentation de SecureTunneling référence sur l'IoT](#)

Java

Pour installer le [AWS SDK for Java](#) et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la [section Commencer avec AWS SDK for Java 2.x](#)

Ces instructions décrivent comment :

- S'inscrire AWS et créer un utilisateur IAM
- Téléchargement du kit SDK
- Configurer les AWS informations d'identification et la région
- Utiliser le SDK avec Apache Maven
- Utiliser le SDK avec Gradle

2. Créez et exécutez un exemple d'application à l'aide de l'un des [exemples de AWS SDK for Java 2.x code](#).

3. Consultez la documentation de [référence de l'API du SDK](#)

Documentation relative aux AWS IoT Core services pris en AWS SDK for Java charge

- [IoClientdocumentation de référence](#)
- [IoDataPlaneClientdocumentation de référence](#)
- [IoJobsDataPlaneClientdocumentation de référence](#)
- [Documentation de SecureTunnelingClient référence sur l'IoT](#)

JavaScript

Pour installer le [AWS SDK for JavaScript](#) et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la [section Configuration du AWS SDK for JavaScript](#). Ces instructions s'appliquent à l'utilisation de AWS SDK for JavaScript dans le navigateur et avec Node.JS. Assurez-vous de suivre les instructions qui s'appliquent à votre installation.

Ces instructions décrivent comment :

- Vérifiez les conditions préalables
- Installez le SDK pour JavaScript
- Chargez le SDK pour JavaScript

2. Créez et exécutez un exemple d'application pour commencer à utiliser le SDK, comme le décrit l'option de démarrage pour votre environnement.

- Commencez à utiliser le [AWSSDK pour JavaScript dans le navigateur](#), ou
- Commencez à utiliser le [AWSSDK pour JavaScript dans Node.js](#)

Documentation relative aux AWS IoT Core services pris en AWS SDK for JavaScript charge

- [AWS.Iot reference documentation](#)

- [AWS.IotData reference documentation](#)
- [AWS.IotJobsDataPlane reference documentation](#)
- [AWS.IotSecureTunneling reference documentation](#)

.NET

Pour installer le [AWS SDK for .NET](#) et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la [section Configuration de votre AWS SDK for .NET environnement](#)
2. Suivez les instructions de la [section Configuration de votre AWS SDK for .NET projet](#)

Ces instructions décrivent comment :

- Démarrer un nouveau projet.
- Obtenir et configurer les AWS informations d'identification
- Installation des AWS packages du SDK

3. Créez et exécutez l'un des exemples de programmes présentés dans la [section Utilisation des AWS services dans le AWS SDK for .NET](#)
4. Consultez la documentation de [référence de l'API du SDK](#)

Documentation relative aux AWS IoT Core services pris en AWS SDK for .NET charge

- [Documentation de référence Amazon.IOT.Model](#)
- [Amazon.IotData.Documentation de référence du modèle](#)
- [Documentation de référence sur Amazon.IoT JobsDataPlane .Model](#)
- [Documentation de référence sur Amazon.IoT SecureTunneling .Model](#)

PHP

Pour installer le [AWS SDK for PHP](#) et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions de la [section Mise en route de la AWS SDK for PHP version 3](#)

Ces instructions décrivent comment :

- Vérifiez les conditions préalables
- Installer le SDK
- Appliquer le SDK à un script PHP

2. Créez et exécutez un exemple d'application à l'aide de l'un des [exemples de code de la AWS SDK for PHP version 3](#)

Documentation relative aux AWS IoT Core services pris en AWS SDK for PHP charge

- [Documentation de référence du client IoT](#)
- [Documentation de DataPlaneClient référence sur l'IoT](#)
- [Documentation de JobsDataPlaneClient référence sur l'IoT](#)
- [Documentation de SecureTunnelingClient référence sur l'IoT](#)

Python

Pour installer le [AWS SDK for Python \(Boto3\)](#) et l'utiliser pour vous connecter à AWS IoT :

1. Suivez les instructions du [AWS SDK for Python \(Boto3\)Quickstart](#)

Ces instructions décrivent comment :

- Installer le SDK
- Configuration du kit SDK
- Utilisez le SDK dans votre code

2. Créez et exécutez un exemple de programme qui utilise AWS SDK for Python (Boto3)

Ce programme affiche les options de journalisation actuellement configurées pour le compte. Après avoir installé le SDK et l'avoir configuré pour votre compte, vous devriez être en mesure d'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 de plus amples informations sur la fonction utilisée dans cet exemple, consultez[the section called "Configurer la journalisation AWS IoT" \(p. 467\)](#).

Documentation relative aux AWS IoT Core services pris en AWS SDK for Python (Boto3) charge

- [Documentation de référence sur l'IoT](#)
- [Documentation de DataPlane référence sur l'IoT](#)
- [Documentation de JobsDataPlane référence sur l'IoT](#)
- [Documentation de SecureTunneling référence sur l'IoT](#)

Ruby

Pour installer le [AWS SDK for Ruby](#) et l'utiliser pour vous connecter à AWS IoT :

- Suivez les instructions de la [section Mise en route du AWS SDK for Ruby](#)

Ces instructions décrivent comment :

- Installer le SDK
- Configuration du kit SDK
- Création et exécution du [didacticiel Hello World](#)

Documentation relative aux AWS IoT Core services pris en charge par le AWS SDK for Ruby

- [Aws::IoT::Client : Documentation de référence pour les clients](#)
- [Aws::IoTDataPlane::Client : Documentation de référence pour les clients](#)
- [Aws::IoTJobsDataPlane::Client : Documentation de référence pour les clients](#)
- [Aws::IoTSecureTunneling::Client : Documentation de référence pour les clients](#)

Kits SDK AWS Mobile

Les kits SDK AWS mobiles fournissent aux développeurs d'applications mobiles un support spécifique à la plate-forme pour les API des AWS IoT Core services, la communication entre appareils IoT à l'aide de MQTT et les API d'autres services AWS.

Android

AWS Mobile SDK for Android

AWS Mobile SDK for AndroidII contient une bibliothèque, des exemples et de la documentation permettant aux développeurs de créer des applications mobiles connectées à l'aide deAWS. Ce SDK inclut également la prise en charge des communications entre appareils MQTT et l'appel des API des AWS IoT Core services. Pour plus d'informations, consultez les ressources suivantes :

- [AWSKit SDK Mobile pour Android sur GitHub](#)
- [AWSKit SDK Mobile — Readme](#)
- [AWSKit SDK Mobile pour Android](#)
- [AWSRéférence de l'API du SDK for Android](#)
- [AWSIoTClientDocumentation de référence pour les classes](#)

iOS

AWS Mobile SDK for iOS

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

- [AWS Mobile SDK for iOS sur GitHub](#)
- [AWSSDK for iOS Readme](#)
- [AWSKit SDK for iOS](#)
- [AWS IoTDocuments de référence relatifs aux classes dans le AWS SDK for iOS](#)

API REST des AWS IoT Core services

Les API REST des AWS IoT Core services peuvent être appelées directement à l'aide de requêtes HTTP.

- URL de point de terminaison

Les points de terminaison de service qui exposent les API REST des AWS IoT Core services varient selon la région et sont répertoriés dans [AWS IoT CorePoints de terminaison et quotas](#). Vous devez utiliser le point de terminaison de la région qui possède les AWS IoT ressources auxquelles vous souhaitez accéder, car les AWS IoT ressources sont spécifiques à la région.

- Authentification

Les API REST des AWS IoT Core services utilisent des informations d'identification AWS IAM pour l'authentification. Pour de plus amples informations, veuillez consulter [Signature des demandes d'AWSAPI](#) dans le AWS General Reference.

- Référence d'API

Pour plus d'informations sur les fonctions spécifiques fournies par les API REST des AWS IoT Core services, consultez :

- [Référence d'API pour l'IoT.](#)
- [Référence d'API pour les données IoT.](#)
- [Référence d'API pour les données de tâches liées à l'IoT.](#)
- [Référence d'API pour le tunneling sécurisé de l'IoT.](#)

Connexion d'appareils à AWS IoT

Les appareils se connectent à d'autres services AWS IoT et par le biais de ceux-ciAWS IoT Core. ViaAWS IoT Core, les appareils envoient et reçoivent des messages à l'aide de terminaux spécifiques à votre compte. Les communications de l'appareil de [the section called "Kits SDK pour les appareils AWS IoT" \(p. 87\)](#) support à l'aide des protocoles MQTT et WSS. Pour de plus amples informations sur les protocoles que les appareils peuvent utiliser, consultez[the section called "Protocoles de communication des appareils" \(p. 89\).](#)

Le courtier de messages

AWS IoT gère les communications entre les appareils via un courtier de messages. Les appareils et les clients publient des messages sur le courtier de messages et s'abonnent également aux messages publiés par le courtier de messages. [Les messages sont identifiés par un sujet défini par l'application. \(p. 115\)](#) Lorsque l'agent de messagerie reçoit un message publié par un appareil ou un client, il le republie sur les appareils et les clients qui se sont abonnés à la rubrique du message. Le courtier de messages transmet également les messages au moteur de AWS IoT [règles \(p. 524\)](#), qui peut agir sur le contenu du message.

AWS IoT sécurité des messages

Connexions des appareils à AWS IoT utiliser [the section called "Certificats client X.509" \(p. 320\)](#) et [AWSsignature V4](#) pour l'authentification. Les communications entre appareils sont sécurisées par la version 1.2 du protocole TLS et AWS IoT nécessitent que les appareils envoient l'[extension SNI \(Server Name Indication\)](#) lorsqu'ils se connectent. Pour de plus amples informations, veuillez consulter [Sécurité du transport dans AWS IoT](#).

AWS IoT données de l'appareil et points de terminaison de service

Important

Vous pouvez mettre en cache ou stocker les points de terminaison sur votre appareil. Cela signifie que vous n'aurez pas besoin d'interroger l'`DescribeEndpoint` API à chaque fois qu'un nouvel appareil est connecté. Les points de terminaison ne changeront pas AWS IoT Core une fois qu'ils auront été créés pour votre compte.

Chaque compte possède plusieurs points de terminaison qui lui sont propres et qui prennent en charge des fonctions IoT spécifiques. Les points de terminaison de données des AWS IoT appareils prennent en charge un protocole de publication/d'abonnement conçu pour répondre aux 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 points de terminaison. Les AWS IoT points de terminaison de service des appareils prennent en charge l'accès centré sur l'appareil aux services de sécurité et de gestion.

Pour connaître le point de terminaison des données de l'appareil de votre compte, vous pouvez le trouver sur la page [Paramètres](#) de votre AWS IoT Core console.

Pour connaître le point de terminaison de l'appareil de votre compte dans un but spécifique, y compris le point de terminaison des données de l'appareil, utilisez la commande `describe-endpoint` CLI illustrée ici,

ou l'DescribeEndpointAPI REST, et fournissez la valeur du *endpointType* paramètre dans le tableau suivant.

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

Cette commande renvoie un point de *terminaison IoT* au format suivant : *account-specific-prefix*.iot.*aws-region*.amazonaws.com

Chaque client possède un point de terminaison *iot:Data-ATS* et un *iot:Data* point de terminaison. 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* point de terminaison permettant aux appareils de récupérer des données à partir d'anciens points de terminaison qui utilisent VeriSign des certificats à des fins de rétrocompatibilité. Pour de plus amples informations, veuillez consulter [Authentification du serveur](#).

AWS IoT points de terminaison pour appareils

Objectif du terminaison	<i>endpointType</i> valeur	Description
AWS IoT Core- opérations du plan de données	<i>iot:Data-ATS</i>	<p>Utilisé pour envoyer et recevoir des données vers et depuis le courtier de messages, Device Shadow (p. 690) et les composants du moteur de règles (p. 524) de AWS IoT.</p> <p><i>iot:Data-ATS</i> renvoie un point de terminaison de données signé ATS.</p>
AWS IoT Core- opérations du plan de données (ancienne)	<i>iot:Data</i>	<p><i>iot:Data</i> renvoie un point de terminaison de données VeriSign signé fourni à des fins de rétrocompatibilité. MQTT 5 n'est pas pris en charge sur les terminaux Symantec (<i>iot:Data</i>).</p>
AWS IoT Core accès aux informations d'identification	<i>iot:CredentialProvider</i>	<p>Utilisé pour échanger le certificat X.509 intégré d'un appareil contre des informations d'identification temporaires permettant de se connecter directement à d'autres AWS services. Pour de plus amples informations sur la connexion à d'autres services AWS, veuillez consulter Autorisation des appels directs vers les services AWS (p. 402).</p>
AWS IoT Device Management- opérations de données sur les emplois	<i>iot:Jobs</i>	<p>Utilisé pour permettre aux appareils d'interagir avec le service AWS IoT Jobs à l'aide des API HTTPS de Jobs Device (p. 824).</p>

Objectif du terminaison	<i>endpointType</i> valeur	Description
AWS IoT Fonctionnement de Device Advisor	iot:DeviceAdvisor	Type de point de terminaison de test utilisé pour tester les appareils avec Device Advisor. Pour plus d'informations, veuillez consulter ??? (p. 1167) .
AWS IoT CoreData bêta (aperçu)	iot:Data-Beta	Type de point de terminaison réservé aux versions bêta. Pour plus d'informations sur son utilisation actuelle, reportez-vous à la section ??? (p. 131) .

Vous pouvez également utiliser votre propre nom de domaine complet (FQDN), tel que example.com, et le certificat de serveur associé pour connecter des appareils à AWS IoT l'aide de. [the section called “Points de terminaison configurables” \(p. 131\)](#)

Kits SDK pour les appareils AWS IoT

Les kits SDK pour AWS IoT appareils vous aident à connecter vos appareils IoT aux protocoles MQTT AWS IoT Core et MQTT over WSS et les prennent en charge.

Les kits SDK pour AWS IoT appareils diffèrent des AWS kits SDK en ce sens qu'ils répondent aux besoins de communication spécialisés des appareils IoT, mais ne prennent pas en charge tous les services pris en charge par les kits SDK. AWS IoT AWS Les kits SDK pour AWS IoT appareils sont compatibles avec les AWS kits SDK qui prennent en charge tous les AWS services ; toutefois, ils utilisent différentes méthodes d'authentification et se connectent à différents points de terminaison, ce qui peut rendre l'utilisation des AWS kits SDK peu pratique sur un appareil IoT.

Appareils mobiles

Ils [the section called “Kits SDK AWS Mobile” \(p. 84\)](#) prennent en charge à la fois les communications entre appareils MQTT, certaines API de AWS IoT service et les API d'autres AWS services. Si vous développez sur un appareil mobile compatible, consultez son SDK pour voir s'il s'agit de la meilleure option pour développer votre solution IoT.

C++

AWS IoTSDK pour appareils C++

Le SDK AWS IoT C++ Device permet aux développeurs de créer des applications connectées à AWS l'aide des API des 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 plus d'informations, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil C++ v2 activé GitHub](#)
- [AWS IoTKit de développement logiciel pour appareils C++ v2 Readme](#)
- [AWS IoTExemples de kits de développement logiciel pour appareils C++ v2](#)
- [AWS IoTDocumentation de l'API C++ v2 du SDK pour appareils](#)

Python

AWS IoTSDK de l'appareil pour Python

Le AWS IoT Device SDK pour Python permet aux développeurs d'écrire des scripts Python afin d'utiliser leurs appareils pour accéder à la AWS IoT plateforme via MQTT ou MQTT via le protocole WebSocket Secure (WSS). En connectant leurs appareils aux API des AWS IoT Core services, les utilisateurs peuvent travailler en toute sécurité avec le courtier de messages, les règles et le service Device Shadow qui AWS IoT Core fournit ces AWS services AWS Lambda, ainsi qu'avec d'autres services tels qu'Amazon Kinesis et Amazon S3, etc.

- [AWS IoTSDK de l'appareil pour Python v2 sur GitHub](#)
- [Kit SDK des appareils AWS IoT pour Python v2 - Readme](#)
- [AWS IoTExemples de SDK de périphérique pour Python v2](#)
- [AWS IoTDocumentation de l'API Device SDK pour Python v2](#)

JavaScript

AWS IoTSDK de l'appareil pour JavaScript

Le AWS IoT Device SDK pour JavaScript permet aux développeurs d'écrire des JavaScript applications qui accèdent aux API AWS IoT Core utilisant le MQTT ou le MQTT via le protocole. WebSocket Il peut être utilisé dans des environnements Node.js et des applications de navigateur. Pour plus d'informations, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil pour JavaScript v2 activé GitHub](#)
- [AWS IoTSDK de l'appareil pour le fichier JavaScript Readme v2](#)
- [AWS IoTExemples de SDK d'appareils pour la JavaScript version 2](#)
- [AWS IoTDocumentation de l'API Device SDK pour JavaScript v2](#)

Java

AWS IoTSDK de l'appareil pour Java

Le AWS IoT Device SDK for Java permet aux développeurs Java d'accéder aux API du AWS IoT Core via MQTT ou MQTT via le protocole. WebSocket Le 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 en utilisant des méthodes getter et setter, sans avoir à sérialiser ou déserialiser des documents JSON. Pour plus d'informations, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil pour Java v2 activé GitHub](#)
- [AWS IoT Kit SDK des périphériques pour Java v2 – Readme](#)
- [AWS IoTKit SDK SDK for Java v2](#)
- [AWS IoTDocumentation sur l'API du Device pour Java v2](#)

Embedded C

AWS IoTSDK de périphérique pour Embedded C

Important

Ce SDK est destiné à être utilisé par des développeurs expérimentés de logiciels embarqués.

Le Kit SDK des appareils AWS IoT pour Embedded C (C-SDK) est une collection de fichiers source C sous la licence open source du MIT qui peuvent être utilisés dans des applications intégrées pour connecter en toute sécurité des appareils IoT à AWS IoT Core. Il inclut les bibliothèques MQTT,

JSON Parser et AWS IoT Device Shadow, entre autres. Il est distribué sous forme source et destiné à être intégré au microprogramme du client avec le code de l'application, d'autres bibliothèques et, éventuellement, un RTOS (Real Time Operating System).

Le Kit SDK des appareils AWS IoT pour 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, nous vous recommandons d'utiliser l'un des autres kits de développement logiciel pour AWS IoT appareils et AWS IoT appareils mobiles, tels que le SDK for C++JavaScript, Java ou Python.

Pour plus d'informations, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil pour Embedded C sur GitHub](#)
- [Kit SDK des appareils AWS IoT pour Embedded C – Readme](#)
- [AWS IoTSDK de périphérique pour les échantillons C intégrés](#)

Protocoles de communication des appareils

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

TLS 1.2 et TLS 1.3

AWS IoT Core utilise [les versions TLS 1.2](#) et [TLS 1.3](#) pour crypter toutes les communications. Les clients doivent également envoyer [l'extension TLS \(SNI\)](#). 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](#).

Ils [Kits SDK pour les appareils AWS IoT \(p. 87\)](#) prennent en charge le MQTT et le MQTT via WSS et répondent aux exigences de sécurité des connexions client. Nous vous recommandons d'utiliser le [Kits SDK pour les appareils AWS IoT \(p. 87\)](#) pour connecter les clients à AWS IoT.

Protocoles, mappages de ports et authentification

La manière dont un appareil ou un client se connecte au courtier de messages à l'aide d'un point de terminaison dépend du protocole utilisé. Le tableau suivant répertorie les protocoles pris en charge par les terminaux des AWS IoT appareils, ainsi que les méthodes d'authentification et les ports qu'ils utilisent.

Protocoles, authentification et mappages de port

Protocole	Opérations prises en charge	Authentification	Port	Nom du protocole ALPN
MQTT terminé WebSocket	Publier, s'abonner	Signature Version 4	443	N/A
MQTT terminé WebSocket	Publier, s'abonner	Authentification personnalisée	443	N/A
MQTT	Publier, s'abonner	Certificat de client X.509	443 [†]	x-amzn-mqtt-ca
MQTT	Publier, s'abonner	Certificat de client X.509	8883	N/A

Protocole	Opérations prises en charge	Authentification	Port	Nom du protocole ALPN
MQTT	Publier, s'abonner	Authentification personnalisée	443 [†]	mqtt
HTTPS	Publier uniquement	Signature Version 4	443	N/A
HTTPS	Publier uniquement	Certificat de client X.509	443 [†]	x-amzn-http-ca
HTTPS	Publier uniquement	Certificat de client X.509	8443	N/A
HTTPS	Publier uniquement	Authentification personnalisée	443	N/A

Négociation du protocole de couche d'application (ALPN)

[†] Les clients qui se connectent sur le port 443 avec une authentification par certificat client X.509 doivent implémenter l'extension TLS [ALPN \(Application Layer Protocol Negotiation\)](#) et utiliser le [nom du protocole ALPN](#) répertorié dans l'ALPN ProtocolNameList envoyé par le client dans le cadre du message. `ClientHello`

[Sur le port 443, le point de terminaison IoT:Data-ATS \(p. 86\) prend en charge le x-amzn-http-ca protocole HTTP ALPN, mais pas le point de terminaison IoT:Jobs. \(p. 86\)](#)

Sur le port 8443 HTTPS et le port 443 MQTT avec ALPNx-amzn-mqtt-ca, [l'authentification personnalisée \(p. 343\)](#) ne peut pas être utilisée.

Les clients se connectent aux terminaux Compte AWS de leurs appareils. Consultez [the section called "AWS IoT données de l'appareil et points de terminaison de service" \(p. 85\)](#) la section pour savoir comment trouver les terminaux de votre compte.

Note

AWS Les SDK n'ont pas besoin de l'URL complète. Ils n'ont besoin que du nom d'hôte du point de terminaison, tel que [l'exemple pour AWS IoT Device SDK for Python](#) on GitHub La transmission de l'URL complète comme indiqué dans le tableau suivant peut générer une erreur telle qu'un nom d'hôte non valide.

Connexion à AWS IoT Core

Protocole	Point de terminaison ou URL
MQTT	<i>iot-endpoint</i>
MQTT sur WSS	wss:// <i>iot-endpoint</i> /mqtt
HTTPS	https:// <i>iot-endpoint</i> /topics

Choix d'un protocole pour la communication de votre appareil

Pour la plupart des communications entre appareils IoT via les points de terminaison des appareils, vous souhaiterez utiliser les protocoles MQTT ou MQTT sur WSS ; toutefois, les points de terminaison des appareils prennent également en charge le protocole HTTPS. Le tableau suivant compare l'AWS IoT Core utilisation des deux protocoles pour la communication entre appareils.

AWS IoT protocoles de l'appareil side-by-side

Fonction	MQTT (p. 92)	HTTPS (p. 112)
Aide à la publication/à l'abonnement	Publier et s'abonner	Publier uniquement
Prise en charge de SDK	AWS Les kits SDK pour appareils prennent en charge les (p. 87) protocoles MQTT et WSS	Aucun support du SDK, mais vous pouvez utiliser des méthodes spécifiques au langage pour effectuer des requêtes HTTPS
Support en matière de qualité de service	Niveaux de QoS MQTT 0 et 1 (p. 94)	La QoS est prise en charge en transmettant un paramètre de chaîne de requête ?qos=qos dont la valeur peut être 0 ou 1. Vous pouvez ajouter cette chaîne de requête pour publier un message avec la valeur de QoS souhaitée.
La réception de messages peut être manquée alors que l'appareil était hors ligne	Oui	Non
clientIds soutien sur le terrain	Oui	Non
Détection de déconnexion de l'appareil	Oui	Non
Communications sécurisées	Oui. Consultez Protocoles, mappages de ports et authentification (p. 89).	Oui. Consultez Protocoles, mappages de ports et authentification (p. 89).
Définitions de sujet	Application définie	Application définie
Format des données du message	Application définie	Application définie
Surcharge du protocole	Inférieur	Plus haut
Consommation d'énergie	Inférieur	Plus haut

Limites de durée de la connexion

Il n'est pas garanti que les connexions HTTPS durent plus longtemps que le temps nécessaire pour recevoir les demandes et y répondre.

La durée de connexion MQTT dépend de la fonction d'authentification que vous utilisez. Le tableau suivant répertorie la durée de connexion maximale dans des conditions idéales pour chaque fonction.

Durée de connexion MQTT par fonction d'authentification

Fonction	Durée ^{maximale*}
Certificat de client X.509	1 à 2 semaines
Authentification personnalisée	1 à 2 semaines

Fonction	Durée maximale*
Signature Version 4	Jusqu'à 24 heures

* Non garanti

Avec les certificats X.509 et l'authentification personnalisée, la durée de connexion n'est pas limitée, mais elle peut être de quelques minutes seulement. Les interruptions de la connexion peuvent se produire pour diverses raisons. La liste suivante répertorie certaines des raisons les plus utilisées parmi celles les plus utilisées parmi celles les plus utilisées.

- Interruptions de disponibilité du Wi-Fi
- Interruptions de connexion avec le fournisseur d'accès à Internet (ISP)
- Correctifs de service
- Déploiements de services
- Service Auto Scaling
- Hôte de service non disponible
- Problèmes d'équilibrage de charge et mises à jour
- Erreurs côté du client

Vos appareils doivent mettre en œuvre des stratégies de détection des déconnexions et de reconnexion. Pour plus d'informations sur les événements de déconnexion et des conseils sur la façon de les gérer, voir [??? \(p. 1268\)](#) dans [??? \(p. 1268\)](#).

MQTT

[Le MQTT](#) (Message Queuing Telemetry Transport) est un protocole de messagerie léger et largement adopté, conçu pour les appareils soumis à des contraintes. AWS IoT Core la prise en charge de [MQTT est basée sur les spécifications MQTT v3.1.1 et MQTT v5.0](#), avec quelques différences, comme indiqué dans [the section called "AWS IoT differences par rapport aux spécifications MQTT" \(p. 111\)](#). En tant que dernière version de la norme, MQTT 5 introduit plusieurs fonctionnalités clés qui renforcent la robustesse d'un système basé sur MQTT, notamment de nouvelles améliorations en matière d'évolutivité, des rapports d'erreurs améliorés avec des réponses au code de motivation, des délais d'expiration des messages et des sessions et des en-têtes de messages utilisateur personnalisés. Pour plus d'informations sur les fonctionnalités prises en charge par AWS IoT Core par MQTT 5, consultez la section [Fonctionnalités prises en charge par MQTT 5 \(p. 103\)](#). AWS IoT Core prend également en charge la communication entre versions MQTT (MQTT 3 et MQTT 5). Un éditeur MQTT 3 peut envoyer un message MQTT 3 à un abonné MQTT 5 qui recevra un message de publication MQTT 5, et vice versa.

AWS IoT Core prend en charge les connexions de périphériques qui utilisent le protocole MQTT et le protocole MQTT over WSS et qui sont identifiées par un identifiant client. Ils [Kits SDK pour les appareils AWS IoT \(p. 87\)](#) prennent en charge les deux protocoles et constituent les méthodes recommandées pour connecter des appareils AWS IoT. Les SDK pour AWS IoT appareils prennent en charge les fonctions nécessaires aux appareils et aux clients pour se connecter aux AWS IoT services et y accéder. Les SDK de l'appareil prennent en charge les protocoles d'authentification requis par les AWS IoT services et les exigences en matière d'ID de connexion requises par le protocole MQTT et les protocoles MQTT over WSS. Pour plus d'informations sur la manière de se connecter à l'AWS IoT, consultez [the section called "Connexion à MQTT à l'aide des SDK de l'AWS IoT" \(p. 93\)](#). Pour plus d'informations sur les méthodes d'authentification et les mappages de ports pour les messages MQTT, consultez [Protocoles, mappages de ports et authentification \(p. 89\)](#).

Bien que nous vous recommandons d'utiliser les SDK de l'AWS IoT pour vous connecter à AWS IoT, ils ne sont pas obligatoires. Si vous n'utilisez pas les SDK de l'AWS IoT, vous devez toutefois

fournir la sécurité de connexion et de communication nécessaire. Les clients doivent envoyer l'[extension TLS 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 des utilisateurs et des AWS informations d'identification IAM pour authentifier les clients doivent fournir l'authentification [Signature version 4](#) correcte.

Dans cette rubrique :

- [Connexion à MQTT à l'aide des SDK de l'AWS IoT appareil \(p. 93\)](#)
- [Options de qualité de service \(QoS\) MQTT \(p. 94\)](#)
- [Sessions permanentes MQTT \(p. 94\)](#)
- [MQTT retenus \(p. 97\)](#)
- [Utilisation de ConnectAttributes \(p. 102\)](#)
- [Fonctionnalités compatibles avec MQTT 5 \(p. 103\)](#)
- [Propriétés de MQTT 5 \(p. 107\)](#)
- [Codes de raison MQTT \(p. 108\)](#)
- [AWS IoTdifférences par rapport aux spécifications MQTT \(p. 111\)](#)

Connexion à MQTT à l'aide des SDK de l'AWS IoT appareil

Cette section contient des liens vers les SDK pour AWS IoT appareils et vers le code source d'exemples de programmes qui illustrent la manière de connecter un appareil AWS IoT. Les exemples d'applications liés ici montrent comment se connecter à AWS IoT l'aide du protocole MQTT et de MQTT via WSS.

Note

Les SDK AWS IoT Device ont publié un client MQTT 5 en version préliminaire pour les développeurs. Au cours de la période de prévisualisation, nous pouvons apporter des modifications rétrocompatibles aux API publiques, et les clients de service des SDK AWS IoT Device continuent d'utiliser le client MQTT 3.1.1.

C++

Utilisation du SDK AWS IoT C++ Device pour connecter des appareils

- [Code source d'un exemple d'application présentant un exemple de connexion MQTT en C++](#)
- [AWS IoTSDK v2 du périphérique C++ activé GitHub](#)

Python

Utilisation du SDK AWS IoT Device pour Python pour connecter des appareils

- [Code source d'un exemple d'application présentant un exemple de connexion MQTT en Python](#)
- [AWS IoTSDK de l'appareil pour Python v2 sur GitHub](#)

JavaScript

Utiliser le SDK de l'AWS IoT appareil JavaScript pour connecter des appareils

- [Code source d'un exemple d'application présentant un exemple de connexion MQTT dans JavaScript](#)
- [AWS IoTSDK de l'appareil pour JavaScript v2 sur GitHub](#)

Java

Utilisation du SDK AWS IoT Device pour Java pour connecter des appareils

- [Code source d'un exemple d'application présentant un exemple de connexion MQTT en Java](#)
- [AWS IoTLe SDK de l'appareil pour Java v2 est activé GitHub](#)

Embedded C

Utilisation du AWS IoT SDK pour Embedded C pour connecter des appareils

Important

Ce SDK est destiné à être utilisé par des développeurs de logiciels embarqués expérimentés.

- [Code source d'un exemple d'application présentant un exemple de connexion MQTT dans Embedded C](#)
- [AWS IoTSDK de l'appareil pour C intégré sur GitHub](#)

Options de qualité de service (QoS) MQTT

AWS IoT et les SDK de l'AWS IoTAppareil prennent en charge les [niveaux de qualité de service \(QoS\) MQTT0 et 1](#). Le protocole MQTT définit un troisième niveau de QoS, le niveau2, mais AWS IoT ne le prend pas en charge. Seul le protocole MQTT prend en charge la fonctionnalité QoS. HTTPS prend en charge la QoS en transmettant un paramètre de chaîne de requête?qos=qos dont la valeur peut être 0 ou 1.

Ce tableau décrit la manière dont chaque niveau de QoS affecte les messages publiés vers et par le courtier de messages.

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

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 confirmés par le client. Lorsque l'appareil se reconnecte à une session persistante, la session reprend, les abonnements sont rétablis et les messages d'abonnement sans accusé de réception reçus et stockés avant la reconnexion sont envoyés au client.

Le traitement des messages stockés est enregistré dans CloudWatch et CloudWatch Logs. Pour plus d'informations sur les entrées écrites dans CloudWatch et CloudWatch les journaux, consultez [Métriques d'agent de messages \(p. 481\)](#) et [entrée de entrée de entrée de entrée de entrée de entrée \(p. 495\)](#).

Création d'une session persistante

Dans MQTT 3, vous créez une session MQTT persistante en envoyant unCONNECT message et en définissant l'cleanSessionindicateur sur`0`. Si aucune session n'existe pour le client qui envoie leCONNECT message, une nouvelle session persistante est créée. Si une session existe déjà pour le client, celui-ci reprend la session existante. Pour créer une session propre, vous envoyez unCONNECT message et définissez l'cleanSessionindicateur sur`1`, et le courtier ne stockera aucun état de session lorsque le client se déconnecte.

Dans MQTT 5, vous gérez les sessions persistantes en définissant l'Clean Startindicateur etSession Expiry Interval. Clean Start contrôle le début de la session de connexion et la fin de la session précédente. Lorsque vous définissezClean Start =`1`, une nouvelle session est créée et une session précédente est terminée si elle existe. Lorsque vous définissezClean Start =`0`, la session de connexion reprend une session précédente si elle existe. L'intervalle d'expiration de session contrôle la fin de la session de connexion. L'intervalle d'expiration de session indique la durée, en secondes (entier de 4 octets), pendant laquelle une session persistera après la déconnexion. ParamètreSession Expiry interval =`0` provoque l'arrêt immédiat de la session lors de la déconnexion. Si l'intervalle d'expiration de session n'est pas spécifié dans le message CONNECT, la valeur par défaut est `0`.

Démarrage normal de MQTT 5 et expiration de session

Valeur de la propriété	Description
Clean Start= <code>1</code>	Crée une nouvelle session et met fin à une session précédente s'il en existe une.
Clean Start= <code>0</code>	Reprend une session si une session précédente existe.
Session Expiry Interval> <code>0</code>	Persiste pendant une session.
Session Expiry interval= <code>0</code>	Ne persiste pas une session.

Dans MQTT 5, si vous définissezClean StartSession Expiry Interval =`1` et =`0`, cela équivaut à une session MQTT 3 propre. Si vous définissezClean Start =`0` etSession Expiry Interval >`0`, cela équivaut à une session persistante MQTT 3.

Note

Les sessions persistantes entre versions MQTT (MQTT 3 et MQTT 5) ne sont pas prises en charge. Une session persistante MQTT 3 ne peut pas être reprise en tant que session MQTT 5, et vice versa.

Opérations au cours d'une session persistante

Les clients utilisent l'sessionPresentattribut dans le message de confirmation de connexion (CONNACK) pour déterminer si une session persistante est présente. Si telsessionPresent est le cas`1`, une session persistante est présente et tous les messages stockés pour le client sont remis au client immédiatement après réception du [messageCONNACK, comme décrit dans Trafic de messages après reconexion à une session persistante \(p. 96\)](#). Si telsessionPresent est le cas`1`, le client n'a pas besoin de se réabonner. Toutefois, si telsessionPresent est le cas`0`, aucune session persistante n'est présente et le client doit se réabonner à ses filtres thématiques.

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

Trafic de messages après la reconnexion à une session persistante

Une session persistante représente une connexion permanente entre un client et un courtier de messages MQTT. Lorsqu'un client se connecte au courtier de messages via une session persistante, le courtier de messages enregistre tous les abonnements que le client conclut 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 en fonction de la limite du compte. Les messages qui dépassent cette limite seront rejettés. Pour plus d'informations sur les limites de messages persistants, consultez la section [AWS IoT CorePoints de terminaison et quotas](#). Lorsque le client se reconnecte à sa session persistante, tous les abonnements sont rétablis et tous les messages stockés sont envoyés au client à une fréquence maximale de 10 messages par seconde. Dans MQTT 5, si un QoS1 sortant avec un intervalle d'expiration des messages expire lorsqu'un client est hors ligne, le client ne recevra pas le message expiré une fois la connexion rétablie.

Après la reconnexion, les messages stockés sont envoyés au client, à un débit limité à 10 messages stockés par seconde, ainsi que tout trafic de messages en cours jusqu'à ce que la [Publish requests per second per connection](#) limite soit atteinte. Le taux de remise des messages stockés étant limité, plusieurs secondes seront nécessaires pour délivrer tous les messages stockés si une session comporte plus de 10 messages stockés à remettre après la reconnexion.

Fin d'une session persistante

Les sessions persistantes peuvent se terminer de différentes manières :

- Le délai d'expiration de la session persistante est expiré. Le temporisateur d'expiration de session persistante démarre lorsque le courtier de messages détecte qu'un client s'est déconnecté, soit parce que le client se déconnecte, soit parce que la connexion a expiré.
- Le client envoie un CONNECT message qui définit l'cleanSession indicateur sur 1.

Dans MQTT 3, la valeur par défaut du délai d'expiration des sessions persistantes est d'une heure, et cela s'applique à toutes les sessions du compte.

Dans MQTT 5, vous pouvez définir l'intervalle d'expiration de session pour chaque session sur les paquets CONNECT et DISCONNECT.

Pour l'intervalle d'expiration des sessions sur le paquet DISCONNECT :

- Si la session en cours a un intervalle d'expiration de session de 0, vous ne pouvez pas définir l'intervalle d'expiration de session sur une valeur supérieure à 0 dans le paquet DISCONNECT.
- Si la session en cours a un intervalle d'expiration de session supérieur à 0 et que vous définissez l'intervalle d'expiration de session sur 0 dans le paquet DISCONNECT, la session sera terminée lors de la déconnexion.
- Sinon, l'intervalle d'expiration de session sur le paquet DISCONNECT mettra à jour l'intervalle d'expiration de session de la session en cours.

Note

Les messages stockés qui attendent d'être envoyés au client à la fin d'une session sont supprimés ; toutefois, 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 la section [AWS IoT Core Tarification](#). Vous pouvez configurer l'intervalle de temps d'expiration.

Reconnexion après 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 supprimés. Lorsqu'un client se reconnecte après l'expiration de la session avec

`uncleanSession` indicateur`0`, le service crée une nouvelle session persistante. Les abonnements ou les messages de la session précédente ne sont pas disponibles pour cette session car ils ont été supprimés à l'expiration de la session précédente.

Frais liés aux messages de session persistants

Les messages vous sont facturés lorsque le courtier de messages envoie un message à un client ou lors d'une session persistante hors ligne. Lorsqu'un appareil hors ligne doté d'une session persistante se reconnecte et reprend sa session, les messages enregistrés sont transmis à l'appareil et débités à nouveau sur votre compte. Pour plus d'informations sur la tarification des messages, voir [AWS IoT Core Tarification - Messagerie](#).

Le délai d'expiration d'une session persistante par défaut d'une heure peut être augmenté à l'aide du processus d'augmentation de la limite standard. Notez que l'augmentation du délai d'expiration de la session peut entraîner une augmentation des frais de messagerie, car cette durée supplémentaire pourrait permettre de stocker davantage de messages pour l'appareil hors ligne et ces messages supplémentaires seraient facturés sur votre compte au tarif de messagerie standard. Le délai d'expiration de la session est approximatif et une session peut persister 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 la section [AWS Service Quotas](#).

MQTT retenus

AWS IoT Core comprend en charge l'indicateur RETAIN décrit dans le protocole MQTT. Lorsqu'un client définit l'indicateur RETAIN sur un message MQTT qu'il publie, AWS IoT Core enregistre le message. Il peut ensuite être envoyé aux nouveaux abonnés, récupéré en appelant l'[GetRetainedMessage](#) opération et visualisé dans la [AWS IoT console](#).

Exemples d'utilisation de messages MQTT conservés

- En tant que message de configuration initial

Les messages conservés par MQTT sont envoyés à un client une fois que celui-ci s'est abonné à une rubrique. Si vous souhaitez que tous les clients abonnés à une rubrique reçoivent le message MQTT conservé immédiatement après leur abonnement, vous pouvez publier un message de configuration avec l'indicateur RETAIN activé. Les clients abonnés reçoivent également des mises à jour de cette configuration chaque fois qu'un nouveau message de configuration est publié.

- En tant que dernier message d'état connu

Les appareils peuvent placer l'indicateur RETAIN sur les messages en cours afin de AWS IoT Core les enregistrer. Lorsque les applications se connectent ou se reconnectent, elles peuvent s'abonner à cette rubrique et obtenir le dernier état signalé immédiatement après s'être abonnées à la rubrique de message conservée. De cette façon, ils peuvent éviter d'avoir à attendre le prochain message de l'appareil pour voir l'état actuel.

Dans cette section :

- [Tâches courantes liées à la conservation des messages MQTT dans AWS IoT Core \(p. 97\)](#)
- [Facturation et messages retenus \(p. 100\)](#)
- [Comparaison des messages MQTT conservés et des sessions MQTT persistantes \(p. 100\)](#)
- [MQTT a conservé les messages et les ombres des AWS IoT appareils \(p. 101\)](#)

Tâches courantes liées à la conservation des messages MQTT dans AWS IoT Core

AWS IoT Core enregistre les messages MQTT avec l'indicateur RETAIN défini. Ces messages conservés sont envoyés à tous les clients qui se sont abonnés au sujet, sous la forme d'un message MQTT normal, et ils sont également stockés pour être envoyés aux nouveaux abonnés au sujet.

Les messages conservés par MQTT nécessitent des actions politiques spécifiques pour autoriser les clients à y accéder. Pour des exemples d'utilisation des politiques de conservation des messages, reportez-vous à la section [Exemples de politiques de conservation des messages \(p. 391\)](#).

Cette section décrit les opérations courantes impliquant des messages conservés.

- **Création d'un message retenu**

Le client détermine si un message est conservé lorsqu'il publie un message MQTT. Les clients peuvent définir l'indicateur RETAIN lorsqu'ils publient un message à l'aide d'un [SDK pour appareils \(p. 1494\)](#). Les applications et les services peuvent définir l'indicateur RETAIN lorsqu'ils utilisent l'[Publishaction](#) pour publier un message MQTT.

Un seul message par nom de rubrique est conservé. Un nouveau message avec l'indicateur RETAIN défini publié dans une rubrique remplace tout message conservé qui a été envoyé précédemment à cette rubrique.

REMARQUE : Vous ne pouvez pas publier dans un [sujet réservé lorsque \(p. 117\)](#) l'indicateur RETAIN est activé.

- **Abonnement à un sujet de message conservé**

Les clients s'abonnent aux sujets de message conservés comme ils le feraient pour n'importe quel autre sujet de message MQTT. L'indicateur RETAIN est activé pour les messages conservés reçus en s'abonnant à un sujet de message conservé.

Les messages conservés sont supprimés AWS IoT Core lorsqu'un client publie un message conservé avec une charge utile de 0 octet dans le sujet du message conservé. Les clients qui se sont abonnés au sujet de message conservé recevront également le message à 0 octet.

L'abonnement à un filtre de sujet générique qui inclut un sujet de message conservé permet au client de recevoir les messages suivants publiés dans le sujet du message conservé, mais il ne transmet pas le message conservé lors de l'abonnement.

REMARQUE : Pour recevoir un message conservé lors de l'abonnement, le filtre de sujet de la demande d'abonnement doit correspondre exactement au sujet du message conservé.

L'indicateur RETAIN est activé pour les messages conservés reçus lors de l'abonnement à un sujet de message conservé. Les messages conservés qui sont reçus par un client abonné après son abonnement ne le sont pas.

- **Récupération d'un message conservé**

Les messages conservés sont transmis automatiquement aux clients lorsqu'ils s'abonnent à la rubrique contenant le message conservé. Pour qu'un client puisse recevoir le message conservé lors de son abonnement, il doit s'abonner au nom de sujet exact du message conservé. L'abonnement à un filtre de sujet générique qui inclut un sujet de message conservé permet au client de recevoir les messages suivants publiés dans le sujet du message conservé, mais il ne transmet pas le message conservé lors de l'abonnement.

Les services et les applications peuvent répertorier et récupérer les messages conservés en appelant [ListRetainedMessages](#) et [GetRetainedMessage](#).

Rien n'empêche un client de publier des messages dans une rubrique de message conservée sans définir l'indicateur RETAIN. Cela peut entraîner des résultats inattendus, tels que le message conservé ne correspond pas au message reçu en vous abonnant à la rubrique.

Avec MQTT 5, si l'intervalle d'expiration du message conservé est défini et que le message conservé expire, un nouvel abonné abonné à cette rubrique ne recevra pas le message conservé en cas d'abonnement réussi.

- **Liste des sujets de messages conservés**

Vous pouvez répertorier les messages conservés en appelant [ListRetainedMessages](#) et les messages conservés peuvent être consultés sur la [AWS IoTconsole](#).

- Obtenir les détails des messages conservés

Vous pouvez obtenir les détails des messages conservés en appelant [GetRetainedMessage](#) et ils peuvent être consultés sur la [AWS IoTconsole](#).

- Conservation d'un message testamentaire

Les [messages MQTT Will](#) créés lors de la connexion d'un appareil peuvent être conservés en plaçant l'Will Retain indicateur dans le Connect Flag bits champ.

- Supprimer un message retenu

Les appareils, les applications et les services peuvent supprimer un message conservé en publiant un message avec l'indicateur RETAIN activé et une charge de message vide (0 octet) sous le nom de rubrique du message conservé à supprimer. Ces messages suppriment le message conservé AWS IoT Core, sont envoyés aux clients abonnés à la rubrique, mais ils ne sont pas conservés par AWS IoT Core.

Les messages conservés peuvent également être supprimés de manière interactive en accédant au message conservé dans la [AWS IoTconsole](#). Les messages conservés qui sont supprimés à l'aide de la [AWS IoTconsole](#) envoient également un message de 0 octet aux clients abonnés à la rubrique du message conservé.

Les messages conservés ne peuvent pas être restaurés après leur suppression. Un client devrait publier un nouveau message conservé pour remplacer le message supprimé.

- Débogage et résolution des problèmes liés aux messages conservés

La [AWS IoTconsole](#) propose plusieurs outils pour vous aider à résoudre les problèmes liés aux messages conservés :

- La page [Messages conservés](#)

La page Messages conservés de la AWS IoT console fournit une liste paginée des messages conservés qui ont été stockés par votre compte dans la région actuelle. Depuis cette page, vous pouvez :

- Consultez les détails de chaque message conservé, tels que la charge utile du message, la QoS l'heure à laquelle il a été reçu.
- Mettez à jour le contenu d'un message conservé.
- Supprimer un message conservé.

- Le [client de test MQTT](#)

La page du client de test MQTT de la AWS IoT console permet de s'abonner à des rubriques MQTT et de les publier. L'option de publication vous permet de définir l'indicateur RETAIN sur les messages que vous publiez afin de simuler le comportement de vos appareils.

Certains résultats inattendus peuvent être le résultat de ces aspects de la manière dont les messages conservés sont implémentés dans AWS IoT Core.

- Limites de messages conservés

Lorsqu'un compte a enregistré le nombre maximum de messages conservés, il AWS IoT Core renvoie une réponse limitée aux messages publiés avec RETAIN défini et des charges utiles supérieures à 0 octet jusqu'à ce que certains messages conservés soient supprimés et que le nombre de messages conservés soit inférieur à la limite.

- Ordre de remise des messages retenu

La séquence entre le message conservé et la livraison du message abonné n'est pas garantie.

Facturation et messages retenus

La publication de messages avec l'indicateur RETAIN activé depuis un client, en utilisant AWS IoT la console ou en appelant [Publish](#) entraîne des frais de messagerie supplémentaires décrits dans la section [AWS IoT Core Tarification - Messagerie](#).

La récupération des messages conservés par un client, en utilisant AWS IoT la console ou en appelant [GetRetainedMessage](#) entraîne des frais de messagerie en plus des frais d'utilisation normaux de l'API. Les frais supplémentaires sont décrits dans la section [AWS IoT Core Tarification - Messagerie](#).

MQTT [Les messages](#) publiés lorsqu'un appareil se déconnecte de manière inattendue entraîneront-ils des frais de messagerie décrits dans la section [AWS IoT Core Tarification - Messagerie](#).

Pour plus d'informations sur les coûts de messagerie, consultez la section [AWS IoT Core Tarification - Messagerie](#).

Comparaison des messages MQTT conservés et des sessions MQTT persistantes

Les messages conservés et les sessions persistantes sont des fonctionnalités standard de MQTT qui permettent aux appareils de recevoir des messages publiés alors qu'ils étaient hors ligne. Les messages conservés peuvent être publiés à partir de sessions persistantes. Cette section décrit les principaux aspects de ces fonctionnalités et la manière dont elles fonctionnent ensemble.

	Messages conservés	Sessions persistantes
Fonctions principales	<p>Les messages conservés peuvent être utilisés pour configurer ou notifier de grands groupes d'appareils après leur connexion.</p> <p>Les messages conservés peuvent également être utilisés lorsque vous souhaitez que les appareils reçoivent uniquement le dernier message publié dans une rubrique après une reconnexion.</p>	<p>Les sessions persistantes sont utiles pour les appareils dont la connectivité est intermittente et qui risquent de manquer plusieurs messages importants.</p> <p>Les appareils peuvent se connecter via une session persistante pour recevoir les messages envoyés lorsqu'ils sont hors ligne.</p>
Exemples	<p>Les messages conservés peuvent fournir aux appareils des informations de configuration relatives à leur environnement lorsqu'ils sont connectés. La configuration initiale peut inclure une liste d'autres sujets de message auxquels il doit s'abonner ou des informations sur la manière dont il doit configurer son fuseau horaire local.</p>	<p>Les appareils qui se connectent via un réseau cellulaire à connectivité intermittente peuvent utiliser des sessions persistantes pour éviter de manquer des messages importants qui sont envoyés alors qu'un appareil n'est pas couvert par le réseau ou doit éteindre sa radio cellulaire.</p>
Messages reçus lors de l'inscription initiale à une rubrique	Après l'abonnement à une rubrique contenant un message conservé, le message conservé le plus récent est reçu.	Après un abonnement à un sujet sans message conservé, aucun message n'est reçu tant qu'un message n'est pas publié dans le sujet.

	Messages conservés	Sessions persistantes
Sujets auxquels vous êtes abonné après la reconnexion	En l'absence de session persistante, le client doit s'abonner aux sujets après s'être reconnecté.	Les sujets auxquels vous êtes abonné sont restaurés après la reconnexion.
Messages reçus après la reconnexion	Après l'abonnement à une rubrique contenant un message conservé, le message conservé le plus récent est reçu.	Tous les messages publiés avec une QOS = 1 et auxquels vous êtes abonné avec une QOS = 1 alors que l'appareil était déconnecté sont envoyés après la reconnexion de l'appareil.
Expiration des données/sessions	Dans MQTT 3, les messages conservés n'expirent pas. Ils sont conservés jusqu'à ce qu'ils soient remplacés ou supprimés. Dans MQTT 5, les messages conservés expirent après l'intervalle d'expiration que vous avez défini. Pour de plus amples informations, veuillez consulter Expiration des messages (p. 103) .	Les sessions persistantes expirent si le client ne se reconnecte pas dans le délai imparti. À l'expiration d'une session persistante, les abonnements et les messages enregistrés du client qui ont été publiés avec une QOS = 1 et auxquels le client s'est abonné avec une QOS = 1 alors que l'appareil était déconnecté sont supprimés. Les messages expirés ne seront pas remis. Pour de plus amples informations sur les expirations de session dans le cadre de sessions persistantes, veuillez consulter the section called "Sessions permanentes MQTT" (p. 94) .

Pour plus d'informations sur les sessions persistantes, consultez [the section called "Sessions permanentes MQTT" \(p. 94\)](#).

Avec les messages conservés, le client de publication détermine si un message doit être conservé et remis à un appareil après sa connexion, qu'il ait déjà eu une session ou non. Le choix de stocker un message est fait par l'éditeur et le message stocké est remis à tous les clients actuels et futurs qui s'abonnent avec un abonnement QoS 0 ou QoS 1. Les messages conservés ne contiennent qu'un seul message à la fois sur un sujet donné.

Lorsqu'un compte a enregistré le nombre maximum de messages conservés, AWS IoT Core renvoie une réponse limitée aux messages publiés avec RETAIN défini et des charges utiles supérieures à 0 octet jusqu'à ce que certains messages conservés soient supprimés et que le nombre de messages conservés soit inférieur à la limite.

[MQTT a conservé les messages et les ombres des AWS IoT appareils](#)

Les messages conservés et les Device Shadows conservent les données d'un appareil, mais ils se comportent différemment et ont des objectifs différents. La présente section explique leurs similitudes et leurs différences.

	Messages conservés	Device Shadows
La charge utile du message possède une structure ou un schéma prédéfini	Tel que défini par la mise en œuvre. MQTT ne spécifie pas de structure ou de schéma pour la charge utile de ses messages.	AWS IoT prend en charge une structure de données spécifique.
La mise à jour de la charge utile des messages génère des messages d'événement	La publication d'un message conservé envoie le message aux clients abonnés, mais ne génère pas de messages de mise à jour supplémentaires.	La mise à jour d'un Device Shadow génère des messages de mise à jour qui décrivent la modification .
Les mises à jour des messages sont numérotées	Les messages conservés ne sont pas numérotés automatiquement.	Les documents Device Shadow possèdent des numéros de version et des horodatages automatiques.
La charge utile du message est attachée à une ressource d'objet	Les messages conservés ne sont pas attachés à une ressource d'objet.	Les ombres de l'appareil sont attachées à une ressource d'objet.
Mettre à jour des éléments individuels de la charge utile du message	Les éléments individuels du message ne peuvent pas être modifiés sans mettre à jour l'intégralité de la charge utile du message.	Les éléments individuels d'un document Device Shadow peuvent être mis à jour sans qu'il soit nécessaire de mettre à jour l'intégralité du document Device Shadow.
Le client reçoit les données du message lors de son abonnement	Le client reçoit automatiquement un message conservé après s'être abonné à une rubrique contenant un message conservé.	Les clients peuvent s'abonner aux mises à jour de Device Shadow, mais ils doivent demander délibérément l'état actuel.
Indexation et recherche	Les messages conservés ne sont pas indexés à des fins de recherche.	L'indexation de la flotte indexe les données Device Shadow à des fins de recherche et d'agrégation.

Utilisation de ConnectAttributes

`ConnectAttributes` vous permettent de spécifier les attributs que vous souhaitez utiliser dans votre message de connexion dans vos politiques IAM, tels que `PersistentConnect` et `LastWill`. Avec `ConnectAttributes`, vous pouvez créer des politiques qui ne permettent pas aux appareils d'accéder aux nouvelles fonctionnalités par défaut, ce qui peut être utile en cas de compromission d'un appareil.

`connectAttributes` prend en charge les fonctions suivantes :

`PersistentConnect`

Utilisez `PersistentConnect` cette fonctionnalité pour enregistrer tous les abonnements effectués par le client pendant la connexion lorsque la connexion entre le client et le courtier est interrompue.

`LastWill`

Utilisez `LastWill` cette fonctionnalité pour publier un message `LastWillTopic` lorsqu'un client se déconnecte de manière inattendue.

Par défaut, votre politique prévoit une connexion non persistante et aucun attribut n'est transmis pour cette connexion. Vous devez spécifier une connexion permanente dans votre politique IAM si vous souhaitez en avoir une.

Pour des `ConnectAttributes` exemples, consultez la section [Exemples de politiques de Connect \(p. 369\)](#).

Fonctionnalités compatibles avec MQTT 5

AWS IoT Core la prise en charge de MQTT 5 est basée sur la [spécification MQTT v5.0](#) avec quelques différences, comme indiqué dans [the section called "AWS IoT differences par rapport aux spécifications MQTT" \(p. 111\)](#).

AWS IoT Core prend en charge les fonctionnalités MQTT 5 suivantes :

- [Abonnements partagés \(p. 103\)](#)
- [Démarrage normal et expiration de session \(p. 105\)](#)
- [Code de raison sur tous les ACK \(p. 105\)](#)
- [Alias de rubrique \(p. 105\)](#)
- [Expiration du message \(p. 105\)](#)
- [Autres fonctionnalités de MQTT 5 \(p. 106\)](#)

Abonnements partagés

AWS IoT Core prend en charge les abonnements partagés pour MQTT 3 et MQTT 5. Les abonnements partagés permettent à plusieurs clients de partager un abonnement à un sujet et un seul client recevra les messages publiés sur ce sujet selon une distribution aléatoire. Les abonnements partagés peuvent équilibrer efficacement la charge des messages MQTT entre plusieurs abonnés. Supposons, par exemple, que 1 000 appareils publient sur le même sujet et que 10 applications dorsales traitent ces messages. Dans ce cas, les applications dorsales peuvent s'abonner au même sujet et chacune recevra de manière aléatoire des messages publiés par les appareils sur le sujet partagé. Cela revient à « partager » efficacement la charge de ces messages. Les abonnements partagés permettent également une meilleure résilience. Lorsqu'une application principale se déconnecte, le courtier répartit la charge entre les abonnés restants du groupe.

Pour utiliser les abonnements partagés, les clients s'abonnent au [filtre thématique](#) d'un abonnement partagé comme suit :

```
$share/{ShareName}/{TopicFilter}
```

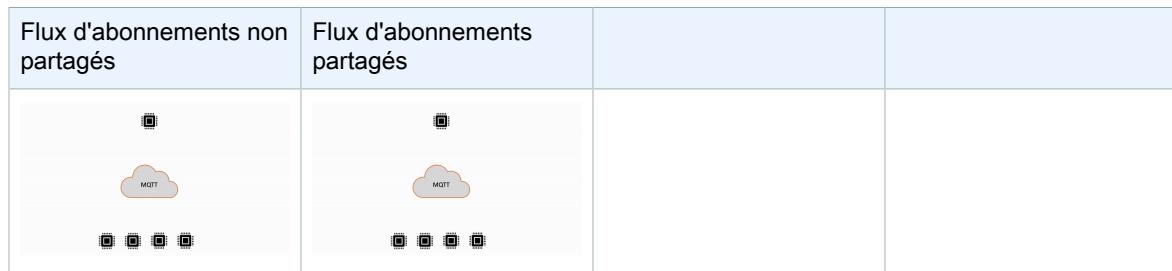
- \$share est une chaîne littérale qui indique le filtre thématique d'un abonnement partagé, qui doit commencer par \$share.
- {ShareName} est une chaîne de caractères permettant de spécifier le nom partagé utilisé par un groupe d'abonnés. Le filtre de sujet d'un abonnement partagé doit contenir un ShareName et être suivi du / caractère. Ils ne {ShareName} doivent pas inclure les caractères suivants : /, +, ou #. La taille maximale pour {ShareName} est de 128 octets.
- {TopicFilter} suit la même syntaxe de [filtrage de sujets](#) qu'un abonnement non partagé et ne peut pas être utilisé pour les [sujets réservés](#). La taille maximale pour {TopicFilter} est de 256 octets.
- Les deux barres obliques (/) requises pour ne \$share/{ShareName}/{TopicFilter} sont pas incluses dans le [nombre maximal de barres obliques dans le sujet et la limite de filtre par sujet](#).

Les abonnements qui ont le même {ShareName}/{TopicFilter} nom appartiennent au même groupe d'abonnements partagés. Vous pouvez créer plusieurs groupes d'abonnements partagés sans dépasser la

[limite d'abonnements partagés par groupe](#). Pour plus d'informations, consultez les [AWS IoT Corepoints de terminaison et les quotas](#) de la RéférenceAWS générale.

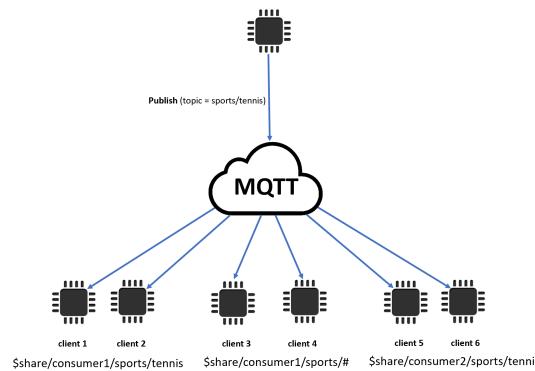
Les tableaux suivants comparent les abonnements non partagés et les abonnements partagés :

Abonnement	Description	Exemples de filtres de rubrique
Abonnements non partagés	Chaque client crée un abonnement distinct pour recevoir les messages publiés. Lorsqu'un message est publié dans une rubrique, tous les abonnés à cette rubrique reçoivent une copie du message.	sports/tennis sports/#
Abonnements partagés	Plusieurs clients peuvent partager un abonnement à un sujet et un seul client recevra les messages publiés sur ce sujet de manière aléatoire.	\$share/consumer/ sports/tennis \$share/consumer/ sports/#



Remarques importantes concernant l'utilisation des abonnements partagés

- Lorsqu'une tentative de publication auprès d'un abonné QoS0 échoue, aucune nouvelle tentative ne se produit et le message est supprimé.
- Lorsqu'une tentative de publication auprès d'un abonné QoS1 avec une session vide échoue, le message est envoyé à un autre abonné du groupe pour plusieurs tentatives. Les messages qui ne parviennent pas à être remis après toutes les tentatives de nouvelle tentative seront supprimés.
- Lorsqu'une tentative de publication destinée à un abonné QoS1 avec [des sessions persistantes \(p. 94\)](#) échoue parce que l'abonné est hors ligne, les messages ne sont pas mis en file d'attente et sont envoyés à un autre abonné du groupe.
- Les abonnements partagés ne reçoivent pas de [messages conservés](#).
- Lorsque les abonnements partagés contiennent des caractères génériques (# ou +), plusieurs abonnements partagés peuvent correspondre à une rubrique. Dans ce cas, le courtier de messages copie le message de publication et l'envoie à un client aléatoire dans chaque abonnement partagé correspondant. Le comportement générique des abonnements partagés peut être expliqué dans le schéma suivant.



Dans cet exemple, trois abonnements partagés correspondent à la rubrique MQTT de publicationsports/tennis. Le courtier de messages copie le message publié et l'envoie à un client aléatoire dans chaque groupe correspondant.

Le client 1 et le client 2 partagent l'abonnement :\$share/consumer1/sports/tennis

Le client 3 et le client 4 partagent l'abonnement :\$share/consumer1/sports/#

Le client 5 et le client 6 partagent l'abonnement :\$share/consumer2/sports/tennis

Pour plus d'informations sur les limites des abonnements partagés, consultez la section [AWS IoT CorePoints de terminaison et quotas](#) de la RéférenceAWS générale. Pour tester les abonnements partagés à l'aide du client AWS IoT MQTT dans la [AWS IoTconsole](#), consultez [???](#) (p. 74). Pour plus d'informations sur les abonnements partagés, consultez la section [Abonnements partagés](#) de la spécification MQTtv5.0.

Démarrage normal et expiration de session

Vous pouvez utiliser Clean Start et Session Expiry pour gérer vos sessions persistantes avec plus de flexibilité. Un indicateur Clean Start indique si la session doit démarrer sans utiliser une session existante. Un intervalle d'expiration de session indique la durée pendant laquelle la session doit être conservée après une déconnexion. L'intervalle d'expiration de la session peut être modifié lors de la déconnexion. Pour plus d'informations, veuillez consulter [the section called “Sessions permanentes MQTT” \(p. 94\)](#).

Code de raison sur tous les ACK

Vous pouvez déboguer ou traiter les messages d'erreur plus facilement à l'aide des codes de raison. Les codes de motif sont renvoyés par le courtier de messages en fonction du type d'interaction avec le courtier (s'abonner, publier, accuser réception). Pour plus d'informations, consultez les [codes de raison MQTT \(p. 108\)](#). Pour une liste complète des codes de justification MQTT, consultez la [spécification MQTT v5](#).

Alias de rubrique

Vous pouvez remplacer le nom d'une rubrique par un alias de rubrique, qui est un entier de deux octets. L'utilisation d'alias de rubrique permet d'optimiser la transmission des noms de rubriques afin de réduire potentiellement les coûts de données sur les services de données limités. AWS IoT Core a une limite par défaut de 8 alias de rubrique. Pour plus d'informations, consultez les [AWS IoT Corepoints de terminaison et les quotas](#) de la RéférenceAWS générale.

Expiration du message

Vous pouvez ajouter des valeurs d'expiration aux messages publiés. Ces valeurs représentent l'intervalle d'expiration des messages en secondes. Si un message n'a pas été envoyé aux abonnés dans cet

intervalle, il expire et est supprimé. Si vous ne définissez pas la valeur d'expiration du message, celui-ci n'expirera pas.

À la sortie, l'abonné recevra un message indiquant le temps restant avant la fin de l'intervalle d'expiration. Par exemple, si un message de publication entrant a une expiration de 30 secondes et qu'il est acheminé vers l'abonné au bout de 20 secondes, le champ d'expiration du message sera mis à jour à 10. Il est possible que le message reçu par l'abonné ait une MEI mise à jour de 0. En effet, dès que le temps restant est inférieur ou égal à 999 ms, il sera mis à jour à 0.

Dans AWS IoT Core, l'intervalle d'expiration minimum des messages est de 1. Si l'intervalle est défini sur 0 côté client, il sera ajusté à 1. L'intervalle d'expiration maximum des messages est de 604 800 (7 jours). Toute valeur supérieure à cette valeur sera ajustée à la valeur maximale.

Dans la communication entre versions, le comportement d'expiration du message est déterminé par la version MQTT du message de publication entrant. Par exemple, un message avec expiration envoyé par une session connectée via MQTT5 peut expirer pour les appareils abonnés à des sessions MQTT3. Le tableau ci-dessous indique comment l'expiration des messages prend en charge les types de messages de publication suivants :

Type de message de publication	Intervalle d'expiration des messages
Publier régulièrement	Si un serveur ne parvient pas à transmettre le message dans le délai spécifié, le message expiré sera supprimé et l'abonné ne le recevra pas. Cela inclut des situations telles que lorsqu'un appareil ne publie pas ses messages QoS 1.
Retain	Si un message conservé expire et qu'un nouveau client s'abonne à la rubrique, le client ne recevra pas le message lors de son abonnement.
Dernier testament	L'intervalle entre les messages de dernière volonté commence une fois que le client se déconnecte et que le serveur tente de transmettre le message de dernier testament à ses abonnés.
Messages mis en file d'attente	Si un QoS1 sortant avec intervalle d'expiration des messages expire alors qu'un client est hors ligne, après la reprise de la session persistante (p. 94) , le client ne recevra pas le message expiré.

Autres fonctionnalités de MQTT 5

Déconnexion du serveur

Lorsqu'une déconnexion se produit, le serveur peut envoyer de manière proactive au client un message DISCONNECT pour notifier la fermeture de la connexion avec un code de motif de déconnexion.

Demande/Réponse

Les éditeurs peuvent demander au destinataire d'envoyer une réponse à un sujet spécifié par l'éditeur dès réception.

Taille de paquet maximale

Le client et le serveur peuvent spécifier indépendamment la taille de paquet maximale qu'ils prennent en charge.

Format de charge utile et type de contenu

Vous pouvez spécifier le format de la charge utile (binaire, texte) et le type de contenu lorsqu'un message est publié. Ils sont transmis au destinataire du message.

Propriétés de MQTT 5

Les propriétés MQTT 5 sont des ajouts importants à la norme MQTT pour prendre en charge les nouvelles fonctionnalités de MQTT 5 telles que l'expiration des sessions et le modèle de demande/réponse.

Dans AWS IoT Core, vous pouvez créer des [règles](#) qui peuvent transférer les propriétés des messages sortants, ou utiliser [HTTP Publish](#) pour publier des messages MQTT avec certaines des nouvelles propriétés.

Le tableau suivant répertorie toutes les propriétés MQTT 5 prises en AWS IoT Core charge.

Propriété	Description	Type d'entrée	Paquet
Indicateur de format de charge utile	Valeur booléenne qui indique si la charge utile est formatée en UTF-8.	Octet	PUBLIER, CONNECTER
Type de contenu	Chaîne UTF-8 qui décrit le contenu de la charge utile.	Chaîne UTF-8	PUBLIER, CONNECTER
Sujet de réponse	Chaîne UTF-8 qui décrit la rubrique dans laquelle le récepteur doit effectuer la publication dans le cadre du flux demande-réponse. Le sujet ne doit pas contenir de caractères génériques.	Chaîne UTF-8	PUBLIER, CONNECTER
Données de corrélation	Données binaires utilisées par l'expéditeur du message de demande pour identifier la demande à laquelle correspond le message de réponse.	Binaire	PUBLIER, CONNECTER
Propriété de l'utilisateur	Une paire de chaînes UTF-8. Cette propriété peut apparaître plusieurs fois dans un même paquet. Les récepteurs recevront les paires clé-valeur dans l'ordre dans lequel elles ont été envoyées.	Paire de chaînes UTF-8	CONNECTEZ-VOUS, PUBLIEZ, Will Properties, ABONNEZ-VOUS, DÉCONNECTEZ-VOUS, DÉSABONNEZ-VOUS
Intervalle d'expiration des messages	Nombre entier de 4 octets qui représente l'intervalle d'expiration du message en secondes. S'il est absent, le message n'expire jamais.	Nombre entier de 4 octets	PUBLIER, CONNECTER
Intervalle d'expiration des sessions	Entier de 4 octets qui représente l'intervalle d'expiration de la session en secondes. AWS IoT Core prend en charge un maximum de 7 jours, avec un maximum par défaut d'une heure. Si la valeur que vous définissez dépasse le maximum de votre compte, la valeur ajustée AWS IoT Core sera renvoyée dans le CONNACK.	Nombre entier de 4 octets	CONNECTER, CONNECTER, DÉCONNECTER
Identifiant client attribué	Un identifiant client aléatoire généré par AWS IoT Core lorsqu'aucun identifiant client n'est spécifié par les appareils. L'identifiant client aléatoire doit être un nouvel identifiant client qui n'est utilisé par aucune autre session actuellement gérée par le courtier.	Chaîne UTF-8	CONNACK
Serveur Keep Alive	Entier de 2 octets qui représente la durée de maintien en vie attribuée par le serveur. Le serveur déconnectera le client si celui-ci est	Nombre entier à 2 octets	CONNACK

Propriété	Description	Type d'entrée	Paquet
	inactif pendant une durée supérieure à la durée de maintien en vie.		
Demander des informations sur le problème	Valeur booléenne qui indique si la chaîne Reason ou les propriétés utilisateur sont envoyées en cas d'échec.	Octet	CONNECT
Recevoir un maximum	Un entier de 2 octets qui représente le nombre maximum de paquets PUBLISH QOS > 0 pouvant être envoyés sans recevoir de PUBACK.	Nombre entier à 2 octets	CONNECTER, CONNECTER
Nombre maximal d'alias de sujet	Cette valeur indique la valeur la plus élevée qui sera acceptée comme alias de rubrique. La valeur par défaut est 0.	Nombre entier à 2 octets	CONNECTER, CONNECTER
QoS maximum	La valeur maximale de la QoS prise AWS IoT Core en charge. La valeur par défaut est 1. AWS IoT Core ne prend pas en charge QoS2.	Octet	CONNACK
Rester disponible	Valeur booléenne qui indique si le courtier AWS IoT Core messages prend en charge les messages conservés. La valeur par défaut est 1.	Octet	CONNACK
Taille maximale de paquet maximum	Taille maximale des paquets AWS IoT Core acceptés et envoyés. Ne peut pas dépasser 128 Ko.	Nombre entier de 4 octets	CONNECTER, CONNECTER
Abonnement Wildcard disponible	Valeur booléenne qui indique si le courtier AWS IoT Core messages prend en charge l'abonnement Wildcard disponible. La valeur par défaut est 1.	Octet	CONNACK
Identifiant d'abonnement disponible	Valeur booléenne qui indique si le courtier AWS IoT Core messages prend en charge l'identifiant d'abonnement disponible. La valeur par défaut est 0.	Octet	CONNACK

Codes de raison MQTT

MQTT 5 introduit des rapports d'erreurs améliorés avec des réponses par code de raison. AWS IoT Core peuvent renvoyer des codes de justification, y compris, mais sans s'y limiter, les suivants, regroupés par paquets. Pour obtenir la liste complète des codes Reason pris en charge par MQTT 5, consultez les [spécifications de MQTT 5](#).

Codes de raison CONNACK

Valeur	Hex	Raison du code de raison	Description
0	0x00	Réussite	La connexion est acceptée.

Valeur	Hex	Raison du code de raison	Description
128	0x80	Erreur non spécifiée	Le serveur ne souhaite pas révéler la raison de la panne, ou aucun autre code de raison ne s'applique.
133	0x85	Identifiant du client non valide	L'identifiant du client est une chaîne valide mais n'est pas autorisée par le serveur.
134	0x86	Nom d'utilisateur ou mot de passe incorrect	Le serveur n'accepte pas le nom d'utilisateur ou le mot de passe spécifiés par le client.
135	0x87	Non autorisé	Le client n'est pas autorisé à se connecter.
144	0x90	Nom du sujet non valide	Le nom du sujet du testament est correctement formé mais n'est pas accepté par le serveur.
151	0x97	Quota dépassé	Une limite de mise en œuvre ou imposée par l'administration a été dépassée.
155	0x9B	QoS non pris en charge	Le serveur ne prend pas en charge la QoS définie dans Will QoS.

Codes de raison PUBACK

Valeur	Hex	Raison du code de raison	Description
0	0x00	Réussite	Le message est accepté. La publication du message QoS 1 se poursuit.
128	0x80	Erreur non spécifiée	Le destinataire n'accepte pas la publication, mais ne souhaite pas en révéler la raison ou elle ne correspond pas à l'une des autres valeurs.
135	0x87	Non autorisé	Le PUBLISH n'est pas autorisé.
144	0x90	Nom du sujet non valide	Le nom de la rubrique n'est pas mal formé, mais n'est pas accepté par le client ou le serveur.
145	0x91	Identifiant de paquet en cours d'utilisation	L'identifiant du paquet est déjà utilisé. Cela peut indiquer une incompatibilité dans l'état de la session entre le client et le serveur.
151	0x97	Quota dépassé	Une limite de mise en œuvre ou imposée par l'administration a été dépassée.

Codes de motif de déconnexion

Valeur	Hex	Raison du code de raison	Description
129	0x81	Paquet invalide	Le paquet reçu n'est pas conforme à cette spécification.
130	0x82	Erreur de protocole	Un paquet inattendu ou hors service a été reçu.

Valeur	Hex	Raison du code de raison	Description
135	0x87	Non autorisé	La demande n'est pas autorisée.
139	0x8B	Arrêt du serveur	Le serveur est en train de s'arrêter.
141	0x8D	Délai d'expiration de Keep Alive	La connexion est fermée car aucun paquet n'a été reçu pendant 1,5 fois la durée de Keep Alive.
142	0x8E	Session reprise	Une autre connexion utilisant le même ID client s'est connectée, ce qui a entraîné la fermeture de cette connexion.
143	0x8F	Filtre de rubrique invalide	Le filtre de rubrique est correctement formé mais n'est pas accepté par le serveur.
144	0x90	Nom du sujet non valide	Le nom de rubrique est correctement formé mais n'est pas accepté par ce client ou ce serveur.
147	0x93	Dépassement du maximum de réception	Le client ou le serveur a reçu plus que la publication Receive Maximum pour laquelle il n'a envoyé ni PUBACK ni PUBCOMP.
148	0x94	Alias de rubrique invalide in	Le client ou le serveur a reçu un paquet PUBLISH contenant un alias de rubrique supérieur à l'alias de rubrique maximum envoyé dans le paquet CONNECT ou CONNACK.
151	0x97	Quota dépassé	Une limite de mise en œuvre ou imposée par l'administration a été dépassée.
152	0x98	Action administrative	La connexion est fermée en raison d'une action administrative.
155	0x9B	QoS non pris en charge	Le client a spécifié une QoS supérieure à la QoS spécifiée dans une QoS maximale dans le CONNACK.
161	0xA1	Identifiants d'abonnement non pris en charge	Le serveur ne prend pas en charge les identifiants d'abonnement ; l'abonnement n'est pas accepté.

Codes de raison SUBACK

Valeur	Hex	Raison du code de raison	Description
0	0x00	QoS accordé 0	L'abonnement est accepté et la QoS maximale envoyée sera de 0. Il s'agit peut-être d'une QoS inférieure à celle demandée.
1	0x01	QoS 1 accordé	L'abonnement est accepté et la QoS maximale envoyée sera de 1. Il s'agit peut-être d'une QoS inférieure à celle demandée.
128	0x80	Erreur non spécifiée	L'abonnement n'est pas accepté et soit le serveur ne souhaite pas en révéler la raison, soit aucun autre code de raison ne s'applique.
135	0x87	Non autorisé	Le Client n'est pas autorisé à effectuer cette souscription.

Valeur	Hex	Raison du code de raison	Description
143	0x8F	Filtre de rubrique invalide invalide	Le filtre thématique est correctement formé mais n'est pas autorisé pour ce client.
145	0x91	Identifiant de paquet utilisé	L'identificateur de paquet spécifié est déjà utilisé.
151	0x97	Quota dépassé	Une limite de mise en œuvre ou imposée par l'administration a été dépassée.

Codes de raison UNSUBACK

Valeur	Hex	Raison du code de raison	Description
0	0x00	Réussite	L'abonnement est supprimé.
128	0x80	Erreur non spécifiée	La désinscription n'a pas pu être terminée et le serveur ne souhaite pas en révéler la raison ou aucun autre code de raison ne s'applique.
143	0x8F	Filtre de rubrique invalide invalide	Le filtre thématique est correctement formé mais n'est pas autorisé pour ce client.
145	0x91	Identifiant de paquet utilisé	L'identificateur de paquet spécifié est déjà utilisé.

AWS IoT différences par rapport aux spécifications MQTT

L'implémentation du courtier de messages est basée sur les [spécifications MQTT v3.1.1](#) et [MQTT v5.0](#), mais elle diffère des spécifications de la manière suivante :

- AWS IoT ne prend pas en charge les paquets suivants pour MQTT 3 : PUBREC, PUBREL et PUBCOMP.
- AWS IoT ne prend pas en charge les paquets suivants pour MQTT 5 : PUBREC, PUBREL, PUBCOMP et AUTH.
- AWS IoT ne prend pas en charge la redirection de serveur MQTT 5.
- AWS IoT prend en charge les niveaux de qualité de service (QoS) MQTT 0 et 1 uniquement. AWS IoT ne prend pas en charge la publication ou l'abonnement avec un niveau de QoS 2. Lorsque le niveau de QoS 2 est demandé, le courtier 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 par le courtier de AWS IoT 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.
- Lorsqu'un client utilise le caractère générique# dans le filtre de rubrique pour s'abonner à une rubrique, toutes les chaînes situées à son niveau et en dessous dans la hiérarchie des rubriques correspondent. Cependant, le sujet parent ne correspond pas. Par exemple, un abonnement à la rubrique sensor/

reçoit les messages publiés dans les rubriquessensor/,sensor/temperaturesensor/temperature/room1, mais pas les messages publiés sur les rubriquessensor. Pour plus d'informations sur les caractères génériques, consultez[Filtres de rubrique \(p. 116\)](#).

- 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 ayant le même identifiant client ne peuvent pas être connectés simultanément au courtier 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'abonnement à des filtres thématiques contenant un caractère générique ne permet pas de recevoir les messages conservés. Pour recevoir un message conservé, la demande d'abonnement doit contenir un filtre de sujet qui correspond exactement au sujet du message conservé.
- Le courtier de messages ne garantit pas l'ordre dans lequel les messages et l'ACK sont reçus.
- AWS IoT peuvent avoir des limites différentes des spécifications. Pour plus d'informations, consultez les [limites et quotas du courtier de AWS IoT messages et des protocoles](#) dans le Guide de AWS IoT référence.

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 \(p. 89\)](#).

Note

HTTPS ne prend pas en charge une `clientId` valeur comme le fait MQTT. `clientId` est disponible lorsque vous utilisez MQTT, mais il n'est pas disponible lorsque vous utilisez HTTPS.

URL du message HTTPS

Les appareils et les clients publient leurs messages en envoyant des requêtes POST à un point de terminaison spécifique au client et à une URL spécifique au sujet :

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

- *IOT_Data_Endpoint* est le point de terminaison des données de l'appareil. [AWS IoT \(p. 85\)](#) Vous pouvez trouver le point de terminaison dans la AWS IoT console, sur la page de détails de l'objet ou sur le client à l'aide de la AWS CLI commande suivante :

```
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 \(p. 115\)](#) complet du message envoyé.

Exemples de codes de message HTTPS

Voici quelques exemples de la façon d'envoyer un message HTTPS à AWS IoT.

Python (port 8443)

```
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)

```

Python (port 443)

```

import requests
import http.client
import json
import ssl

ssl_context = ssl.SSLContext(protocol=ssl.PROTOCOL_TLS_CLIENT)
ssl_context.minimum_version = ssl.TLSVersion.TLSv1_2

# note the use of ALPN
ssl_context.set_alpn_protocols(["x-amzn-http-ca"])
ssl_context.load_verify_locations(cafile="./<root_certificate>")

# update the certificate and the AWS endpoint
ssl_context.load_cert_chain("./<certificate_in_PEM_Format>",
                            "<private_key_in_PEM_format>")
connection = http.client.HTTPSConnection('<the ats IoT endpoint>', 443,
                                         context=ssl_context)
message = {'data': 'Hello, I\'m using TLS Client authentication!'}
json_data = json.dumps(message)
connection.request('POST', '/topics/device%2Fmessage?qos=1', json_data)

# make request
response = connection.getresponse()

# print results
print(response.read().decode())

```

CURL

Vous pouvez utiliser [curl](#) depuis un client ou 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 la curl version.

- 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 `--tlsv1.2` cette option ne s'affiche pas ou si vous recevez un command not found message d'erreur, vous devrez peut-être mettre à jour ou installer curl sur votre client ou procéder à l'installation 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).
- Le fichier de certificat du client ([device.pem.crt](#) dans cet exemple).
- Le fichier de clé privée du client ([private.pem.key](#) dans cet exemple).

3. Créez la ligne de curl commande en remplaçant les valeurs remplaçables par 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 [device.pem.crt](#)

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

--clé privée.[private.pem.key](#)

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

--requête POST

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

--data "{\"message\" : \"Bonjour tout le 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/topics/ topic ? qos=1` »

L'URL du point de terminaison des données de l'AWS IoT appareil de votre client, suivie du port HTTPS :8443, qui est ensuite suivi du mot clé, /topics/ et du 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 AWS IoT console.

Suivez les instructions [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#) et configurez la console pour vous abonner aux messages avec le nom du sujet utilisé dans votre curl commande, ou utilisez le filtre de sujet 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

Les rubriques MQTT identifient les AWS IoT messages. AWS IoT clients identifient les messages qu'ils publient en attribuant des noms de sujets aux messages. 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.

Le courtier de messages utilise des rubriques pour identifier les messages envoyés via MQTT et envoyés via HTTP au [URL du message HTTPS \(p. 112\)](#).

Bien que AWS IoT prenne en charge certaines [rubriques système réservées \(p. 117\)](#), 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 \(p. 117\)](#) utilisées uniquement par AWS IoT Core.
- AWS IoT Corene peut pas envoyer ou recevoir de messages entre Compte AWS des régions ou des régions.

Pour plus d'informations sur la conception des noms de rubriques et de votre espace de nommage, consultez notre livre blanc, [Designing MQTT Topics for AWS IoT Core](#)

Pour obtenir des exemples de la façon dont les applications peuvent publier des messages et s'y abonner, commencez par [Démarrer avec AWS IoT Core \(p. 18\)](#) et [AWS IoT SDK pour appareils, kits SDK mobiles et client de AWS IoT l'appareil \(p. 1494\)](#).

Important

L'espace de nommage de la rubrique est limité à une région Compte AWS et. Par exemple, le sensor/temp/room1 sujet utilisé par un utilisateur d'une Compte AWS région est distinct de celui utilisé par le sensor/temp/room1 même AWS compte dans une autre région ou utilisé par un autre compte 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 sujet peut être un nom de sujet unique pour s'abonner à un seul nom de sujet ou il peut inclure des caractères génériques pour s'abonner à plusieurs noms de sujets en même temps.

Les clients de publication ne peuvent pas utiliser de caractères génériques dans les noms des rubriques 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 +.

Caractère générique	Correspondance	Remarques
+	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.

Filtre thématique ARN

Tous les ARN des filtres thématiques (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 rubrique `application/topic/+/-sensor`.

Chargement utile des messages MQTT

La charge utile des messages qui est envoyée dans vos messages MQTT n'est pas spécifiée par AWS IoT, sauf si c'est pour l'un des. [the section called "Rubriques réservées" \(p. 117\)](#) Pour répondre aux besoins de votre application, nous vous recommandons de définir la charge utile des messages pour vos sujets en respectant les contraintes des [AWS IoT CoreService Quotas pour les protocoles](#).

L'utilisation d'un format JSON pour la charge utile de vos messages permet au moteur de AWS IoT règles d'analyser vos messages et d'y appliquer des requêtes SQL. Si votre application n'a pas besoin du moteur de règles pour appliquer des requêtes SQL aux charges utiles de vos messages, vous pouvez utiliser n'importe quel format de données requis par votre application. Pour plus d'informations sur les limitations et les caractères réservés dans un document JSON utilisé dans les requêtes SQL, consultez [Extensions JSON \(p. 680\)](#).

Pour plus d'informations sur la conception de vos rubriques MQTT et des charges utiles de message correspondantes, consultez la section [Conception de rubriques MQTT](#) pour AWS IoT Core.

Si la limite de taille d'un message dépasse les quotas de service, cela se traduira par un message CLIENT_ERROR avec raison PAYLOAD_LIMIT_EXCEEDED et « La charge utile du message dépasse la limite de taille pour le type de message ». Pour plus d'informations sur la limite de taille des messages, consultez la section [Limites et quotas pour les courtiers de AWS IoT Core messages](#).

Rubriques réservées

Les rubriques qui commencent par un signe dollar (\$) sont réservées à l'utilisation par AWS IoT. Vous pouvez vous abonner à ces rubriques réservées et les publier dans la mesure où elles le permettent ; toutefois, vous ne pouvez pas créer de nouvelles rubriques commençant par le signe du 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
<code>\$aws/sitewise/asset-models/ /assets/ AssetId /properties/ assetModelIdpropertyId</code>	S'abonner	AWS IoT SiteWise publie des notifications sur les propriétés de ressource pour cette rubrique. Pour plus d'informations, consultez la section Interaction avec d'autres AWS services dans le Guide de AWS IoT SiteWise l'utilisateur.

Rubriques AWS IoT Device Defender

Ces messages prennent en charge les zones tampon de réponse au format CBOR (Concise Binary Object Representation) et au format JSON (JavaScriptObject Notation), en fonction du format de *charge utile du sujet*. AWS IoT Device Defender les rubriques ne prennent en charge que la publication MQTT.

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

Pour plus d'informations, veuillez consulter [Envoi de métriques à partir d'appareils \(p. 1125\)](#).

Sujet	Opérations autorisées	Description
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code>	Publier	AWS IoT Device Defender les agents publient des métriques dans cette rubrique. Pour plus d'informations, veuillez consulter Envoi de métriques à partir d'appareils (p. 1125) .
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i>/accepted</code>	S'abonner	AWS IoT publie dans cette rubrique une fois qu'un AWS IoT Device Defender agent a publié un message réussi sur <code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code> . Pour plus d'informations, veuillez consulter Envoi de métriques à partir d'appareils (p. 1125) .
<code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i>/rejected</code>	S'abonner	AWS IoT publie dans cette rubrique lorsqu'un AWS IoT Device Defender agent a publié un message infructueux dans <code>\$aws/things/<i>thingName</i>/defender/metrics/<i>payload-format</i></code> . Pour plus d'informations, veuillez consulter Envoi de métriques à partir d'appareils (p. 1125) .

AWS IoT CoreRubriques de localisation des appareils

AWS IoT CoreLa localisation de l'appareil peut résoudre les données de mesure de votre appareil et fournir une estimation de l'emplacement de vos appareils IoT. Les données de mesure provenant du dispositif peuvent inclure un ou plusieurs des types de mesures suivants : adresse GNSSWiFi, réseau cellulaire ou IP. AWS IoT Core Device Location choisit ensuite le type de mesure qui fournit la meilleure précision et résout les informations de localisation de l'appareil. Pour plus d'informations, consultez [AWS IoT CoreEmplacement de l'appareil \(p. 1234\)](#) et [Résolution de la localisation de l'appareil à l'aide des rubriques MQTT AWS IoT Core Device Location \(p. 1241\)](#).

Sujet	Opérations autorisées	Description
<code>\$aws/device/location/ customer_device_id / get_position_estimate</code>	Publier	Un appareil publie des informations sur cette rubrique pour obtenir les données de mesure brutes numérisées à résoudre par AWS IoT Core Device Location.
<code>\$aws/device/location/ customer_device_id / get_position_estimate/ accepted</code>	S'abonner	AWS IoT CoreL'emplacement de l'appareil publie dans cette rubrique une fois que la localisation de l'appareil a été correctement résolue.
<code>\$aws/device/location/ customer_device_id / get_position_estimate/ rejected</code>	S'abonner	AWS IoT CoreDevice Location publie dans cette rubrique lorsqu'il n'est pas en mesure de résoudre correctement la localisation de l'appareil en raison d'erreurs 4xx.

Rubriques d'événement

Sujet	Opérations autorisées du client	Description
<code>\$aws/événements/ certificats/enregistrés/ caCertificateId</code>	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, veuillez consulter the section called "Configuration de la première connexion par un client pour l'enregistrement automatique" (p. 335) .
<code>\$aws/events/job/ jobId /annulé</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est annulée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ job/ jobId / cancellation_in_progress</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est annulée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/job/ jobId /terminé</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est terminée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .

Sujet	Opérations autorisées du client	Description
<code>\$aws/events/job/ jobId /supprimé</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est supprimée. Pour plus d'informations, consultez Événements Jobs (p. 1264) .
<code>\$aws/events/ job/ jobId / deletion_in_progress</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est supprimée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId/ annulé</code>	S'abonner	AWS IoT publie ce message lorsque l'exécution d'une tâche est annulée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId / supprimé</code>	S'abonner	AWS IoT publie ce message lorsqu'une exécution de tâche est supprimée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId / failed</code>	S'abonner	AWS IoT publie ce message en cas d'échec de l'exécution d'une tâche. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId / rejeté</code>	S'abonner	AWS IoT publie ce message lorsque l'exécution d'une tâche a été refusée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId / supprimé</code>	S'abonner	AWS IoT publie ce message lorsqu'une exécution de tâche a été supprimée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/ jobExecution/ jobId / réussi</code>	S'abonner	AWS IoT publie ce message lorsqu'une tâche est exécutée avec succès. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/événements/ jobExecution/ jobId / timed_out</code>	S'abonner	AWS IoT publie ce message lorsque le délai d'exécution d'une tâche a expiré. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
<code>\$aws/events/presence/ connected/clientId</code>	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, veuillez consulter Événements de connexion/déconnexion (p. 1268) .

Sujet	Opérations autorisées du client	Description
\$aws/events/presence/disconnected/ <i>clientId</i>	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, veuillez consulter Événements de connexion/déconnexion (p. 1268) .
\$aws/events/subscriptions/subscribed/ <i>clientId</i>	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, veuillez consulter Événements d'abonnement/désabonnement (p. 1270) .
\$aws/events/subscriptions/unsubscribed/ <i>clientId</i>	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, veuillez consulter Événements d'abonnement/désabonnement (p. 1270) .
\$aws/events/thing/ <i>thingName</i> /created	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet <i>thingName</i> est créé. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .
\$aws/events/thing/ <i>thingName</i> /updated	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet <i>thingName</i> est mis à jour. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .
\$aws/events/thing/ <i>thingName</i> /deleted	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet <i>thingName</i> est supprimé. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .
\$aws/Events/ThingGroup/ / created <i>thingGroupName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets <i>thingGroupName</i> , est créé. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .
\$AWS/Events/ThingGroup/ / updated <i>thingGroupName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, <i>thingGroupName</i> , est mis à jour. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .
\$AWS/Events/ThingGroup/ / supprimé <i>thingGroupName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets <i>thingGroupName</i> , est supprimé. Pour plus d'informations, veuillez consulter the section called "Événements de registre" (p. 1257) .

Sujet	Opérations autorisées du client	Description
\$aws/Events/ThingType/ / created <i>thingTypeName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d' <i>thingTypeName</i> objet est créé. Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$AWS/Events/ThingType/ / updated <i>thingTypeName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d' <i>thingTypeName</i> objet est mis à jour. Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$AWS/Events/ThingType/ / supprimé <i>thingTypeName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque le type d' <i>thingTypeName</i> objet est supprimé. Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$aws/events/ / thing/ <i>thingName</i> / thingTypeAssociation <i>thingTypeName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, <i>thingName</i> , est associé ou dissocié du type d'objet, <i>thingTypeName</i> . Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$aws/events/ / ThingGroup/ / thing/ <i>ThingName</i> thingGroupMembership / ajouté <i>thingGroupName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, <i>thingName</i> , est ajouté au groupe d'objets, <i>thingGroupName</i> . Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$aws/events/ / ThingGroup/ / thing/ <i>ThingName</i> thingGroupMembership / supprimé <i>thingGroupName</i>	S'abonner	AWS IoT publie dans cette rubrique lorsque l'objet, <i>thingName</i> , est supprimé du groupe d'objets, <i>thingGroupName</i> . Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$aws/events/ / ThingGroup/ Nom//Nom thingGroupHierarchy / ajouté parentThingGroup childThingGroup childThingGroup	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, <i>childThingGroupNom</i> , est ajouté au groupe d'objets, <i>parentThingGroupNom</i> . Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .
\$aws/events/ / ThingGroup/ Nom//Nom thingGroupHierarchy / removed parentThingGroup childThingGroup childThingGroup	S'abonner	AWS IoT publie dans cette rubrique lorsque le groupe d'objets, <i>childThingGroupNom</i> , est supprimé du groupe d'objets, <i>parentThingGroupNom</i> . Pour plus d'informations, veuillez consulter the section called “Événements de registre” (p. 1257) .

Rubriques de mise en service d'une flotte

Note

Les opérations client répertoriées sous la forme Receive dans ce tableau indiquent les rubriques AWS IoT qui sont publiées directement pour le client qui en a fait la demande, que le client soit abonné à la rubrique ou non. Les clients doivent s'attendre à recevoir ces messages de réponse même s'ils n'y sont pas abonnés. Ces messages de réponse ne passent pas par l'intermédiaire du gestionnaire de messages et d'autres clients ou règles ne peuvent pas s'y abonner.

Ces messages prennent en charge les zones tampon de réponse au format CBOR (Concise Binary Object Representation) et au format JSON (JavaScriptObject Notation), en fonction du format de *charge utile du sujet*.

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

Pour plus d'informations, veuillez consulter [API MQTT de mise en service des appareils \(p. 923\)](#).

Sujet	Opérations autorisées du client	Description
\$aws/certificates/create/ <i>payload-format</i>	Publier	Pour créer un certificat à partir d'une demande de signature de certificat (CSR), publiez sur cette rubrique.
\$aws/certificates/create/ <i>payload-format</i> /accepted	Abonnez-vous, recevez	AWS IoT publie dans cette rubrique après un appel réussi à \$aws/certificates/create/ <i>payload-format</i> .
\$aws/certificates/create/ <i>payload-format</i> /rejected	Abonnez-vous, recevez	AWS IoT publie dans cette rubrique après un appel infructueux à \$aws/certificates/create/ <i>payload-format</i> .
create-from-csr\$aws/certificates//format de charge utile	Publier	Publie dans cette rubrique pour créer un certificat à partir d'un CSR.
\$aws/certificates//format de charge utile /accepté create-from-csr	Abonnez-vous, recevez	AWS IoT <i>publie dans cette rubrique un appel réussi à create-from-csr \$aws/certificates//payload-format</i> .
\$aws/certificates//format de charge utile /rejeté create-from-csr	Abonnez-vous, recevez	AWS IoT <i>publie dans cette rubrique un appel infructueux à create-from-csr \$aws/certificates//payload-format</i> .
\$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i>	Publier	Publiez dans cette rubrique pour enregistrer un objet.
\$aws/provisioning-templates/ <i>templateName</i>	Abonnez-vous, recevez	AWS IoT publie dans cette rubrique après un appel réussi à \$aws/

Sujet	Opérations autorisées du client	Description
provision/ <i>payload-format</i> /accepted		provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i> .
\$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i> /rejected	Abonnez-vous, recevez	AWS IoT publie dans cette rubrique après un appel infructueux à \$aws/provisioning-templates/ <i>templateName</i> /provision/ <i>payload-format</i> .

Rubriques de tâche

Note

Les opérations client répertoriées sous la forme Receive dans ce tableau indiquent les rubriques AWS IoT qui sont publiées directement pour le client qui en a fait la demande, que le client soit abonné à la rubrique ou non. Les clients doivent s'attendre à recevoir ces messages de réponse même s'ils n'y sont pas abonnés.

Ces messages de réponse ne passent pas par l'intermédiaire du gestionnaire de messages et d'autres clients ou règles ne peuvent pas s'y abonner. Pour vous abonner aux messages relatifs aux activités professionnelles, utilisez les notify-next rubriques notify et.

Lorsque vous vous abonnez aux rubriques relatives aux tâches et aux jobExecution événements de votre solution de surveillance de flotte, vous devez d'abord activer les [événements d'exécution des tâches et \(p. 1253\)](#) des tâches afin de recevoir tous les événements du côté cloud.

Pour plus d'informations, veuillez consulter [Opérations de l'API MQTT de l'appareil de tâches \(p. 824\)](#).

Sujet	Opérations autorisées du client	Description
\$aws/things/ <i>thingName</i> /jobs/get	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande GetPendingJobExecutions. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
\$aws/things/ <i>thingName</i> /jobs/get/accepted	Abonnez-vous, recevez	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande GetPendingJobExecutions. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
\$aws/things/ <i>thingName</i> /jobs/get/rejected	Abonnez-vous, recevez	Les appareils s'abonnent à cette rubrique pour recevoir une réponse lorsqu'une GetPendingJobExecutions demande est rejetée. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
\$aws/things/ <i>thingName</i> /jobs/start-next	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande

Sujet	Opérations autorisées du client	Description
		<code>StartNextPendingJobExecution.</code> Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/start-next/accepted</code>	Abonnez-vous, recevez	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande <code>StartNextPendingJobExecution</code> . Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/start-next/rejected</code>	Abonnez-vous, recevez	Les appareils s'abonnent à cette rubrique pour recevoir une réponse lorsqu'une <code>StartNextPendingJobExecution</code> demande est rejetée. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/get</code>	Publier	Les périphériques publient un message dans cette rubrique pour envoyer une demande <code>DescribeJobExecution</code> . Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/get/accepted</code>	Abonnez-vous, recevez	Les périphériques s'abonnent à cette rubrique pour recevoir des réponses positives à une demande <code>DescribeJobExecution</code> . Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/get/rejected</code>	Abonnez-vous, recevez	Les appareils s'abonnent à cette rubrique pour recevoir une réponse lorsqu'une <code>DescribeJobExecution</code> demande est rejetée. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update</code>	Publier	Les appareils publient un message dans cette rubrique pour envoyer une demande <code>UpdateJobExecution</code> . Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824) .

Sujet	Opérations autorisées du client	Description
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update/accepted</code>	Abonnez-vous, recevez	<p>Les appareils s'abonnent à cette rubrique pour recevoir des réponses positives à une demande <code>UpdateJobExecution</code>. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824).</p> <p>Remarque</p> <p>Seul l'appareil qui publie sur <code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update</code> reçoit les messages sur cette rubrique.</p>
<code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update/rejected</code>	Abonnez-vous, recevez	<p>Les appareils s'abonnent à cette rubrique pour recevoir une réponse lorsqu'une <code>UpdateJobExecution</code> demande est rejetée. Pour plus d'informations, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824).</p> <p>Remarque</p> <p>Seul l'appareil qui publie sur <code>\$aws/things/<i>thingName</i>/jobs/<i>jobId</i>/update</code> reçoit les messages sur cette rubrique.</p>
<code>\$aws/things/<i>thingName</i>/jobs/notify</code>	S'abonner	<p>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, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824).</p>
<code>\$aws/things/<i>thingName</i>/jobs/notify-next</code>	S'abonner	<p>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, veuillez consulter Opérations de l'API MQTT de l'appareil de tâches (p. 824).</p>
<code>\$aws/events/job/<i>jobId</i>/completed</code>	S'abonner	<p>Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est terminée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264).</p>
<code>\$aws/events/job/<i>jobId</i>/canceled</code>	S'abonner	<p>Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est annulée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264).</p>

Sujet	Opérations autorisées du client	Description
\$aws/events/job/ <i>jobId</i> /deleted	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsqu'une tâche est supprimée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/job/ <i>jobId</i> /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, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/job/ <i>jobId</i> /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, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /succeeded	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche aboutit. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /failed	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche échoue. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /rejected	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est rejetée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /canceled	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est annulée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /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, veuillez consulter Événements Jobs (p. 1264) .
\$aws/events/jobExecution/ <i>jobId</i> /removed	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est retirée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .

Sujet	Opérations autorisées du client	Description
\$aws/events/ jobExecution/ <i>jobId</i> /deleted	S'abonner	Le service Jobs publie un événement sur cette rubrique lorsque l'exécution d'une tâche est supprimée. Pour plus d'informations, veuillez consulter Événements Jobs (p. 1264) .

Rubriques de règle

Sujet	Opérations autorisées du client	Description
\$aws/rules/ <i>ruleName</i>	Publier	Un appareil ou une application publie dans cette rubrique pour déclencher des règles directement. Pour plus d'informations, veuillez consulter Réduire les coûts de messagerie grâce à Basic Ingest (p. 617) .

Rubriques liées au tunneling sécurisé

Sujet	Opérations autorisées du client	Description
\$aws/things/ <i>thing-name</i> / 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, veuillez consulter the section called “Extrait de l'agent IoT” (p. 883) .

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.

ShadowTopicPrefix valeur	Type de shadow
\$aws/things/ <i>thingName</i> /shadow	Shadow non nommé (classique)
\$aws/things/ <i>thingName</i> /shadow/ name/ <i>shadowName</i>	Shadow nommé

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

Sujet	Opérations autorisées du client	Description
<i>ShadowTopicPrefix/</i> supprimer	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. 727) .
<i>ShadowTopicPrefix/</i> supprimer/accepté	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. 728) .
<i>ShadowTopicPrefix/</i> supprimer/rejeté	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, veuillez consulter /delete/rejected (p. 728) .
<i>ShadowTopicPrefix/</i> obtenir	Publier/s'abonner	Une application ou un objet publie un message vide dans cette rubrique pour obtenir un shadow. Pour plus d'informations, veuillez consulter Rubriques MQTT de Device Shadow (p. 721) .
<i>ShadowTopicPrefix/</i> obtenir/accepté	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. 722) .
<i>ShadowTopicPrefix/</i> 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. 723) .
<i>ShadowTopicPrefix/</i> mettre à jour	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. 723) .
<i>ShadowTopicPrefix/</i> update/accepté	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. 725) .
<i>ShadowTopicPrefix/</i> update/rejeté	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. 726) .
<i>ShadowTopicPrefix/</i> update/delta	S'abonner	Le service Device Shadow envoie des messages à cette rubrique lorsqu'un écart est constaté entre les sections déclarées

Sujet	Opérations autorisées du client	Description
		et les sections souhaitées d'un shadow. Pour de plus amples informations, veuillez consulter /update/delta (p. 724) .
<i>ShadowTopicPrefix</i> /mise à jour/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. 726) .

Rubriques de livraison de fichiers basées sur MQTT

Ces messages prennent en charge les zones tampon de réponse au format CBOR (Concise Binary Object Representation) et au format JSON (JavaScriptObject Notation), en fonction du format de *charge utile du sujet*.

<i>payload-format</i>	Type de données du format de réponse
CBOR	CBOR (Concise Binary Object Representation, représentation concise d'objets binaires)
json	JavaScriptJavaScript (JSON)

Sujet	Opérations autorisées du client	Description
<i>format de charge utile \$aws/things/ /streams/ /data/ ThingNameStreamId</i>	S'abonner	AWS La livraison de fichiers basée sur MQTT est publiée dans cette rubrique si la demande « GetStream » provenant d'un appareil est acceptée. La charge utile contient les données du flux. Pour plus d'informations, veuillez consulter Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils (p. 967) .
<i>format de charge utile \$aws/things/ /streams/ /get/ ThingNameStreamId</i>	Publier	Un appareil publie des informations dans cette rubrique pour exécuter une GetStream requête « ». Pour plus d'informations, veuillez consulter Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils (p. 967) .
<i>\$aws/things/ /streams/ /description/ format ThingName de charge utile StreamId</i>	S'abonner	AWS La livraison de fichiers basée sur MQTT est publiée dans cette rubrique si la demande « DescribeStream » provenant d'un appareil est acceptée. La charge utile contient la description du flux. Pour plus d'informations, veuillez consulter Utilisation de AWS IoT la

Sujet	Opérations autorisées du client	Description
		livraison de fichiers basée sur MQTT sur les appareils (p. 967) .
<i>\$aws/things/ /streams/ /describe/ format ThingName de charge utile StreamId</i>	Publier	Un appareil publie des informations dans cette rubrique pour exécuter une DescribeStream requête « ». Pour plus d'informations, veuillez consulter Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils (p. 967) .
<i>format de charge utile \$aws/things/ /streams/ /rejected/ ThingNameStreamId</i>	S'abonner	AWS La livraison de fichiers basée sur MQTT est publiée dans cette rubrique si une demande DescribeStream « » ou « GetStream » provenant d'un appareil est rejetée. Pour plus d'informations, veuillez consulter Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils (p. 967) .

(ARN)

Tous les ARN de rubriques réservées (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 la rubrique réservée \$aws/things/thingName/jobs/get/accepted.

Points de terminaison configurables

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. Les configurations de domaine vous permettent de générer plusieurs points de terminaison de AWS IoT Core données, de personnaliser les points de terminaison de données avec vos propres noms de domaine complets (FQDN) et les certificats de serveur associés, et également d'associer un autorisateur personnalisé. Pour plus d'informations, veuillez consulter [Authentification et autorisation personnalisées \(p. 343\)](#).

Note

Cette fonctionnalité n'est pas disponible dans GovCloudRégions AWS.

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

- Migrez des appareils vers AWS IoT Core.
- 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 Core.

AWS IoT Core utilise l'extension TLS d'indication de nom de serveur (SNI) pour appliquer des configurations de domaine. Les appareils doivent utiliser cette extension lorsqu'ils se connectent. Ils doivent également transmettre un nom de serveur identique au nom de domaine que vous avez spécifié dans la configuration du domaine. Pour tester ce service, utilisez la version v2 des [kits SDK pour AWS IoT appareils](#) dans GitHub.

Si vous créez plusieurs points de terminaison de données dans votre Compte AWS, ils partageront AWS IoT Core des ressources telles que des rubriques MQTT, des ombres sur les appareils et des règles.

Lorsque vous fournissez les certificats de serveur pour une configuration de domaine AWS IoT Core personnalisée, les certificats comportent au maximum quatre noms de domaine. Pour de plus amples informations, consultez [Points de terminaison et quotas AWS IoT Core](#).

Rubriques

- [Création et configuration de domaines gérés par AWS \(p. 132\)](#)
- [Création et configuration de domaines personnalisés \(p. 133\)](#)
- [Gestion des configurations de domaine \(p. 136\)](#)
- [Configuration des paramètres TLS dans les configurations de domaine \(p. 137\)](#)

Création et configuration de domaines gérés par AWS

Vous créez un point de terminaison configurable sur un AWS domaine géré à l'aide de l'[CreateDomainConfiguration](#) API. 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 et dont la valeur doit être unique à votre Région AWS. Vous ne pouvez pas utiliser de noms de configuration de domaine commençant par IoT : car ils sont réservés aux points de terminaison par défaut.

- **defaultAuthorizerName** (facultatif)

Le nom de l'autorisateur personnalisé à utiliser sur le terminal.

- **allowAuthorizerOverride**

Valeur booléenne qui indique si les appareils peuvent remplacer l'autorisateur par défaut en spécifiant un autre dans l'en-tête HTTP de la demande. Cette valeur est requise si une valeur est spécifiée pour defaultAuthorizerName.

- **serviceType**

Type de service fourni par le point de terminaison. AWS IoT Core prend en charge que le type de DATA service. Lorsque vous le spécifiez DATA, AWS IoT Core renvoie un point de terminaison avec un type de point de terminaison de `iot:Data-ATS`. Vous ne pouvez pas créer de point de terminaison configurable `iot:Data` (VeriSign).

- **TlsConfig** (facultatif)

Objet qui spécifie la configuration TLS pour un domaine. Pour plus d'informations, veuillez consulter [???](#) (p. 137).

L'exemple de AWS CLI commande suivant crée une configuration de domaine pour un Data point de terminaison.

```
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 Core. Les domaines personnalisés vous permettent de gérer vos propres certificats de serveur afin de pouvoir gérer des détails tels que l'autorité de certification (CA) racine utilisée pour signer le certificat, l'algorithme de signature, la profondeur 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 \(p. 133\)](#)
2. [Création d'une configuration de domaine \(p. 134\)](#)
3. [Création d'enregistrements DNS \(p. 135\)](#)

Enregistrement des certificats de serveur dans le gestionnaire de AWS certificats

Avant de créer une configuration de domaine avec un domaine personnalisé, vous devez enregistrer votre chaîne de certificats de serveur dans [AWS Certificate Manager\(ACM\)](#). Vous pouvez utiliser les trois types de certificats de serveur suivants.

- [Certificats publics générés par ACM \(p. 134\)](#)
- [Certificats externes signés par une autorité de certification publique \(p. 134\)](#)
- [Certificats externes signés par une autorité de certification privée \(p. 134\)](#)

Note

AWS IoT Core 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](#).

Exigences relatives aux certificats

Reportez-vous à la section [Conditions requises pour l'importation de certificats](#) pour connaître les exigences relatives à l'importation de certificats dans ACM. En plus de ces exigences, AWS IoT Core ajoute les exigences suivantes.

- Le certificat feuille doit inclure l'extension Extended Key Usage x509 v3 avec la valeur ServerAuth (authentification du serveur Web TLS). Si vous demandez le certificat auprès d'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 certificats est de 16 Ko.

Utilisation d'un seul 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. Si des SAN sont présents, le CN est ignoré et un SAN doit couvrir le nom de domaine complet que vous souhaitez utiliser comme domaine personnalisé. Cette couverture peut être une correspondance exacte ou une correspondance générique. Une fois qu'un certificat générique a été validé et enregistré sur un compte, les autres comptes de la région sont empêchés de créer des domaines personnalisés qui recoupent le certificat.

Les sections suivantes décrivent comment obtenir chaque type de certificat. Chaque ressource de certificat nécessite un Amazon Resource Name (ARN) enregistré dans pour 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'[RequestCertificateAPI](#). 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](#) dans le Guide de l'utilisateur AWS Certificate Manager.

Certificats externes signés par une autorité de certification publique

Si vous possédez déjà un certificat de serveur signé par une autorité de certification publique (une autorité de certification incluse dans le bundle CA sécurisé de Mozilla), vous pouvez importer la chaîne de certificats directement dans ACM à l'aide de l'API. [ImportCertificate](#) 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 possédez déjà un certificat de serveur signé par une autorité de certification privée ou autosigné, vous pouvez utiliser le certificat pour créer la configuration de votre 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'[ImportCertificateAPI](#). 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](#).

Création d'un certificat de validation

Après avoir importé votre certificat dans ACM, générez un certificat public pour votre domaine personnalisé à l'aide de l'[RequestCertificateAPI](#). 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'[CreateDomainConfigurationAPI](#). 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. Les noms de configuration de domaine commençant par IoT : sont réservés aux points de terminaison par défaut et ne peuvent pas être utilisés. De plus, cette valeur doit être unique à votreRégion AWS.

- `domainName`

Le nom de domaine complet que vos appareils utilisent pour se connecterAWS IoT Core. AWS IoT Core utilise l'extension TLS SNI (Server Name Indication) pour appliquer les 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 de serveur que vous avez enregistrée dans dans pour toutes les demandes dans pour toutes les demandes que vous avez enregistrées auprès d' AWS IoT Corene prend actuellement en charge qu'un seul certificat de serveur.

- `validationCertificateArn`

ARN du certificat public que vous avez généré dans pour valider la propriété de votre domaine personnalisé. Cet argument n'est pas obligatoire si vous utilisez un certificat de serveur signé publiquement ou généré par ACM.

- **defaultAuthorizerName (optional)**

Le nom de l'autorisateur personnalisé à utiliser sur le terminal.

- **allowAuthorizerOverride**

Valeur booléenne qui indique si les appareils peuvent remplacer l'autorisateur par défaut en spécifiant un autre dans l'en-tête HTTP de la demande. Cette valeur est requise si une valeur est spécifiée pour `defaultAuthorizerName`.

- **serviceType**

AWS IoT Core ne prend actuellement en charge que le type de service DATA. Lorsque vous le spécifiez DATA, AWS IoT renvoie un point de terminaison avec un type de point de terminaison `deiot:Data-ATS`.

- **TlsConfig (facultatif)**

Objet qui spécifie la configuration TLS pour un domaine. Pour plus d'informations, veuillez consulter [??? \(p. 137\)](#).

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

Une fois que vous avez créé la configuration de votre domaine, la fourniture de vos certificats de serveur personnalisés peut prendre jusqu'à AWS IoT Core 60 minutes.

Pour plus d'informations, veuillez consulter [??? \(p. 136\)](#).

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'[DescribeEndpointAPI](#).

La AWS CLI commande suivante 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 terminalCompte AWS, attribuez-leur un alias à ce même `iot:Data-ATS` point de terminaison.

Résolution des problèmes

Si vous ne parvenez pas à connecter des appareils à un domaine personnalisé, assurez-vous que AWS IoT Core celui-ci a accepté et appliqué votre certificat de serveur. Vous pouvez vérifier que votre certificat AWS IoT Core a bien été accepté en utilisant la AWS IoT Core console ou le AWS CLI.

Pour utiliser la AWS IoT Core console, accédez à la page Paramètres et sélectionnez le nom de configuration du domaine. Dans la section Détails du certificat du serveur, vérifiez l'état et les détails du statut. Si le certificat n'est pas valide, remplacez-le dans ACM par un certificat répondant aux [exigences de certification \(p. 133\)](#) répertoriées dans la section précédente. Si le certificat possède le même ARN, AWS IoT Core nous le récupérerons et l'appliquerons automatiquement.

Pour vérifier l'état du certificat à l'aide du AWS CLI, appelez l'[DescribeDomainConfiguration](#) API et spécifiez le nom de configuration de votre domaine.

Note

Si votre certificat n'est pas valide, AWS IoT Core nous continuerons à délivrer le dernier certificat valide.

Vous pouvez vérifier quel certificat est fourni sur votre terminal 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

Pour renvoyer une liste paginée de toutes les configurations de domaine de votre Compte AWS, utilisez l'[ListDomainConfigurations](#) API. Vous pouvez voir les détails d'une configuration de domaine particulière à l'aide de l'[DescribeDomainConfiguration](#) API. Cette API prend un seul paramètre domainConfigurationName et renvoie les détails de la configuration spécifiée.

Example (Exemple)

Mise à jour des configurations de domaine

Pour mettre à jour le statut ou l'autorisateur personnalisé de la configuration de votre domaine, utilisez l'[UpdateDomainConfiguration](#) API. 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. 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.

Example (Exemple)

Supprimer des configurations de domaine

Avant de supprimer la configuration d'un domaine, utilisez l'[UpdateDomainConfiguration](#) API pour définir le statut 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'[DeleteDomainConfiguration](#) API. Vous devez attribuer AWS un DISABLED statut aux domaines gérés pendant 7 jours avant de pouvoir les supprimer. Vous pouvez placer des domaines personnalisés dans DISABLED le statut, puis les supprimer immédiatement.

Example (Exemple)

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

Il se peut que vous deviez régulièrement remplacer votre certificat de serveur par un certificat mis à jour. Le rythme auquel vous le faites dépend de la période de validité de votre certificat. Si vous avez généré votre certificat de serveur à l'aide de AWS Certificate Manager (ACM), vous pouvez configurer le certificat pour qu'il soit renouvelé automatiquement. Lorsque ACM renouvelle votre certificat, il récupère AWS IoT Core automatiquement le nouveau certificat. Vous n'avez aucune action supplémentaire à effectuer. Si vous avez importé votre certificat de serveur depuis une autre source, vous pouvez le faire pivoter en le réimportant dans ACM. Pour plus d'informations sur la réimportation de certificats, voir [Réimporter un certificat](#).

Note

AWS IoT Core récupère les mises à jour des certificats que dans les conditions suivantes.

- Le nouveau certificat possède le même ARN que l'ancien.
- Le nouveau certificat possède le même algorithme de signature, le même nom commun ou le même nom de sujet que l'ancien.

Configuration des paramètres TLS dans les configurations de domaine

AWS IoT Core fournit des [politiques de sécurité prédéfinies \(p. 410\)](#) vous permettant de personnaliser vos paramètres TLS (Transport Layer Security) pour [TLS 1.2 et TLS 1.3](#) dans les configurations de domaine. Une politique de sécurité est une combinaison de protocoles TLS et de leurs chiffrements qui déterminent les protocoles et les chiffrements pris en charge lors des négociations TLS entre un client et un serveur. Grâce aux politiques de sécurité prises en charge, vous pouvez gérer les paramètres TLS de vos appareils avec plus de flexibilité, appliquer les mesures de up-to-date sécurité les plus strictes lors de la connexion de nouveaux appareils et maintenir des configurations TLS cohérentes pour les appareils existants.

Le tableau suivant décrit les stratégies de sécurité, leurs versions TLS et régions prises en charge :

Nom de la politique de sécurité	Versions TLS	Régions AWS prises en charge
IoT SecurityPolicy _TLS13_1_3_2022_10	TLS 1.3 1.3 1.3 1.3 1.3 1.3 1.3	Toutes les Régions AWS
IoT SecurityPolicy _TLS13_1_2_2022_10	TLS 1.2 + 1.3 1.3 1.3 1.2 + 1.3 1.3	Toutes les Régions AWS
IoT SecurityPolicy _TLS12_1_2_2022_10	TLS 1.2 1.2 1.2 1.2, uniquement disponible	Toutes les Régions AWS
IoT SecurityPolicy _TLS12_1_0_2016_01	TLS 1.0 + 1.1 + 1.2	ap-east-1, ap-northeast-1, ap-southeast-2, ca-west-1, us-west-1, us-west-1, eu-northeast-1, eu-west-1, eu-west-2, eu-west-1,

Les noms des politiques de sécurité AWS IoT Core incluent des informations de version basées sur l'année et le mois de leur publication. Si vous créez une nouvelle configuration de domaine, la politique de sécurité par défaut est `IoTSecurityPolicy_TLS13_1_2_2022_10`. Pour un tableau complet des politiques de sécurité avec des détails sur les protocoles, les ports TCP et les chiffrements, voir Politiques de sécurité. (p. 410) AWS IoT Core ne prend pas en charge les stratégies de sécurité personnalisées. Pour plus d'informations, veuillez consulter ??? (p. 409).

Pour configurer les paramètres TLS dans les configurations de domaine, vous pouvez utiliser la AWS IoT console ou le AWS CLI.

Table des matières

- [Configurer les paramètres TLS dans les configurations de domaine \(console\) \(p. 138\)](#)
 - [Configurer les paramètres TLS dans les configurations de domaine \(CLI\) \(p. 139\)](#)

Configurer les paramètres TLS dans les configurations de domaine (console)

Pour configurer les paramètres TLS à l'aide de la console AWS IoT

1. Connectez-vous à AWS Management Console et ouvrez la [console AWS IoT](#).
 2. Pour configurer les paramètres TLS lorsque vous créez une nouvelle configuration de domaine, procédez comme suit.
 1. Dans le volet de navigation de gauche, choisissez Paramètres, puis, dans la section Configurations de domaine, choisissez Créez une configuration de domaine.
 2. Sur la page Créez une configuration de domaine, dans la section Paramètres de domaine personnalisés - facultatif, choisissez une politique de sécurité dans Sélectionner une politique de sécurité.
 3. Suivez le widget et suivez le reste des étapes. Choisissez Créez une configuration de domaine.
 3. Pour mettre à jour les paramètres TLS dans une configuration de domaine existante, procédez comme suit.
 1. Dans le volet de navigation de gauche, choisissez Paramètres, puis, sous Configurations de domaine, choisissez une configuration de domaine.
 2. Sur la page des détails de configuration du domaine, choisissez Modifier. Ensuite, dans la section Paramètres de domaine personnalisés - facultatif, sous Sélectionner une politique de sécurité, choisissez une politique de sécurité.
 3. Choisissez Mettre à jour la configuration du domaine.

Pour plus d'informations, voir [Création d'une configuration de domaine](#) et [Gérer les configurations de domaine \(p. 136\)](#).

Configurer les paramètres TLS dans les configurations de domaine (CLI)

Vous pouvez utiliser les commandes [create-domain-configuration](#) et [update-domain-configuration](#) CLI pour configurer vos paramètres TLS dans les configurations de domaine.

1. Pour spécifier les paramètres TLS à l'aide de la [create-domain-configuration](#) commande CLI :

```
aws iot create-domain-configuration \
--domain-configuration-name domainConfigurationName \
--tls-config securityPolicy=IoTSecurityPolicy_TLS13_1_2_2022_10
```

La sortie de cette commande peut ressembler à ce qui suit :

```
{  
  "domainConfigurationName": "test",  
  "domainConfigurationArn": "arn:aws:iot:us-west-2:123456789012:domainconfiguration/  
  test/34ga9"  
}
```

Si vous créez une nouvelle configuration de domaine sans spécifier la politique de sécurité, la valeur par défaut est :`IoTSecurityPolicy_TLS13_1_2_2022_10`.

2. Pour décrire les paramètres TLS à l'aide de la [describe-domain-configuration](#) commande CLI :

```
aws iot describe-domain-configuration \
--domain-configuration-name domainConfigurationName
```

Cette commande peut renvoyer les détails de configuration du domaine, y compris les paramètres TLS, comme suit :

```
{  
  "tlsConfig": {  
    "securityPolicy": "IoTSecurityPolicy_TLS13_1_2_2022_10"  
  },  
  "domainConfigurationStatus": "ENABLED",  
  "serviceType": "DATA",  
  "domainType": "AWS_MANAGED",  
  "domainName": "d1234567890abcdefhijkl-ats.iot.us-west-2.amazonaws.com",  
  "serverCertificates": [],  
  "lastStatusChangeDate": 1678750928.997,  
  "domainConfigurationName": "test",  
  "domainConfigurationArn": "arn:aws:iot:us-west-2:123456789012:domainconfiguration/  
  test/34ga9"  
}
```

3. Pour mettre à jour les paramètres TLS à l'aide de la [update-domain-configuration](#) commande CLI :

```
aws iot update-domain-configuration \
--domain-configuration-name domainConfigurationName \
--tls-config securityPolicy=IoTSecurityPolicy_TLS13_1_2_2022_10
```

La sortie de cette commande peut ressembler à ce qui suit :

```
{  
  "domainConfigurationName": "test",  
  "domainConfigurationArn": "arn:aws:iot:us-west-2:123456789012:domainconfiguration/  
  test/34ga9"  
}
```

4. Pour mettre à jour les paramètres TLS de votre point de terminaison ATS, exécutez la [update-domain-configuration](#) commande CLI. Le nom de configuration de domaine de votre point de terminaison ATS est `iot:Data-ATS`.

```
aws iot update-domain-configuration \
--domain-configuration-name "iot:Data-ATS" \
--tls-config securityPolicy=IoTSecurityPolicy_TLS13_1_2_2022_10
```

La sortie de la commande peut ressembler à ce qui suit :

```
{  
  "domainConfigurationName": "iot:Data-ATS",  
  "domainConfigurationArn": "arn:aws:iot:us-west-2:123456789012:domainconfiguration/  
  iot:Data-ATS"  
}
```

Pour plus d'informations, veuillez consulter les sections [CreateDomainConfiguration](#) et [UpdateDomainConfiguration](#) (français non garanti) de la Référence d'API AWS.

Connexion aux points de AWS IoT terminaison FIPS

AWS IoT fournit des points de terminaison compatibles avec la [Federal Information Processing Norme fédérale de traitement de l'information \(FIPS\) 140-2](#). Les points de terminaison conformes à la norme FIPS sont différents des points de terminaison standard AWS. Pour interagir d'une manière conforme AWS IoT à la norme FIPS, vous devez utiliser les points de terminaison décrits ci-dessous avec votre client conforme à la norme FIPS. La AWS IoT console n'est pas conforme à la norme FIPS.

Les sections suivantes décrivent comment accéder aux AWS IoT points de terminaison conformes à la norme FIPS à l'aide de l'API REST, d'un SDK ou du AWS CLI

Rubriques

- [AWS IoT Core- points d'extrémité du plan de contrôle \(p. 140\)](#)
- [AWS IoT Core- points de terminaison du plan de données \(p. 141\)](#)
- [AWS IoT Device Management- points de terminaison des données sur les emplois \(p. 141\)](#)
- [AWS IoT Device Management- Points de terminaison Fleet Hub \(p. 141\)](#)
- [AWS IoT Device Management- points de terminaison de tunneling sécurisés \(p. 142\)](#)

AWS IoT Core- points d'extrémité du plan de contrôle

Les points de terminaison du plan de contrôle conformes AWS IoT Core à la norme FIPS qui prennent en charge les [AWS IoT opérations](#) et leurs [commandes CLI](#) associées sont répertoriés dans la section Points de terminaison [FIPS](#) par service. Dans [FIPS Endpoints by Service](#), recherchez le service AWS IoT Core-control plane et recherchez le point de terminaison correspondant à votre Région AWS

Pour utiliser le point de terminaison conforme à la norme FIPS lorsque vous accédez aux [AWS IoT opérations](#), utilisez le AWS SDK ou l'API REST avec le point de terminaison qui convient à votre Région AWS

Pour utiliser le point de terminaison conforme à la norme FIPS lorsque vous exécutez des [commandes aws iot CLI](#), ajoutez le --endpoint paramètre avec le point de terminaison approprié Région AWS à la commande.

AWS IoT Core- points de terminaison du plan de données

Les points de terminaison du plan de données conformes AWS IoT Core à la norme FIPS sont répertoriés dans la section Points de [terminaison FIPS](#) par service. Dans [FIPS Endpoints by Service](#), recherchez le service AWS IoT Core- data plane et recherchez le point de terminaison correspondant à votre Région AWS

Vous pouvez utiliser le point de terminaison conforme à la norme FIPS pour votre Région AWS avec un client conforme à la norme FIPS en utilisant le SDK AWS IoT Device et en fournissant le point de terminaison à la fonction de connexion du SDK au lieu du point de terminaison par défaut de votre compte, à savoir le point de terminaison du plan de données AWS IoT Core. La fonction de connexion est spécifique au AWS IoT Device SDK. Pour un exemple de fonction de connexion, consultez la [fonction de connexion dans le AWS IoT Device SDK pour Python](#).

Note

AWS IoT prend pas en charge les points de terminaison Compte AWS spécifiques AWS IoT Core du plan de données qui sont conformes à la norme FIPS. Les fonctionnalités de service qui nécessitent un point de terminaison Compte AWS spécifique dans l'[indication du nom du serveur \(SNI\) \(p. 409\)](#) ne peuvent pas être utilisées. [Conforme à la norme FIPS AWS IoT Core: les terminaux du plan de données ne peuvent pas prendre en charge les certificats d'enregistrement multicomptes \(p. 321\), les domaines personnalisés, les autorisateurs \(p. 343\) personnalisés \(p. 133\) et les points de terminaison configurables \(p. 131\) \(y compris les politiques TLS prises en charge\)](#). (p. 410)

AWS IoT Device Management- points de terminaison des données sur les emplois

Les points de terminaison des données sur les tâches conformes AWS IoT Device Management à la norme FIPS sont répertoriés dans la section Points de [terminaison FIPS](#) par service. Dans [FIPS Endpoints by Service](#), recherchez le service de données AWS IoT Device Management - jobs et recherchez le point de terminaison correspondant à votre Région AWS

Pour utiliser le point de terminaison de données conforme à la norme FIPS AWS IoT Device Management - jobs lorsque vous exécutez des [commandes aws iot-jobs-data CLI](#), ajoutez le --endpoint paramètre avec le point de terminaison approprié Région AWS à la commande. Vous pouvez également utiliser l'API REST avec ce point de terminaison.

Vous pouvez utiliser le point de terminaison conforme à la norme FIPS pour vous Région AWS avec un client conforme à la norme FIPS en utilisant le SDK AWS IoT Device et en fournissant le point de terminaison à la fonction de connexion du SDK au lieu du point de terminaison par défaut AWS IoT Device Management de votre compte, à savoir les données sur les tâches. La fonction de connexion est spécifique au AWS IoT Device SDK. Pour un exemple de fonction de connexion, consultez la [fonction de connexion dans le AWS IoT Device SDK pour Python](#).

AWS IoT Device Management- Points de terminaison Fleet Hub

Les points de terminaison Fleet Hub conformes AWS IoT Device Management à la norme FIPS à utiliser avec les [commandes CLI de Fleet Hub for AWS IoT Device Management](#) sont répertoriés dans la section

Points de terminaison [FIPS](#) par service. Dans [FIPS Endpoints by Service](#), recherchez le service AWS IoT Device Management- Fleet Hub et recherchez le point de terminaison correspondant à votre. Région AWS

Pour utiliser le point de terminaison Fleet Hub conforme AWS IoT Device Management à la norme FIPS lorsque vous exécutez des [commandes aws iotfleethub CLI](#), ajoutez le --endpoint paramètre avec le point de terminaison approprié Région AWS à la commande. Vous pouvez également utiliser l'API REST avec ce point de terminaison.

AWS IoT Device Management- points de terminaison de tunneling sécurisés

[Les points de terminaison de tunneling sécurisés conformes AWS IoT Device Management à la norme FIPS pour l'API de tunneling AWS IoT sécurisé et les commandes CLI correspondantes sont répertoriés dans FIPS Endpoints by Service](#). Dans [FIPS Endpoints by Service](#), recherchez le AWS IoT Device Management service de tunneling sécurisé et recherchez le point de terminaison correspondant à votre. Région AWS

Pour utiliser le point de terminaison de tunneling sécurisé conforme AWS IoT Device Management à la norme FIPS lorsque vous exécutez des [commandes aws iotsecuretunneling CLI](#), ajoutez le --endpoint paramètre avec le point de terminaison approprié à votre Région AWS commande. Vous pouvez également utiliser l'API REST avec ce point de terminaison.

Didacticiels AWS IoT

LeAWS IoTles tutoriels sont divisés en deux parcours d'apprentissage pour soutenir deux objectifs différents. Choisissez le parcours d'apprentissage le mieux adapté à votre objectif.

- Vous souhaitez créer un proof-of-concept pour tester ou démontrer unAWS IoTidée de solution

Pour démontrer les tâches et applications IoT courantes à l'aide duAWS IoTDevice Client sur vos appareils, suivez le[the section called “Construire des démonstrations avec leAWS IoTClient d'appareil” \(p. 143\)](#)parcours d'apprentissage. LeAWS IoTDevice Client fournit un logiciel d'appareil avec lequel vous pouvez appliquer vos propres ressources cloud pour démontrer un end-to-end solution avec un développement minimal.

Pour obtenir des informations sur leAWS IoTDevice Client, consultez le[AWS IoTClient d'appareil](#).

- Vous souhaitez apprendre comment créer un logiciel de production pour déployer votre solution

Pour créer votre propre logiciel de solution qui répond à vos exigences spécifiques à l'aide d'unAWS IoTSDK pour appareil, suivez le[the section called “Construire des solutions avec leAWS IoTKits SDK pour les appareils” \(p. 196\)](#)parcours d'apprentissage.

Pour obtenir des informations sur les options disponiblesAWS IoTKits SDK pour les appareils, consultez[??? \(p. 1494\)](#). Pour obtenir des informations sur leAWSKits SDK, consultez[Outils sur lesquels s'appuyerAWS](#).

AWS IoTOptions de parcours de didacticiel

- [Construire des démonstrations avec leAWS IoTClient d'appareil \(p. 143\)](#)
- [Construire des solutions avec leAWS IoTKits SDK pour les appareils \(p. 196\)](#)

Construire des démonstrations avec leAWS IoTClient d'appareil

Les tutoriels de ce parcours d'apprentissage vous guident à travers les étapes à suivre pour développer un logiciel de démonstration à l'aide duAWS IoTClient de l'appareil. LeAWS IoTDevice Client fournit un logiciel qui s'exécute sur votre appareil IoT pour tester et démontrer les aspects d'une solution IoT basée surAWS IoT.

L'objectif de ces tutoriels est de faciliter l'exploration et l'expérimentation afin que vous puissiez vous sentir sûr queAWS IoTprend en charge votre solution avant de développer le logiciel de votre appareil.

Ce que vous apprendrez dans ces tutoriels :

- Comment préparer un Raspberry Pi pour une utilisation comme appareil IoT avecAWS IoT
- Comment faire une démonstrationAWS IoTfonctionnalités à l'aide de laAWS IoTDevice Client sur votre appareil

Dans ce parcours d'apprentissage, vous allez installer leAWS IoTDevice Client sur votre propre Raspberry Pi et créez leAWS IoTressources dans le cloud pour démontrer les idées de solutions IoT. Bien que les tutoriels de ce parcours d'apprentissage présentent les fonctionnalités à l'aide d'un Raspberry Pi, ils expliquent les objectifs et les procédures qui vous aideront à les adapter à d'autres appareils.

Conditions préalables à la création de démonstrations avec leAWS IoTClient d'appareil

Cette section décrit ce que vous devez avoir avant de commencer les didacticiels de ce parcours d'apprentissage.

Pour compléter les didacticiels de ce parcours d'apprentissage, vous aurez besoin des éléments suivants :

- Un Compte AWS

Vous pouvez utiliser votre appareil existantCompte AWS, si vous en possédez un, mais vous devrez peut-être ajouter des rôles ou des autorisations supplémentaires pour utiliser leAWS IoTprésente ces didacticiels utilisés.

Si vous devez créer un nouveauCompte AWS, voir[the section called “Configurez votre Compte AWS” \(p. 19\)](#).

- Un Raspberry Pi ou un appareil IoT compatible

Les didacticiels utilisent un[Raspberry Pi](#)car il existe différents facteurs de forme, il est omniprésent et c'est un appareil de démonstration relativement peu coûteux. Les didacticiels ont été testés sur le[Raspberry Pi 3 Modèle B+](#), le[Raspberry Pi 4 Modèle B](#), et sur une instance Amazon EC2 exécutant Ubuntu Server 20.04 LTS (HVM). Pour utiliser le pluginAWS CLItexécutez les commandes, nous vous recommandons d'utiliser la dernière version du Raspberry Pi OS ([Raspberry Pi OS \(64 bits\)](#)ou OS Lite). Les versions antérieures du système d'exploitation pourraient fonctionner, mais nous ne l'avons pas testé.

Note

Les didacticiels expliquent les objectifs de chaque étape pour vous aider à les adapter au matériel IoT sur lequel nous ne les avons pas essayés. Cependant, ils ne décrivent pas spécifiquement comment les adapter à d'autres appareils.

- Connaissance du système d'exploitation de l'appareil IoT

Les étapes de ces didacticiels supposent que vous connaissez l'utilisation des commandes et des opérations Linux de base à partir de l'interface de ligne de commande prise en charge par un Raspberry Pi. Si vous n'êtes pas familier avec ces opérations, vous voudrez peut-être vous laisser plus de temps pour terminer les didacticiels.

Pour compléter ces didacticiels, vous devez déjà comprendre comment :

- Effectuez en toute sécurité les opérations de base de l'appareil telles que l'assemblage et la connexion de composants, la connexion de l'appareil aux sources d'alimentation requises et l'installation et le retrait de cartes mémoire.
- Téléchargez et téléchargez le logiciel système et les fichiers sur l'appareil. Si votre appareil n'utilise pas de périphérique de stockage amovible, tel qu'une carte microSD, vous devez savoir comment vous connecter à votre appareil et charger et télécharger le logiciel système et les fichiers sur l'appareil.
- Connect votre appareil aux réseaux sur lesquels vous envisagez de l'utiliser.
- Connect à votre appareil depuis un autre ordinateur à l'aide d'un terminal SSH ou d'un programme similaire.
- Utilisez une interface de ligne de commande pour créer, copier, déplacer, renommer et définir les autorisations des fichiers et des répertoires sur l'appareil.
- Installez de nouveaux programmes sur l'appareil.
- Transférez des fichiers depuis et vers votre appareil à l'aide d'outils tels que FTP ou SCP.
- Un environnement de développement et de test pour votre solution IoT

Les didacticiels décrivent le logiciel et le matériel requis. Cependant, les didacticiels supposent que vous serez en mesure d'effectuer des opérations qui peuvent ne pas être décrites explicitement. Voici quelques exemples de tels matériaux et opérations :

- Un ordinateur hôte local sur lequel télécharger et stocker des fichiers

Pour le Raspberry Pi, il s'agit généralement d'un ordinateur personnel ou d'un ordinateur portable capable de lire et d'écrire sur des cartes mémoire microSD. L'ordinateur hôte local doit :

- Soyez connecté à Internet.
- Avoir le[AWS CLI](#)installé et configuré.
- Disposer d'un navigateur Web qui prend en charge leAWSconsole
- Un moyen de connecter votre ordinateur hôte local à votre appareil pour communiquer avec lui, entrer des commandes et transférer des fichiers

Sur le Raspberry Pi, cela se fait souvent à l'aide de SSH et de SCP de l'ordinateur hôte local.

- Un moniteur et un clavier pour se connecter à votre appareil IoT

Ceux-ci peuvent être utiles, mais ne sont pas nécessaires pour compléter les didacticiels.

- Un moyen pour votre ordinateur hôte local et vos appareils IoT de se connecter à Internet

Il peut s'agir d'une connexion réseau câblée ou sans fil à un routeur ou à une passerelle connecté à Internet. L'hôte local doit également pouvoir se connecter au Raspberry Pi. Cela peut nécessiter qu'ils se trouvent sur le même réseau local. Les didacticiels ne peuvent pas vous montrer comment le configurer pour la configuration de votre appareil ou de votre appareil particulier, mais ils montrent comment tester cette connectivité.

- Accès au routeur de votre réseau local pour afficher les appareils connectés

Pour terminer les didacticiels de ce parcours d'apprentissage, vous devez être en mesure de trouver l'adresse IP de votre appareil IoT.

Sur un réseau local, cela peut être fait en accédant à l'interface d'administration du routeur réseau auquel vos appareils se connectent. Si vous pouvez attribuer une adresse IP fixe à votre appareil dans le routeur, vous pouvez simplifier la reconnexion après chaque redémarrage de l'appareil.

Si vous disposez d'un clavier et d'un moniteur connectés à l'appareil, ifconfig peut afficher l'adresse IP de l'appareil.

Si aucune de ces options ne constitue une option, vous devez trouver un moyen d'identifier l'adresse IP de l'appareil après chaque redémarrage.

Une fois que vous avez tous vos matériaux, continuez à[the section called “Préparation de vos appareils pour leAWS IoTClient d'appareil” \(p. 146\)](#).

Tutoriels sur ce parcours d'apprentissage

- [Didacticiel : Préparation de vos appareils pour leAWS IoTClient d'appareil \(p. 146\)](#)
- [Didacticiel : Installation et configuration de l'AWS IoTClient d'appareil \(p. 156\)](#)
- [Didacticiel : Démontrer la communication des messages MQTT avec leAWS IoTAppareil client \(p. 165\)](#)
- [Didacticiel : Démontrez des actions à distance \(tâches\) avecAWS IoTPériphérique Périphérique \(p. 179\)](#)
- [Didacticiel : Nettoyage après l'exécutionAWS IoTTutoriels Device Client \(p. 189\)](#)

Didacticiel : Préparation de vos appareils pour le AWS IoTClient d'appareil

Ce tutoriel vous guide tout au long de l'initialisation de votre Raspberry Pi pour le préparer aux tutoriels suivants de ce parcours d'apprentissage.

L'objectif de ce didacticiel est d'installer la version actuelle du système d'exploitation de l'appareil et de s'assurer que vous pouvez communiquer avec votre appareil dans le contexte de votre environnement de développement.

Pour démarrer ce didacticiel :

- Avoir les articles répertoriés dans [the section called “Conditions préalables à la création de démonstrations avec le AWS IoTClient d'appareil” \(p. 144\)](#) disponible et prêt à être utilisée.

Ce didacticiel vous prendra environ 90 minutes.

Lorsque vous avez terminé ce didacticiel :

- Votre appareil IoT dispose d'un système d'exploitation à jour.
- Votre appareil IoT disposera du logiciel supplémentaire dont il a besoin pour les didacticiels suivants.
- Vous savez que votre appareil dispose d'une connectivité à Internet.
- Vous avez installé un certificat requis sur votre appareil.

Une fois que vous avez terminé ce tutoriel, le prochain tutoriel prépare votre appareil aux démonstrations qui utilisent le AWS IoTClient de l'appareil.

Procédures de ce didacticiel

- [Étape 1 : Installez et mettez à jour le système d'exploitation de l'appareil \(p. 146\)](#)
- [Étape 2 : Installez et vérifiez le logiciel requis sur votre appareil \(p. 149\)](#)
- [Étape 3 : Testez votre appareil et enregistrez le certificat Amazon CA \(p. 152\)](#)

Étape 1 : Installez et mettez à jour le système d'exploitation de l'appareil

Les procédures de cette section décrivent comment initialiser la carte microSD utilisée par le Raspberry Pi pour son lecteur système. La carte microSD du Raspberry Pi contient son logiciel de système d'exploitation (OS) ainsi que de l'espace pour le stockage de ses fichiers applicatifs. Si vous n'utilisez pas de Raspberry Pi, suivez les instructions de l'appareil pour installer et mettre à jour le logiciel du système d'exploitation de l'appareil.

Une fois cette section terminée, vous devriez pouvoir démarrer votre appareil IoT et vous y connecter à partir du programme Terminal sur votre ordinateur hôte local.

Equipement requis :

- Votre environnement local de développement et de test
- Un Raspberry Pi qui ou votre appareil IoT, qui peut se connecter à Internet
- Une carte mémoire microSD d'au moins 8 Go de capacité ou suffisamment de stockage pour le système d'exploitation et le logiciel requis.

Note

Lorsque vous sélectionnez une carte microSD pour ces exercices, choisissez-en une qui soit aussi grande que nécessaire mais aussi petite que possible.

Une petite carte SD sera plus rapide à sauvegarder et à mettre à jour. Sur le Raspberry Pi, vous n'aurez pas besoin de plus d'une carte microSD de 8 Go pour ces tutoriels. Si vous avez besoin de plus d'espace pour votre application spécifique, les petits fichiers image enregistrés dans ces didacticiels peuvent redimensionner le système de fichiers sur une carte plus grande pour utiliser tout l'espace pris en charge de la carte que vous choisissez.

Équipement en option :

- Un clavier USB connecté au Raspberry Pi
- Un moniteur HDMI et un câble pour connecter le moniteur au Raspberry Pi

Procédures décrites dans cette section :

- [Chargez le système d'exploitation de l'appareil sur une carte microSD \(p. 147\)](#)
- [Démarrez votre appareil IoT avec le nouveau système d'exploitation \(p. 148\)](#)
- [Connectez votre ordinateur hôte local à votre appareil \(p. 148\)](#)

Chargez le système d'exploitation de l'appareil sur une carte microSD

Cette procédure utilise l'ordinateur hôte local pour charger le système d'exploitation de l'appareil sur une carte microSD.

Note

Si votre appareil n'utilise pas de support de stockage amovible pour son système d'exploitation, installez le système d'exploitation à l'aide de la procédure applicable à cet appareil et continuez à la section ["Démarrez votre appareil IoT avec le nouveau système d'exploitation" \(p. 148\)](#).

Pour installer le système d'exploitation sur votre Raspberry Pi

1. Sur votre ordinateur hôte local, téléchargez et décompressez l'image du système d'exploitation Raspberry Pi que vous souhaitez utiliser. Les dernières versions sont disponibles sur <https://www.raspberrypi.com/software/operating-systems/>

Choix d'une version de Raspberry Pi OS

Ce didacticiel utilise .Raspberry Pi OS Lite car c'est la plus petite version qui prend en charge ces tutoriels dans ce parcours d'apprentissage. Cette version du système d'exploitation Raspberry Pi ne possède qu'une interface de ligne de commande et ne possède pas d'interface utilisateur graphique. Une version du dernier système d'exploitation Raspberry Pi dotée d'une interface utilisateur graphique fonctionnera également avec ces didacticiels. Cependant, les procédures décrites dans ce parcours d'apprentissage utilisent uniquement l'interface de ligne de commande pour effectuer des opérations sur le Raspberry Pi.

2. Insérez votre carte microSD dans l'ordinateur hôte local.
3. À l'aide d'un outil d'imagerie de carte SD, écrivez le fichier image du système d'exploitation décompressé sur la carte microSD.
4. Après avoir écrit l'image Raspberry Pi OS sur la carte microSD :
 - a. Ouvrez la partition BOOT sur la carte microSD dans une fenêtre de ligne de commande ou une fenêtre d'explorateur de fichiers.

- b. Dans la partition BOOT de la carte microSD, dans le répertoire racine, créez un fichier vide nommé ssh sans extension de fichier et sans contenu. Cela indique au Raspberry Pi d'activer les communications SSH dès le premier démarrage.
5. Éjectez la carte microSD et retirez-la de l'ordinateur hôte local en toute sécurité.

Votre carte microSD est prête à [the section called “Démarrez votre appareil IoT avec le nouveau système d’exploitation” \(p. 148\)](#).

Démarrez votre appareil IoT avec le nouveau système d’exploitation

Cette procédure installe la carte microSD et démarre votre Raspberry Pi pour la première fois à l'aide du système d'exploitation téléchargé.

Pour démarrer votre appareil IoT avec le nouveau système d’exploitation

1. Lorsque l'alimentation est déconnectée de l'appareil, insérez la carte microSD de l'étape précédente, [the section called “Chargez le système d’exploitation de l'appareil sur une carte microSD” \(p. 147\)](#), dans le Raspberry Pi.
2. Connect l'appareil à un réseau filaire.
3. Ces tutoriels interagiront avec votre Raspberry Pi depuis votre ordinateur hôte local à l'aide d'un terminal SSH.

Si vous souhaitez également interagir directement avec l'appareil, vous pouvez :

- a. Connectez-y un moniteur HDMI pour regarder les messages de la console du Raspberry Pi avant de pouvoir connecter la fenêtre du terminal de votre ordinateur hôte local à votre Raspberry Pi.
 - b. Connectez-y un clavier USB si vous souhaitez interagir directement avec le Raspberry Pi.
4. Connect l'alimentation au Raspberry Pi et attendez environ une minute qu'il s'initialise.
- Si un moniteur est connecté à votre Raspberry Pi, vous pouvez regarder le processus de démarrage.
5. Découvrez l'adresse IP de votre appareil :
 - Si vous avez connecté un moniteur HDMI au Raspberry Pi, l'adresse IP apparaît dans les messages affichés sur le moniteur
 - Si vous avez accès au routeur auquel votre Raspberry Pi se connecte, vous pouvez voir son adresse dans l'interface d'administration du routeur.

Une fois que vous avez l'adresse IP de votre Raspberry Pi, vous êtes prêt à [the section called “Connect votre ordinateur hôte local à votre appareil” \(p. 148\)](#).

Connect votre ordinateur hôte local à votre appareil

Cette procédure utilise le programme Terminal sur votre ordinateur hôte local pour se connecter à votre Raspberry Pi et modifier son mot de passe par défaut.

Pour connecter votre ordinateur hôte local à votre appareil

1. Sur votre ordinateur hôte local, ouvrez le programme terminal SSH :
 - Windows: PuTTY
 - Linux/macOS :Terminal

Note

PuTTY n'est pas installé automatiquement sous Windows. S'il n'est pas sur votre ordinateur, vous devrez peut-être le télécharger et l'installer.

2. Connectez le programme Terminal à l'adresse IP de votre Raspberry Pi et Connect à l'aide de ses informations d'identification par défaut.

```
username: pi  
password: raspberry
```

3. Une fois que vous vous êtes connecté à votre Raspberry Pi, modifiez le mot de passe du utilisateur.

```
passwd
```

Suivez les instructions pour modifier le mot de passe.

```
Changing password for pi.  
Current password: raspberry  
New password: YourNewPassword  
Retype new password: YourNewPassword  
passwd: password updated successfully
```

Une fois que vous avez installé l'invite de ligne de commande du Raspberry Pi dans la fenêtre du terminal et modifié le mot de passe, vous êtes prêt à continuer [the section called "Étape 2 : Installez et vérifiez le logiciel requis sur votre appareil" \(p. 149\)](#).

Étape 2 : Installez et vérifiez le logiciel requis sur votre appareil

Les procédures décrites dans cette section se poursuivent à partir de [la section précédente \(p. 146\)](#) pour mettre à jour le système d'exploitation de votre Raspberry Pi et installer le logiciel sur le Raspberry Pi qui sera utilisé dans la section suivante pour créer et installer le AWS IoTClient de l'appareil.

Une fois cette section terminée, votre Raspberry Pi dispose d'un système d'exploitation à jour, le logiciel requis par les didacticiels de ce parcours d'apprentissage, et il sera configuré pour votre localisation.

Equipement requis :

- Votre environnement de développement et de test local à partir de [la section précédente \(p. 146\)](#)
- Le Raspberry Pi que vous avez utilisé dans [la section précédente \(p. 146\)](#)
- La carte mémoire microSD de [la section précédente \(p. 146\)](#)

Note

Le Raspberry Pi Model 3+ et le Raspberry Pi Model 4 peuvent exécuter toutes les commandes décrites dans ce parcours d'apprentissage. Si votre appareil IoT ne peut pas compiler de logiciel ou exécuter le AWS Command Line Interface, vous devrez peut-être installer les compilateurs requis sur votre ordinateur hôte local pour créer le logiciel, puis le transférer sur votre appareil IoT. Pour plus d'informations sur l'installation et la création d'un logiciel pour votre appareil, consultez la documentation du logiciel de votre appareil.

Procédures décrites dans cette section :

- [Mise à jour du logiciel du système d'exploitation \(p. 150\)](#)

- [Installez les applications et bibliothèques requises \(p. 151\)](#)
- [\(Facultatif\) Enregistrez l'image de la carte microSD \(p. 151\)](#)

Mise à jour du logiciel du système d'exploitation

Cette procédure met à jour le logiciel du système d'exploitation.

Pour mettre à jour le logiciel du système d'exploitation sur le Raspberry Pi

Effectuez ces étapes dans la fenêtre du terminal de votre ordinateur hôte local.

1. Entrez ces commandes pour mettre à jour le logiciel système sur votre Raspberry Pi.

```
sudo apt-get -y update
sudo apt-get -y upgrade
sudo apt-get -y autoremove
```

2. Mettez à jour les paramètres régionaux et de fuseau horaire du Raspberry Pi (facultatif).

Entrez cette commande pour mettre à jour les paramètres régionaux et de fuseau horaire de l'appareil.

```
sudo raspi-config
```

- a. Pour définir les paramètres régionaux de l'appareil :

- i. Dans l'outil de configuration du logiciel Raspberry Pi (raspi-config) écran, choisissez une option 5.

5 Localisation Options Configure language and regional settings

Utilisation de l'abréviation Tabclé pour passer à <Select>, puis appuyez sur la barre d'espace.

- ii. Dans le menu des options de localisation, choisissez l'option L1.

L1 Locale Configure language and regional settings

Utilisation de l'abréviation Tabclé pour passer à <Select>, puis appuyez sur la barre d'espace.

- iii. Dans la liste des options de paramètres régionaux, choisissez les paramètres régionaux que vous souhaitez installer sur votre Raspberry Pi en utilisant les touches fléchées pour faire défiler et le bouton de la barre d'espace pour marquer ceux que vous voulez.

Aux États-Unis, **en_US.UTF-8** est un bon choix.

- iv. Après avoir sélectionné les paramètres régionaux de votre appareil, utilisez l'abréviation Tabclé pour choisir <OK>, puis appuyez sur la barre d'espace pour afficher la configuration des paramètres régionaux page de confirmation.

- b. Pour définir le fuseau horaire de l'appareil, procédez comme suit :

- i. Dans l'écran raspi-config, choisissez une option 5.

5 Localisation Options Configure language and regional settings

Utilisation de l'abréviation Tabclé pour passer à <Select>, puis appuyez sur la barre d'espace.

- ii. Dans le menu des options de localisation, utilisez la touche fléchée pour choisir l'option L2 :

L2 time zone Configure time zone

Utilisation de l'abréviation Tabclé pour passer à <Select>, puis appuyez sur la barre d'espace.

- iii. Dans Configuration de tzdata, choisissez votre zone géographique dans la liste.

- Utilisation de l'Tabclé pour passer à<OK>, puis appuyez surspace bar.
- iv. Dans la liste des villes, utilisez les touches fléchées pour choisir une ville dans votre fuseau horaire.

Pour définir le fuseau horaire, utilisez leTabclé pour passer à<OK>, puis appuyez surspace bar.
 - c. Lorsque vous avez terminé la mise à jour des paramètres, utilisez leTabclé pour passer à<Finish>, puis appuyez surspace bar pour fermer la raspi-configapp.
 3. Entrez cette commande pour redémarrer votre Raspberry Pi.

```
sudo shutdown -r 0
```

4. Attendez que votre Raspberry Pi redémarre.
5. Une fois votre Raspberry Pi redémarré, reconnectez la fenêtre du terminal de votre ordinateur hôte local à votre Raspberry Pi.

Votre logiciel système Raspberry Pi est maintenant configuré et vous êtes prêt à continuer[the section called "Installez les applications et bibliothèques requises" \(p. 151\)](#).

Installez les applications et bibliothèques requises

Cette procédure installe le logiciel d'application et les bibliothèques utilisés par les didacticiels suivants.

Si vous utilisez un Raspberry Pi ou si vous pouvez compiler le logiciel requis sur votre appareil IoT, effectuez ces étapes dans la fenêtre du terminal de votre ordinateur hôte local. Si vous devez compiler un logiciel pour votre appareil IoT sur votre ordinateur hôte local, consultez la documentation logicielle de votre appareil IoT pour obtenir des informations sur la procédure à suivre sur votre appareil.

Pour installer l'application, le logiciel et les bibliothèques sur votre Raspberry Pi

1. Entrez cette commande pour installer le logiciel d'application et les bibliothèques.

```
sudo apt-get -y install build-essential libssl-dev cmake unzip git python3-pip
```

2. Saisissez ces commandes pour confirmer que la version correcte du logiciel a été installée.

```
gcc --version  
cmake --version  
openssl version  
git --version
```

3. Vérifiez que ces versions du logiciel d'application sont installées :

- gcc: 9.3.0 ou version ultérieure
- cmake: 3.10.x ou version ultérieure
- OpenSSL: 1.1.1 ou version ultérieure
- git: 2.20.1 ou version ultérieure

Si votre Raspberry Pi dispose de versions acceptables du logiciel d'application requis, vous êtes prêt à continuer[the section called "\(Facultatif\) Enregistrez l'image de la carte microSD" \(p. 151\)](#).

(Facultatif) Enregistrez l'image de la carte microSD

Tout au long des didacticiels de ce parcours d'apprentissage, vous trouverez ces procédures pour enregistrer une copie de l'image de la carte microSD du Raspberry Pi dans un fichier de votre ordinateur

hôte local. Bien qu'ils soient encouragés, ils ne sont pas des tâches obligatoires. En enregistrant l'image de la carte microSD comme suggéré, vous pouvez ignorer les procédures précédant le point de sauvegarde dans ce parcours d'apprentissage, ce qui peut gagner du temps si vous avez besoin de réessayer quelque chose. La conséquence de l'absence d'enregistrement périodique de l'image de la carte microSD est que vous devrez peut-être redémarrer les didacticiels du parcours d'apprentissage dès le début si votre carte microSD est endommagée ou si vous configurez accidentellement une application ou ses paramètres incorrectement.

À ce stade, la carte microSD de votre Raspberry Pi est dotée d'un système d'exploitation mis à jour et du logiciel d'application de base chargé. Vous pouvez gagner du temps nécessaire pour effectuer les étapes précédentes en enregistrant le contenu de la carte microSD dans un fichier maintenant. L'image actuelle de l'image de la carte microSD de votre appareil vous permet de commencer à partir de ce moment pour continuer ou réessayer un tutoriel ou une procédure sans avoir besoin d'installer et de mettre à jour le logiciel à partir de zéro.

Pour enregistrer l'image de la carte microSD dans un fichier

1. Entrez cette commande pour arrêter le Raspberry Pi.

```
sudo shutdown -h 0
```

2. Une fois que le Raspberry Pi s'est complètement arrêté, retirez son alimentation.
3. Retirez la carte microSD du Raspberry Pi.
4. Sur votre ordinateur hôte local :
 - a. Insérez la carte microSD.
 - b. À l'aide de l'outil d'imagerie de votre carte SD, enregistrez l'image de la carte microSD dans un fichier.
 - c. Une fois l'image de la carte microSD enregistrée, éjectez-la de l'ordinateur hôte local.
5. Lorsque l'alimentation est déconnectée du Raspberry Pi, insérez la carte microSD dans le Raspberry Pi.
6. Appliquez de l'énergie au Raspberry Pi.
7. Après avoir attendu environ une minute, sur l'ordinateur hôte local, reconnectez la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi., puis connectez-vous au Raspberry Pi.

Étape 3 : Testez votre appareil et enregistrez le certificat Amazon CA

Les procédures décrites dans cette section se poursuivent à partir de la section précédente (p. 149) pour installer le kit AWS Command Line Interface et le certificat de l'autorité de certification utilisé pour authentifier vos connexions avec AWS IoT Core.

Une fois cette section terminée, vous saurez que votre Raspberry Pi dispose du logiciel système nécessaire pour installer le AWS IoT Device Client et qu'il dispose d'une connexion fonctionnelle à Internet.

Equipement requis :

- Votre environnement de développement et de test local à partir de la section précédente (p. 149)
- Le Raspberry Pi que vous avez utilisé dans la section précédente (p. 149)
- La carte mémoire microSD de la section précédente (p. 149)

Procédures décrites dans cette section :

- [Installation de l'AWS Command Line Interface \(p. 153\)](#)
- [Configuration de vos informations d'identification pour l'Compte AWS \(p. 153\)](#)
- [Télécharger le certificat de l'autorité de certification racine Amazon \(p. 154\)](#)
- [\(Facultatif\) Enregistrez l'image de la carte microSD \(p. 155\)](#)

Installation de l'AWS Command Line Interface

Cette procédure installe le AWS CLI sur votre Raspberry Pi.

Si vous utilisez un Raspberry Pi ou si vous pouvez compiler des logiciels sur votre appareil IoT, effectuez ces étapes dans la fenêtre du terminal de votre ordinateur hôte local. Si vous devez compiler un logiciel pour votre appareil IoT sur votre ordinateur hôte local, consultez la documentation logicielle de votre appareil IoT pour obtenir des informations sur les bibliothèques dont il a besoin.

Pour installer AWS CLI sur votre Raspberry Pi

1. Pour télécharger et installer l'appareil, exécutez les commandes suivantes : AWS CLI.

```
export PATH=$PATH:~/local/bin # configures the path to include the directory with the
AWS CLI
git clone https://github.com/aws/aws-cli.git # download the AWS CLI code from GitHub
cd aws-cli && git checkout v2 # go to the directory with the repo and checkout version
2
pip3 install -r requirements.txt # install the prerequisite software
```

2. Pour installer l'appareil, exécutez la commande suivante : AWS CLI. Cette commande peut prendre jusqu'à 15 minutes.

```
pip3 install . # install the AWS CLI
```

3. Pour confirmer que la version correcte de l'appareil AWS CLI a été installé.

```
aws --version
```

La version de AWS CLI doit être de 2.2 ou ultérieure.

Si l'icône AWS CLI affiche sa version actuelle, vous êtes prêt à continuer à [the section called “Configuration de vos informations d'identification pour l'Compte AWS” \(p. 153\)](#).

Configuration de vos informations d'identification pour l'Compte AWS

Dans cette procédure, vous obtiendrez Compte AWS et ajoutez-les pour les utiliser sur votre Raspberry Pi.

Pour ajouter votre Compte AWS, informations d'identification de votre appareil

1. Obtenir un ID de clé d'accès et une clé d'accès secrète de votre Compte AWS pour authentifier le AWS CLI sur votre appareil.

Si le kit ne vous est pas destiné AWS IAM, <https://aws.amazon.com/premiumsupport/knowledge-center/create-access-key/> décrit le processus à exécuter dans le AWS console à créer AWS Informations d'identification IAM à utiliser sur votre appareil.

2. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, ID de clé d'accès et clé d'accès secrète informations d'identification de votre appareil :

- a. Exécutez le AWS Configurez l'application avec cette commande :

aws configure

- b. Saisissez vos informations d'identification et de configuration lorsque vous y êtes invité :

```
AWS Access Key ID: your Access Key ID
AWS Secret Access Key: your Secret Access Key
Default region name: your Region AWS code
Default output format: json
```

3. Exécutez cette commande pour tester l'accès de votre appareil à votre Compte AWS IoT Core.

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

Il devrait renvoyer votre Compte AWS-spécifiques pour AWS IoT point de terminaison de données, tel que cet exemple :

```
{  
    "endpointAddress": "a3EXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

Si vous voyez votre Compte AWS-spécifiques pour AWS IoT point de terminaison de données, votre Raspberry Pi dispose de la connectivité et des autorisations nécessaires pour continuer [the section called "Télécharger le certificat de l'autorité de certification racine Amazon" \(p. 154\)](#).

Important

Votre Compte AWS les informations d'identification sont désormais stockées sur la carte microSD de votre Raspberry Pi. Bien que cela crée des interactions futures avec AWS faciles pour vous et le logiciel que vous allez créer dans ces tutoriels, ils seront également enregistrés et dupliqués dans toutes les images de carte microSD que vous créez après cette étape par défaut.

Pour protéger la sécurité de votre Compte AWS avant d'enregistrer d'autres images de carte microSD, envisagez d'effacer les informations d'identification en exécutant `aws configure` et en saisissant des caractères aléatoires pour le ID de clé d'accès et Clé d'accès secrète pour empêcher votre Compte AWS informations d'identification de compromises.

Si vous constatez que vous avez enregistré votre Compte AWS par inadvertance, vous pouvez les désactiver dans la AWS Console IAM.

Télécharger le certificat de l'autorité de certification racine Amazon

Cette procédure permet de télécharger et d'enregistrer une copie d'un certificat de l'autorité de certification racine Amazon (CA). Le téléchargement de ce certificat permet de l'enregistrer pour l'utiliser dans les didacticiels suivants et il teste également la connectivité de votre appareil avec AWS Services.

Pour télécharger et enregistrer le certificat de l'autorité de certification racine Amazon

1. Pour créer un répertoire pour le certificat, exécutez la commande suivante.

mkdir ~/certs

2. Pour télécharger le certificat de l'autorité de certification racine Amazon, exécutez la commande suivante.

```
curl -o ~/certs/AmazonRootCA1.pem https://www.amazontrust.com/repository/
AmazonRootCA1.pem
```

3. Exécutez ces commandes pour définir l'accès au répertoire de certificats et à son fichier.

```
chmod 745 ~  
chmod 700 ~/certs  
chmod 644 ~/certs/AmazonRootCA1.pem
```

4. Pour afficher le fichier du certificat de l'autorité de certification, exécutez la commande suivante dans le nouveau répertoire.

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

Elle doit se présenter comme suit. La date et l'heure seront différentes, mais la taille du fichier et toutes les autres informations doivent être identiques à celles indiquées ici.

```
-rw-r--r-- 1 pi pi 1188 Oct 28 13:02 AmazonRootCA1.pem
```

Si la taille du fichier n'est pas 1188, vérifiez la section curlparamètres de commande. Vous avez peut-être téléchargé un fichier incorrect.

(Facultatif) Enregistrez l'image de la carte microSD

À ce stade, la carte microSD de votre Raspberry Pi est dotée d'un système d'exploitation mis à jour et du logiciel d'application de base chargé.

Pour enregistrer l'image de la carte microSD dans un fichier

1. Dans la fenêtre du terminal de votre ordinateur hôte local, effacez votre AWS Informations d'identification .

- a. Exécutez le AWS Configurez l'application avec cette commande :

```
aws configure
```

- b. Remplacez vos informations d'identification lorsque vous y êtes invité. Vous pouvez partir Nom de la région par défaut et Format de sortie par défaut comme ils le sont en appuyant sur Saisissez.

```
AWS Access Key ID [*****YT2H]: XYXYXYXYX  
AWS Secret Access Key [*****9p1H]: XYXYXYXYX  
Default region name [us-west-2]:  
Default output format [json]:
```

2. Entrez cette commande pour arrêter le Raspberry Pi.

```
sudo shutdown -h 0
```

3. Une fois que le Raspberry Pi s'est complètement arrêté, retirez son connecteur d'alimentation.
4. Retirez la carte microSD de votre appareil.
5. Sur votre ordinateur hôte local :

- a. Insérez la carte microSD.
 - b. À l'aide de l'outil d'imagerie de votre carte SD, enregistrez l'image de la carte microSD dans un fichier.
 - c. Une fois l'image de la carte microSD enregistrée, éjectez-la de l'ordinateur hôte local.
6. Lorsque l'alimentation est déconnectée du Raspberry Pi, insérez la carte microSD dans le Raspberry Pi.
 7. Appliquez de l'alimentation à l'appareil.

8. Après environ une minute, sur l'ordinateur hôte local, redémarrez la session de fenêtre du terminal et connectez-vous à l'appareil.

Ne rentrez pas dans votreCompte AWSinformations d'identification encore.

Après avoir redémarré et connecté à votre Raspberry Pi, vous êtes prêt à continuer[the section called "Installation et configuration de l'AWS IoTClient d'appareil" \(p. 156\)](#).

Didacticiel : Installation et configuration de l'AWS IoTClient d'appareil

Ce didacticiel explique l'installation et la configuration duAWS IoTDevice Client et création deAWS IoTressources que vous utiliserez dans cette démonstration et dans d'autres démos.

Pour démarrer ce didacticiel :

- Ayez votre ordinateur hôte local et votre Raspberry Pi depuis[Didacticiel précédent \(p. 146\)](#)prêt.

Ce didacticiel dure jusqu'à 90 minutes.

Lorsque vous avez terminé avec ce sujet :

- Votre appareil IoT sera prêt à être utilisé dans d'autresAWS IoTDemos Device Client.
- Vous aurez provisionné votre appareil IoT dansAWS IoT Core.
- Vous aurez téléchargé et installé leAWS IoTDevice Client sur votre appareil.
- Vous aurez enregistré une image de la carte microSD de votre appareil qui pourra être utilisée dans les didacticiels suivants.

Equipement requis :

- Votre environnement de développement et de test local à partir de[la section précédente \(p. 152\)](#)
- Le Raspberry Pi que vous avez utilisé dans[la section précédente \(p. 152\)](#)
- La carte mémoire microSD du Raspberry Pi que vous avez utilisée dans[la section précédente \(p. 152\)](#)

Procédures de ce didacticiel

- [Étape 1 : Téléchargez et enregistrez leAWS IoTClient d'appareil \(p. 156\)](#)
- [\(Facultatif\) Enregistrez l'image de la carte microSD \(p. 158\)](#)
- [Étape 2 : Provisionnez votre Raspberry PiAWS IoT \(p. 158\)](#)
- [Étape 3 : Configurer l'AWS IoTDevice Client pour tester la connectivité \(p. 162\)](#)

Étape 1 : Téléchargez et enregistrez leAWS IoTClient d'appareil

Les procédures de cette section téléchargent leAWS IoTDevice Client, compilez-le et installez-le sur votre Raspberry Pi. Après avoir testé l'installation, vous pouvez enregistrer l'image de la carte microSD du Raspberry Pi pour l'utiliser ultérieurement lorsque vous souhaitez réessayer les didacticiels.

Procédures décrites dans cette section :

- [Téléchargez et créez leAWS IoTClient d'appareil \(p. 157\)](#)
- [Créer les répertoires utilisés par les didacticiels \(p. 157\)](#)

Téléchargez et créez leAWS IoTClient d'appareil

Cette procédure installe leAWS IoTDevice Client sur votre Raspberry Pi.

Exécutez ces commandes dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi.

Pour installerAWS IoTDevice Client sur votre Raspberry Pi

1. Entrez ces commandes pour télécharger et générer leAWS IoTDevice Client sur votre Raspberry Pi.

```
cd ~  
git clone https://github.com/awslabs/aws-iot-device-client aws-iot-device-client  
mkdir ~/aws-iot-device-client/build && cd ~/aws-iot-device-client/build  
cmake ../
```

2. Exécutez cette commande pour générer leAWS IoTClient de l'appareil. Cette commande peut prendre jusqu'à 15 minutes.

```
cmake --build . --target aws-iot-device-client
```

Les messages d'avertissement affichés en tant queAWS IoTLes compilations de Device Client peuvent être ignorées.

Ces tutoriels ont été testés avec leAWS IoTDevice Client basé sur gcc, version (Raspbian 10.2.1-6+rpi1) 10.2.1 20210110 sur la version 30 octobre 2021 de Raspberry Pi OS (bullseye) sur gcc, version (Raspbian 8.3.0-6+rpi1) 8.3.0 sur la version du 7 mai 2021 du système d'exploitation Raspberry Pi (buster).

3. After theAWS IoTDevice Client termine la construction, testez-le en exécutant cette commande.

```
./aws-iot-device-client --help
```

Si vous voyez l'aide de la ligne de commande pour laAWS IoTDevice Client, leAWS IoTDevice Client a été créé avec succès et est prêt à être utilisé.

Créer les répertoires utilisés par les didacticiels

Cette procédure crée les répertoires du Raspberry Pi qui seront utilisés pour stocker les fichiers utilisés par les didacticiels de ce parcours d'apprentissage.

Pour créer les répertoires utilisés par les didacticiels de ce parcours d'apprentissage :

1. Exécutez ces commandes pour créer les répertoires requis.

```
mkdir ~/dc-configs  
mkdir ~/policies  
mkdir ~/messages  
mkdir ~/certs/testconn  
mkdir ~/certs/pubsub  
mkdir ~/certs/jobs
```

2. Exécutez ces commandes pour définir les autorisations sur les nouveaux répertoires.

```
chmod 745 ~  
chmod 700 ~/certs/testconn  
chmod 700 ~/certs/pubsub
```

```
chmod 700 ~/certs/jobs
```

Après avoir créé ces répertoires et défini leur autorisation, continuez à [the section called “\(Facultatif\) Enregistrez l'image de la carte microSD” \(p. 158\)](#).

(Facultatif) Enregistrez l'image de la carte microSD

À ce stade, la carte microSD de votre Raspberry Pi est dotée d'un système d'exploitation mis à jour, du logiciel d'application de base et du AWS IoTClient de l'appareil.

Si vous souhaitez revenir essayer à nouveau ces exercices et ces tutoriels, vous pouvez ignorer les procédures précédentes en écrivant l'image de la carte microSD que vous enregistrez avec cette procédure sur une nouvelle carte microSD et continuez les tutoriels de [the section called “Étape 2 : Provisionnez votre Raspberry PiAWS IoT” \(p. 158\)](#).

Pour enregistrer l'image de la carte microSD dans un fichier :

Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi :

1. Confirmer que votreCompte AWSles informations d'identification n'ont pas été stockées.

- a. Exécutez leAWSconfigurez l'application avec cette commande :

```
aws configure
```

- b. Si vos informations d'identification ont été stockées (si elles sont affichées dans l'invite), entrez le champ **XYYXXYXXYX**. Lorsque vous y êtes invité, comme indiqué ici. PartezNom de la région par défaut etFormat de sortie par défautvide.

```
AWS Access Key ID [*****YXYX]: XYYXXYXXYX
AWS Secret Access Key [*****YXYX]: XYYXXYXXYX
Default region name:
Default output format:
```

2. Entrez cette commande pour arrêter le Raspberry Pi.

```
sudo shutdown -h 0
```

3. Une fois que le Raspberry Pi s'est complètement arrêté, retirez son connecteur d'alimentation.
4. Retirez la carte microSD de votre appareil.
5. Sur votre ordinateur hôte local :

- a. Insérez la carte microSD.
- b. À l'aide de l'outil d'imagerie de votre carte SD, enregistrez l'image de la carte microSD dans un fichier.
- c. Une fois l'image de la carte microSD enregistrée, éjectez-la de l'ordinateur hôte local.

Vous pouvez continuer avec cette carte microSD dans [the section called “Étape 2 : Provisionnez votre Raspberry PiAWS IoT” \(p. 158\)](#).

Étape 2 : Provisionnez votre Raspberry PiAWS IoT

Les procédures de cette section commencent par l'image microSD enregistrée qui possède le paramètre AWS CLIAWS IoTDevice Client a installé et créé leAWS IoTressources et certificats d'appareils qui provisionnent votre Raspberry Pi dansAWS IoT.

Installez la carte microSD dans votre Raspberry Pi

Cette procédure installe la carte microSD avec le logiciel nécessaire chargé et configuré dans le Raspberry Pi et configure votreCompte AWSafin que vous puissiez continuer avec les didacticiels de ce parcours d'apprentissage.

Utiliser une carte microSD de[the section called “\(Facultatif\) Enregistrez l'image de la carte microSD” \(p. 158\)](#)qui dispose du logiciel nécessaire pour les exercices et les tutoriels de ce parcours d'apprentissage.

Pour installer la carte microSD dans votre Raspberry Pi

1. Lorsque l'alimentation est déconnectée du Raspberry Pi, insérez la carte microSD dans le Raspberry Pi.
2. Appliquez de l'énergie au Raspberry Pi.
3. Après environ une minute, sur l'ordinateur hôte local, redémarrez la session de fenêtre du terminal et connectez-vous au Raspberry Pi.
4. Sur votre ordinateur hôte local, dans la fenêtre du terminal, et avec leID de clé d'accèssetClé d'accès secrèteinformations d'identification de votre Raspberry Pi :
 - a. Exécutez leAWSconfigurez l'application avec cette commande :

```
aws configure
```

- b. Saisissez vosCompte AWSinformations d'identification et de configuration lorsque vous y êtes invité :

```
AWS Access Key ID [*****YXYX]: your Access Key ID
AWS Secret Access Key [*****YXYX]: your Secret Access Key
Default region name [us-west-2]: your Région AWS code
Default output format [json]: json
```

Une fois que vous avez restauré votreCompte AWSvos informations d'identification, vous êtes prêt à continuer[the section called “Provisionnez votre appareil dansAWS IoT Core” \(p. 159\)](#).

Provisionnez votre appareil dansAWS IoT Core

Les procédures de cette section créent leAWS IoTressources qui approvisionnent votre Raspberry Pi dansAWS IoT. Lorsque vous créez ces ressources, il vous sera demandé d'enregistrer diverses informations. Ces informations sont utilisées par leAWS IoTConfiguration de Device Client dans la procédure suivante.

Pour que votre Raspberry Pi fonctionne avecAWS IoT, il doit être provisionné. Le provisioning est le processus de création et de configuration de laAWS IoTressources nécessaires à la prise en charge de votre Raspberry Pi en tant qu'appareil IoT.

Une fois votre Raspberry Pi sous tension et redémarré, connectez la fenêtre du terminal de votre ordinateur hôte local au Raspberry Pi et effectuez ces procédures.

Procédures décrites dans cette section :

- [Créer et télécharger des fichiers de certificats d'appareil \(p. 159\)](#)
- [CréerAWS IoTressources \(p. 160\)](#)

Créer et télécharger des fichiers de certificats d'appareil

Cette procédure crée les fichiers de certificat de périphérique pour cette démonstration.

Pour créer et télécharger les fichiers de certificat d'appareil pour votre Raspberry Pi

1. Dans la fenêtre du terminal de votre ordinateur hôte local, entrez ces commandes pour créer les fichiers de certificat de périphérique pour votre appareil.

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

La commande renvoie une réponse telle que la suivante. Enregistrez le *certificateArn* valeur en vue d'une utilisation ultérieure.

```
{
    "certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",
    "certificateId":
    "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",
    "certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
    "keyPair": {
        "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBqkqhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
        "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
    }
}
```

2. Entrez les commandes suivantes pour définir les autorisations sur le répertoire de certificats et ses fichiers.

```
chmod 745 ~
chmod 700 ~/certs/testconn
chmod 644 ~/certs/testconn/*
chmod 600 ~/certs/testconn/private.pem.key
```

3. Exécutez cette commande pour vérifier les autorisations sur vos répertoires et fichiers de certificats.

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

La sortie de la commande doit être la même que celle que vous voyez ici, sauf que les dates et heures du fichier seront différentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
-rw----- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi 451 Oct 28 13:02 public.pem.key
```

À ce stade, les fichiers de certificat de l'appareil sont installés sur votre Raspberry Pi et vous pouvez continuer à [the section called “Créer AWS IoTressources” \(p. 160\)](#).

Créer AWS IoTressources

Cette procédure permet à votre appareil de AWS IoT en créant les ressources auxquelles votre appareil a besoin d'accéder AWS IoT fonctionnalités et services.

Pour provisionner votre appareil dans AWS IoT

1. Dans la fenêtre du terminal de votre ordinateur hôte local, entrez la commande suivante pour obtenir l'adresse du point de terminaison de données du périphérique pour votre Compte AWS.

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

La commande des étapes précédentes renvoie une réponse telle que la suivante. Enregistrez le **endpointAddress** valeur en vue d'une utilisation ultérieure.

```
{  
    "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Saisissez cette commande pour créer un AWS IoT ressource objet pour votre Raspberry Pi.

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

Si vos recettes AWS IoT ressource a été créée, la commande renvoie une réponse telle que celle-ci.

```
{  
    "thingName": "DevCliTestThing",  
    "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/DevCliTestThing",  
    "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. Dans la fenêtre du terminal :

- Ouvrez un éditeur de texte, tel qu'enano.
- Copiez ce document de stratégie JSON et collez-le dans votre éditeur de texte ouvert.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish",  
                "iot:Subscribe",  
                "iot:Receive",  
                "iot:Connect"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

Note

Ce document de stratégie accorde généreusement à chaque ressource l'autorisation de se connecter, de recevoir, de publier et de s'abonner. Normalement, les stratégies autorisent uniquement des ressources spécifiques pour leur permettre d'effectuer des actions spécifiques. Toutefois, pour le test initial de connectivité des appareils, cette stratégie trop générale et permissive est utilisée pour minimiser le risque de problème d'accès pendant ce test. Dans les didacticiels suivants, des documents de politique plus

étroits seront utilisés pour démontrer les meilleures pratiques en matière de conception de politiques.

- c. Enregistrez le fichier dans votre éditeur de texte sous~/**policies/dev_cli_test_thing_policy.json**.
4. Exécutez cette commande pour utiliser le document de stratégie des étapes précédentes pour créer unAWS IoTpolitique.

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

Si la stratégie est créée, la commande renvoie une réponse telle que celle-ci.

```
{  
    "policyName": "DevCliTestThingPolicy",  
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/DevCliTestThingPolicy",  
    "policyDocument": "{\n        \"Version\": \"2012-10-17\", \"Statement\": [\n            {\n                \"Effect\": \"Allow\", \"Action\": \"[\n                    \"iot:Publish\", \"iot:Subscribe\", \"iot:Receive\"\n                ]\", \"Resource\": \"[\n                    \"*\"\n                ]\"}\n            ]\n        },\n        \"policyVersionId\": \"1\"\n    }"
```

5. Exécutez cette commande pour attacher la stratégie au certificat de l'appareil. Remplacez**certificateArn**avec le**certificateArn**valeur que vous avez enregistrée précédemment.

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

Si elle aboutit, cette commande ne renvoie rien.

6. Exécutez cette commande pour attacher le certificat de périphérique à laAWS IoTressource d'objet. Remplacez**certificateArn**avec le**certificateArn**valeur que vous avez enregistrée précédemment.

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

Si elle aboutit, cette commande ne renvoie rien.

Une fois que vous avez correctement provisionné votre appareil dansAWS IoT, vous êtes prêt à continuer[the section called “Étape 3 : Configurer l'AWS IoTDevice Client pour tester la connectivité” \(p. 162\)](#).

Étape 3 : Configurer l'AWS IoTDevice Client pour tester la connectivité

Les procédures de cette section permettent de configurer leAWS IoTDevice Client pour publier un message MQTT à partir de votre Raspberry Pi.

Procédures décrites dans cette section :

- [Créer le fichier de configuration \(p. 163\)](#)

- [Client de test MQTT ouvert \(p. 164\)](#)
- [Run \(Exécuter Lambda\)AWS IoTClient d'appareil \(p. 164\)](#)

Créer le fichier de configuration

Cette procédure crée le fichier de configuration pour tester leAWS IoTClient de l'appareil.

Pour créer le fichier de configuration afin de tester leAWS IoTClient d'appareil

- Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi :
 - Entrez ces commandes pour créer un répertoire pour les fichiers de configuration et définir l'autorisation sur le répertoire :

```
mkdir ~/dc-configs
chmod 745 ~/dc-configs
```

- Ouvrez un éditeur de texte, tel qu'enano.
- Copiez ce document JSON et collez-le dans votre éditeur de texte ouvert.

```
{
  "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
  "cert": "~/certs/testconn/device.pem.crt",
  "key": "~/certs/testconn/private.pem.key",
  "root-ca": "~/certs/AmazonRootCA1.pem",
  "thing-name": "DevCliTestThing",
  "logging": {
    "enable-sdk-logging": true,
    "level": "DEBUG",
    "type": "STDOUT",
    "file": ""
  },
  "jobs": {
    "enabled": false,
    "handler-directory": ""
  },
  "tunneling": {
    "enabled": false
  },
  "device-defender": {
    "enabled": false,
    "interval": 300
  },
  "fleet-provisioning": {
    "enabled": false,
    "template-name": "",
    "template-parameters": "",
    "csr-file": "",
    "device-key": ""
  },
  "samples": {
    "pub-sub": {
      "enabled": true,
      "publish-topic": "test/dc/pubtopic",
      "publish-file": "",
      "subscribe-topic": "test/dc/subtopic",
      "subscribe-file": ""
    }
  },
  "config-shadow": {
    "enabled": false
  }
},
```

```
"sample-shadow": {  
    "enabled": false,  
    "shadow-name": "",  
    "shadow-input-file": "",  
    "shadow-output-file": ""  
}
```

- d. Remplacez le *point final* valeur avec le point de terminaison de données de périphérique pour votre Compte AWS que vous avez trouvé dans [the section called “Provisionnez votre appareil dans AWS IoT Core” \(p. 159\)](#).
- e. Enregistrez le fichier dans votre éditeur de texte sous **~/dc-configs/dc-testconn-config.json**.
- f. Exécutez cette commande pour définir les autorisations sur le nouveau fichier de configuration.

```
chmod 644 ~/dc-configs/dc-testconn-config.json
```

Lorsque vous avez enregistré le fichier, vous êtes prêt à continuer [the section called “Client de test MQTT ouvert” \(p. 164\)](#).

Client de test MQTT ouvert

Cette procédure prépare le Client MQTT dans le AWS IoT pour s'abonner au message MQTT indiquant que AWS IoT Device Client publie lorsqu'il s'exécute.

Pour préparer le Client MQTT pour vous abonner à tous les messages MQTT

1. Sur votre ordinateur hôte local, dans le [AWS IoT console](#), choisissez Client MQTT.
2. Dans S'abonner à une rubrique onglet, dans Filtre de rubriques, entrez # (un signe de livre unique), et choisissez S'abonner pour vous abonner à tous les sujets MQTT.
3. Au-dessous du Subscriptions étiquette, confirmez que vous voyez # (un signe de livre unique).

Laissez la fenêtre avec le Client MQTT ouvert pendant que vous continuez à [the section called “Run \(Exécuter Lambda\) AWS IoTClient d'appareil” \(p. 164\)](#).

Run (Exécuter Lambda) AWS IoTClient d'appareil

Cette procédure exécute le AWS IoT Device Client afin qu'il publie un seul message MQTT que le Client MQTT reçoit et affiche.

Pour envoyer un message MQTT depuis le AWS IoTClient d'appareil

1. Assurez-vous que la fenêtre du terminal connectée à votre Raspberry Pi et la fenêtre avec le Client MQTT sont visibles pendant que vous effectuez cette procédure.
2. Dans la fenêtre du terminal, entrez ces commandes pour exécuter le AWS IoT Device Client utilisant le fichier de configuration créé dans [the section called “Créer le fichier de configuration” \(p. 163\)](#).

```
cd ~/aws-iot-device-client/build  
./aws-iot-device-client --config-file ~/dc-configs/dc-testconn-config.json
```

Dans la fenêtre de terminal, AWS IoT Device Client affiche les messages d'information et les erreurs éventuelles qui se produisent lors de son exécution.

Si aucune erreur n'est affichée dans la fenêtre du terminal, consultez la Client MQTT.

3. DansClient MQTT, dans la fenêtre Abonnements, reportez-vous auHello World !message envoyé autest/dc/pubtopicsujet de message.
4. Si l'icôneAWS IoTDevice Client n'affiche aucune erreur et vous voyezHello World !envoyé autest/dc/pubtopicmessage dans leClient MQTT, vous avez démontré une connexion réussie.
5. Dans la fenêtre de terminal, entrez^C(Ctrl+C) pour arrêter leAWS IoTClient de l'appareil.

Une fois que vous avez démontré que leAWS IoTDevice Client fonctionne correctement sur votre Raspberry Pi et peut communiquer avecAWS IoT, vous pouvez passer à[the section called “Démontrer la communication des messages MQTT avec leAWS IoTAppareil client” \(p. 165\)](#).

Didacticiel : Démontrer la communication des messages MQTT avec leAWS IoTAppareil client

Ce didacticiel explique comment leAWS IoTDevice Client peut s'abonner et publier des messages MQTT, couramment utilisés dans les solutions IoT.

Pour démarrer ce didacticiel :

- Configurez votre ordinateur hôte local et votre Raspberry Pi comme utilisé dans[la section précédente \(p. 156\)](#).

Si vous avez enregistré l'image de la carte microSD après avoir installé leAWS IoTDevice Client, vous pouvez utiliser une carte microSD avec cette image avec votre Raspberry Pi.

- Si vous avez déjà exécuté cette démonstration, consultez[??? \(p. 190\)](#)pour tout supprimerAWS IoTressources que vous avez créées lors d'exécutions antérieures pour éviter les erreurs de ressources en double.

Ce didacticiel vous prendra environ 45 minutes.

Lorsque vous avez terminé avec ce sujet :

- Vous aurez démontré différentes façons dont votre appareil IoT peut s'abonner aux messages MQTT depuisAWS IoTet publiez des messages MQTT surAWS IoT.

Equipement requis :

- Votre environnement de développement et de test local à partir de[la section précédente \(p. 156\)](#)
- Le Raspberry Pi que vous avez utilisé dans[la section précédente \(p. 156\)](#)
- La carte mémoire microSD du Raspberry Pi que vous avez utilisée dans[la section précédente \(p. 156\)](#)

Procédures de ce didacticiel

- [Étape 1 : Préparer Raspberry Pi pour montrer la communication des messages MQTT \(p. 165\)](#)
- [Étape 2 : Démontrez la publication de messages avec leAWS IoTAppareil client \(p. 171\)](#)
- [Étape 3 : Démontrer que vous vous abonnez à des messages avec leAWS IoTAppareil client \(p. 173\)](#)

Étape 1 : Préparer Raspberry Pi pour montrer la communication des messages MQTT

Cette procédure crée les ressources dansAWS IoTet dans le Raspberry Pi pour démontrer la communication des messages MQTT à l'aide duAWS IoTClient de l'appareil.

Procédures de cette section :

- [Créez les fichiers de certificats pour démontrer la communication MQTT \(p. 166\)](#)
- [Provisionnez votre appareil pour démontrer la communication MQTT \(p. 167\)](#)
- [Configurer l'AWS IoTFichier de configuration Device Client et client de test MQTT pour démontrer la communication MQTT \(p. 169\)](#)

Créez les fichiers de certificats pour démontrer la communication MQTT

Cette procédure crée les fichiers de certificat de périphérique pour cette démonstration.

Pour créer et télécharger les fichiers de certificat d'appareil pour votre Raspberry Pi

1. Dans la fenêtre du terminal de votre ordinateur hôte local, entrez la commande suivante pour créer les fichiers de certificat de périphérique pour votre appareil.

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

La commande renvoie une réponse telle que la suivante : Enregistrer le*certificateArn*valeur pour une utilisation ultérieure.

```
{
"certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",
"certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",
"certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfgtS\n-----END CERTIFICATE-----\n",
"keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBkgqhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
}
}
```

2. Entrez les commandes suivantes pour définir les autorisations sur le répertoire de certificats et ses fichiers.

```
chmod 700 ~/certs/pubsub
chmod 644 ~/certs/pubsub/*
chmod 600 ~/certs/pubsub/private.pem.key
```

3. Exécutez cette commande pour vérifier les autorisations sur vos répertoires et fichiers de certificats.

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

La sortie de la commande doit être la même que celle que vous voyez ici, sauf que les dates et heures du fichier seront différentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
-rw----- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi 451 Oct 28 13:02 public.pem.key
```

4. Entrez ces commandes pour créer les répertoires des fichiers journaux.

```
mkdir ~/.aws-iot-device-client
mkdir ~/.aws-iot-device-client/log
chmod 745 ~/.aws-iot-device-client/log
echo " " > ~/.aws-iot-device-client/log/aws-iot-device-client.log
echo " " > ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
chmod 600 ~/.aws-iot-device-client/log/*
```

Provisionnez votre appareil pour démontrer la communication MQTT

Cette section crée leAWS IoTressources qui approvisionnent votre Raspberry Pi dansAWS IoT.

Pour provisionner votre appareil dansAWS IoT :

1. Dans la fenêtre du terminal de votre ordinateur hôte local, entrez la commande suivante pour obtenir l'adresse du point de terminaison de données du périphérique pour votreCompte AWS.

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

La valeur du point de terminaison n'a pas changé depuis que vous avez exécuté cette commande pour le didacticiel précédent. L'exécution de la commande ici est effectuée pour faciliter la recherche et la collage de la valeur du point de terminaison de données dans le fichier de configuration utilisé dans ce didacticiel.

La commande des étapes précédentes renvoie une réponse telle que la suivante : Enregistrez le`endpointAddress`valeur pour une utilisation ultérieure.

```
{  
  "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Entrez cette commande pour créer un nouveauAWS IoTressource pour votre Raspberry Pi.

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

Etant donné queAWS IoTchose ressource est unvirtuelreprésentation de votre appareil dans le cloud, nous pouvons créer plusieurs ressources d'objets dansAWS IoTUtiliser à différentes fins. Ils peuvent tous être utilisés par le même périphérique IoT physique pour représenter différents aspects de l'appareil.

Ces tutoriels n'utiliseront qu'une ressource à la fois pour représenter le Raspberry Pi. De cette façon, dans ces tutoriels, ils représentent les différentes démos, de sorte qu'après avoir créé leAWS IoTressources pour une démonstration, vous pouvez revenir en arrière et répéter la démo en utilisant les ressources que vous avez créées spécifiquement pour chacune d'elles.

Si vos recettesAWS IoTLa ressource objet a été créée, la commande renvoie une réponse telle que celle-ci.

```
{  
  "thingName": "PubSubTestThing",  
  "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/PubSubTestThing",  
  "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. Dans la fenêtre du terminal :

- a. Ouvrez un éditeur de texte, tel quenano.
- b. Copiez ce document JSON et collez-le dans votre éditeur de texte ouvert.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic"  
            ]  
        }  
    ]  
}
```

- c. Dans l'éditeur, dans chaqueResource du document de politique, remplacer**us-west-2:57 EXAMPLE 833**avec vosRégion AWS, un caractère deux-points (:) et vos 12 chiffresCompte AWSnombre.
 - d. Enregistrez le fichier dans votre éditeur de texte sous~/**policies/publish_test_thing_policy.json**.
4. Exécutez cette commande pour utiliser le document de stratégie des étapes précédentes pour créer unAWS IoTpolitique.

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

Si la stratégie est créée, la commande renvoie une réponse telle que celle-ci.

```
{
```

```
"policyName": "PubSubTestThingPolicy",
"policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",
"policyDocument": "{\n\"Version\": \"2012-10-17\", \n\"Statement\": [\n\n\"Effect\": \"Allow\", \n\"Action\": [\n\\\"iot:Connect\\\"], \n\"Resource\": [\n\\\"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\\\"], \n\"Effect\": \"Allow\", \n\"Action\": [\n\\\"iot:Publish\\\"], \n\"Resource\": [\n\\\"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\\\"], \n\"Effect\": \"Allow\", \n\"Action\": [\n\\\"iot:Subscribe\\\"], \n\"Resource\": [\n\\\"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\\\"], \n\"Effect\": \"Allow\", \n\"Action\": [\n\\\"iot:Receive\\\"], \n\"Resource\": [\n\\\"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*\\\"], \n\"Effect\": \"Allow\"], \n\"PolicyVersionId\": \"1\"}
```

- Exécutez cette commande pour attacher la stratégie au certificat de l'appareil.
Remplacez *certificateArn* avec le *certificateArn* valeur que vous avez enregistrée plus tôt dans cette section.

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

Si elle aboutit, cette commande ne renvoie rien.

- Exécutez cette commande pour attacher le certificat de périphérique à la AWS IoT ressource de l'objet.
Remplacez *certificateArn* avec le *certificateArn* valeur que vous avez enregistrée plus tôt dans cette section.

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

Si elle aboutit, cette commande ne renvoie rien.

Une fois que vous avez correctement configuré votre appareil dans AWS IoT, vous êtes prêt à continuer [the section called "Configurer l'AWS IoT Fichier de configuration Device Client et client de test MQTT pour démontrer la communication MQTT" \(p. 169\)](#).

Configurer l'AWS IoT Fichier de configuration Device Client et client de test MQTT pour démontrer la communication MQTT

Cette procédure crée un fichier de configuration pour tester le AWS IoT Client de l'appareil.

Pour créer le fichier de configuration afin de tester le AWS IoT Appareil client

- Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi :
 - Ouvrez un éditeur de texte, tel qu'enano.
 - Copiez ce document JSON et collez-le dans votre éditeur de texte ouvert.

```
{
  "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
  "cert": "~/certs/pubsub/device.pem.crt",
  "key": "~/certs/pubsub/private.pem.key",
  "root-ca": "~/certs/AmazonRootCA1.pem",
  "thing-name": "PubSubTestThing",
  "logging": {
    "enable-sdk-logging": true,
    "level": "DEBUG",
    "type": "STDOUT",
```

```
        "file": "",  
    },  
    "jobs": {  
        "enabled": false,  
        "handler-directory": ""  
    },  
    "tunneling": {  
        "enabled": false  
    },  
    "device-defender": {  
        "enabled": false,  
        "interval": 300  
    },  
    "fleet-provisioning": {  
        "enabled": false,  
        "template-name": "",  
        "template-parameters": "",  
        "csr-file": "",  
        "device-key": ""  
    },  
    "samples": {  
        "pub-sub": {  
            "enabled": true,  
            "publish-topic": "test/dc/pubtopic",  
            "publish-file": "",  
            "subscribe-topic": "test/dc/subtopic",  
            "subscribe-file": "~/aws-iot-device-client/log/pubsub_rx_msgs.log"  
        }  
    },  
    "config-shadow": {  
        "enabled": false  
    },  
    "sample-shadow": {  
        "enabled": false,  
        "shadow-name": "",  
        "shadow-input-file": "",  
        "shadow-output-file": ""  
    }  
}
```

- c. Remplacez le*point final* valeur avec le point de terminaison de données de périphérique pour votreCompte AWSque vous avez trouvé dans[the section called “Provisionnez votre appareil dansAWS IoT Core” \(p. 159\)](#).
- d. Enregistrez le fichier dans votre éditeur de texte sous~/**dc-configs/dc-pubsub-config.json**.
- e. Exécutez cette commande pour définir les autorisations sur le nouveau fichier de configuration.

```
chmod 644 ~/dc-configs/dc-pubsub-config.json
```

2. Pour préparer leClient MQTTpour vous abonner à tous les messages MQTT :
 - a. Sur votre ordinateur hôte local, dans le[AWS IoTconsole](#), choisissezClient MQTT.
 - b. DansS'abonner à une rubrique, dansFiltre de rubriques, saisissez#(un signe de livre unique), et choisissezS'abonner.
 - c. Au-dessous duSubscriptionsétiquette, confirmez que vous voyez#(un signe de livre unique).

Laissez la fenêtre avec leClient MQTTouvrez pendant que vous continuez ce didacticiel.

Une fois que vous avez enregistré le fichier et configuré le fichierClient MQTT, vous êtes prêt à continuer [the section called "Étape 2 : Démontrez la publication de messages avec leAWS IoTAppareil client" \(p. 171\)](#).

Étape 2 : Démontrez la publication de messages avec leAWS IoTAppareil client

Les procédures présentées dans cette section démontrent comment leAWS IoTDevice Client peut envoyer des messages MQTT personnalisés et par défaut.

Ces instructions de stratégie de la stratégie que vous avez créée à l'étape précédente pour ces exercices donnent au Raspberry Pi l'autorisation d'effectuer ces actions :

- **iot:Connect**

Donne le nom du clientPubSubTestThing, votre Raspberry Pi exécutant leAWS IoTDevice Client, pour se connecter.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Connect"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing"  
    ]  
}
```

- **iot:Publish**

Donne au Raspberry Pi l'autorisation de publier des messages avec un sujet MQTT detest/dc/pubtopic.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Publish"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic"  
    ]  
}
```

Le **iot:Publish** donne l'autorisation de publier sur les rubriques MQTT répertoriées dans le tableau Resource. Le contenu de ces messages ne sont pas contrôlés par l'énoncé de stratégie.

Publiez le message par défaut à l'aide de laAWS IoTAppareil client

Cette procédure exécute la commande AWS IoTDevice Client afin qu'il publie un seul message MQTT par défaut indiquant que le Client MQTT reçoit et affiche.

Pour envoyer le message MQTT par défaut depuis leAWS IoTAppareil client

1. Assurez-vous que la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi et la fenêtre avec le Client MQTT sont visibles pendant que vous effectuez cette procédure.
2. Dans la fenêtre du terminal, entrez ces commandes pour exécuter leAWS IoTDevice Client utilisant le fichier de configuration créé dans [the section called "Créer le fichier de configuration" \(p. 163\)](#).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-config.json
```

Dans la fenêtre de terminal, leAWS IoTDevice Client affiche les messages d'information et les erreurs éventuelles qui se produisent lors de son exécution.

Si aucune erreur n'est affichée dans la fenêtre du terminal, passez en revue leClient MQTT.

3. DansClient MQTT, dans leSubscriptions, voir la fenêtreHello World !message envoyé autest/dc/pubtopicsujet de message.
4. Si l'icôneAWS IoTDevice Client n'affiche aucune erreur et vous voyezHello World !envoyées à l'test/dc/pubtopicdans le messageClient MQTT, vous avez démontré une connexion réussie.
5. Dans la fenêtre de terminal, entrez^C(Ctrl+C) afin d'arrêter leAWS IoTClient de l'appareil.

Une fois que vous avez démontré que leAWS IoTDevice Client a publié le message MQTT par défaut, vous pouvez continuer vers le[the section called "Publier un message personnalisé à l'aide duAWS IoTClient de l'appareil." \(p. 172\)](#).

Publier un message personnalisé à l'aide duAWS IoTClient de l'appareil.

Les procédures de cette section créent un message MQTT personnalisé, puis exécutent leAWS IoTDevice Client afin qu'il publie le message MQTT personnalisé une fois pour leClient MQTT à recevoir et à afficher.

Créer un message MQTT personnalisé pour leAWS IoTAppareil client

Effectuez ces étapes dans la fenêtre du terminal sur l'ordinateur hôte local connecté à votre Raspberry Pi.

Pour créer un message personnalisé pour leAWS IoTDevice Client à publier

1. Dans la fenêtre du terminal, ouvrez un éditeur de texte tel quenano.
2. Dans l'éditeur de texte, copiez-collez le document JSON suivant. Il s'agit de la charge utile du message MQTT que leAWS IoTDevice Client publie.

```
{
  "temperature": 28,
  "humidity": 80,
  "barometer": 1013,
  "wind": {
    "velocity": 22,
    "bearing": 255
  }
}
```

3. Enregistrez le contenu de l'éditeur de texte sous~/messages/sample-ws-message.json.
4. Entrez la commande suivante pour définir les autorisations du fichier de messages que vous venez de créer.

```
chmod 600 ~/messages/*
```

Pour créer un fichier de configuration pour leAWS IoTDevice Client à utiliser pour envoyer le message personnalisé

1. Dans la fenêtre du terminal, dans un éditeur de texte tel quenano, ouvrez le fichier existantAWS IoTFichier de configuration Device Client :~/dc-configs/dc-pubsub-config.json.
2. Modifier l'outil samplesse présente ainsi. Aucune autre partie de ce fichier ne doit être modifiée.

```
"samples": {  
    "pub-sub": {  
        "enabled": true,  
        "publish-topic": "test/dc/pubtopic",  
        "publish-file": "~/messages/sample-ws-message.json",  
        "subscribe-topic": "test/dc/subtopic",  
        "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
```

3. Enregistrez le contenu de l'éditeur de texte sous~/dc-configs/dc-pubsub-custom-config.json.
4. Exécutez cette commande pour définir les autorisations sur le nouveau fichier de configuration.

```
chmod 644 ~/dc-configs/dc-pubsub-custom-config.json
```

Publiez le message MQTT personnalisé à l'aide de laAWS IoTAppareil client

Cette modification affecte uniquement lescontenuede la charge utile des messages MQTT, de sorte que la stratégie actuelle continuera de fonctionner. Toutefois, si leRubrique MQTT(comme défini parpublish-topicvaleur dans~/dc-configs/dc-pubsub-custom-config.json) a été modifié, leiot::PublishIl faudrait également modifier l'énoncé de stratégie pour permettre au Raspberry Pi de publier sur la nouvelle rubrique MQTT.

Pour envoyer le message MQTT depuis leAWS IoTAppareil client

1. Assurez-vous que la fenêtre du terminal et la fenêtre avec leClient MQTTsont visibles pendant que vous effectuez cette procédure. Assurez-vous également que votreClient MQTTTest toujours abonné au#Filtre de rubriques. Si ce n'est pas le cas, abonnez-vous au#filtre de sujet à nouveau.
2. Dans la fenêtre du terminal, entrez ces commandes pour exécuter leAWS IoTDevice Client utilisant le fichier de configuration créé dans[the section called “Créer le fichier de configuration” \(p. 163\)](#).

```
cd ~/aws-iot-device-client/build  
../aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
```

Dans la fenêtre de terminal, leAWS IoTDevice Client affiche les messages d'information et les erreurs éventuelles qui se produisent lors de son exécution.

- Si aucune erreur n'est affichée dans la fenêtre du terminal, passez en revue le client de test MQTT.
3. DansClient MQTT, dans leSubscriptions, consultez la charge utile des messages personnalisés envoyée à latest/dc/pubtopicsujet de message.
 4. Si l'icôneAWS IoTDevice Client n'affiche aucune erreur et la charge utile des messages personnalisés que vous avez publiée sur letest/dc/pubtopicdans le messageClient MQTT, vous avez publié un message personnalisé avec succès.
 5. Dans la fenêtre de terminal, entrez^C(Ctrl+C) afin d'arrêter leAWS IoTClient de l'appareil.

Une fois que vous avez démontré que leAWS IoTDevice Client a publié une charge utile de message personnalisée, vous pouvez continuer à[the section called “Étape 3 : Démontrer que vous vous abonnez à des messages avec leAWS IoTAppareil client” \(p. 173\)](#).

Étape 3 : Démontrer que vous vous abonnez à des messages avec leAWS IoTAppareil client

Dans cette section, vous allez montrer deux types d'abonnements aux messages :

- Abonnement à un sujet
- Abonnement à un sujet Wild-Card

Ces énoncés de stratégie de la stratégie créée pour ces exercices donnent au Raspberry Pi l'autorisation d'effectuer ces actions :

- **iot:Receive**

Donne leAWS IoTAuthorization Device Client pour recevoir des rubriques MQTT qui correspondent à celles nommées dans leResourceobjet.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Receive"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic"  
    ]  
}
```

- **iot:Subscribe**

Donne leAWS IoTAuthorization Device Client pour s'abonner aux filtres de rubriques MQTT qui correspondent à ceux nommés dans leResourceobjet.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Subscribe"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic"  
    ]  
}
```

Abonnez-vous à un seul sujet de message MQTT

Cette procédure explique comment AWS IoTDevice Client peut s'abonner aux messages MQTT et les enregistrer.

Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, indiquez le contenu de `~/dc-configs/dc-pubsub-custom-config.json` ou ouvrez le fichier dans un éditeur de texte pour consulter son contenu. Recherchez le plugin `samples` qui devrait ressembler à ce qui suit.

```
"samples": {  
    "pub-sub": {  
        "enabled": true,  
        "publish-topic": "test/dc/pubtopic",  
        "publish-file": "~/messages/sample-ws-message.json",  
        "subscribe-topic": "test/dc/subtopic",  
        "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"  
    }  
}
```

Remarquez que `subscribe-topic` est la rubrique MQTT dans laquelle le AWS IoTDevice Client s'abonnera lorsqu'il est exécuté. Le AWS IoTDevice Client écrit les charges utiles des messages qu'il reçoit de cet abonnement dans le fichier nommé dans `subscribe-file`.

Pour vous abonner à un sujet de message MQTT à partir duAWS IoTAppareil client

1. Assurez-vous que la fenêtre du terminal et la fenêtre avec le client de test MQTT sont visibles pendant que vous effectuez cette procédure. Assurez-vous également que votreClient MQTTest toujours abonné au#Filtre de rubriques. Si ce n'est pas le cas, abonnez-vous au#filtre de sujet à nouveau.
2. Dans la fenêtre du terminal, entrez ces commandes pour exécuter leAWS IoTDevice Client utilisant le fichier de configuration créé dans[the section called "Créer le fichier de configuration" \(p. 163\)](#).

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-custom-config.json
```

Dans la fenêtre de terminal, leAWS IoTDevice Client affiche les messages d'information et les erreurs éventuelles qui se produisent lors de son exécution.

Si aucune erreur n'est affichée dans la fenêtre du terminal, continuez dans leAWS IoTconsole

3. DansAWS IoT, dans la consoleClient MQTT, choisissez lePublier dans une rubriqueonglet.
4. DansNom de la rubrique, saisissez**test/dc/subtopic**
5. DansCharge utile des messages, passez en revue le contenu du message.
6. ChoisissezPublierpour publier le message MQTT.
7. Dans la fenêtre du terminal, surveillez lamessage reçue de l'outilAWS IoTDevice Client qui se présente sous la forme suivante.

```
2021-11-10T16:02:20.890Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
subscribe topic, size: 45 bytes
```

8. Après avoir vu lemessage reçue indiquant que le message a été reçu, entrez^C(Ctrl+C) afin d'arrêter leAWS IoTClient de l'appareil.
9. Saisissez cette commande pour afficher la fin du fichier journal des messages et voir le message que vous avez publié à partir duClient MQTT.

```
tail ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
```

En affichant le message dans le fichier journal, vous avez démontré que leAWS IoTDevice Client a reçu le message que vous avez publié depuis le client de test MQTT.

S'abonner à plusieurs rubriques de messages MQTT à l'aide de caractères génériques

Ces procédures démontrent comment leAWS IoTDevice Client peut s'abonner aux messages MQTT et les enregistrer à l'aide de caractères génériques. Pour ce faire, vous allez :

1. Mettez à jour le filtre de rubriques que leAWS IoTDevice Client utilise pour s'abonner aux rubriques MQTT.
2. Mettez à jour la stratégie utilisée par l'appareil pour autoriser les nouveaux abonnements.
3. Exécutez leAWS IoTDevice Client et publiez des messages depuis la console de test MQTT.

Pour créer un fichier de configuration pour s'abonner à plusieurs rubriques de message MQTT à l'aide d'un filtre de rubriques MQTT générique

1. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, ouvrez~/**dc-configs/dc-pubsub-custom-config.json**pour éditer et localiser lesamplesobjet.
2. Dans l'éditeur de texte, localisez lesampleset mettez à jour lesubscribe-topics présentés ainsi.

```
"samples": {  
    "pub-sub": {  
        "enabled": true,  
        "publish-topic": "test/dc/pubtopic",  
        "publish-file": "~/messages/sample-ws-message.json",  
        "subscribe-topic": "test/dc/#",  
        "subscribe-file": "~/.aws-iot-device-client/log/pubsub_rx_msgs.log"
```

La nouvelle `subscribe-topic` valeur est une [Filtre de rubriques MQTT \(p. 116\)](#) avec un caractère joker MQTT à la fin. Ceci décrit un abonnement à toutes les rubriques MQTT qui commencent par `test/dc/`. Le AWS IoT Device Client écrit les charges utiles des messages qu'il reçoit de cet abonnement dans le fichier nommé `subscribe-file`.

3. Enregistrez le fichier de configuration modifié sous `~/dc-configs/dc-pubsub-wild-config.json` et quittez l'éditeur.

Pour modifier la stratégie utilisée par votre Raspberry Pi afin de permettre l'abonnement et la réception de plusieurs rubriques de messages MQTT

1. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, dans votre éditeur de texte préféré, ouvrez `~/policies/pubsub_test_thing_policy.json` pour éditer, puis localiser le `iot::Subscribe` et `iot::Receive` instructions de stratégie dans le fichier.
2. Dans `iot::Subscribe`, mettez à jour la chaîne de l'objet Resource pour la remplacez `subtopic` avec `*` afin que l'adresse se présente ainsi.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Subscribe"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/*"  
    ]  
}
```

Note

Le [Filtre de thème MQTT caractères génériques \(p. 116\)](#) sont les + (signe plus) et # (signe livre). Une demande d'abonnement avec un # à la fin, s'abonne à tous les sujets qui commencent par la chaîne qui précède le # caractère (par exemple, `test/dc/` dans ce cas).

La valeur de la ressource dans l'énoncé de stratégie qui autorise cet abonnement doit toutefois utiliser un * (un astérisque) à la place du # (signe livre) dans le filtre de rubrique ARN. En effet, le processeur de stratégies utilise un caractère joker différent de celui utilisé par MQTT.

Pour plus d'informations sur l'utilisation de caractères génériques pour les rubriques et les filtres de rubriques dans les stratégies, voir [Utilisation de caractères génériques dans MQTT et les politiques AWS IoT Core \(p. 376\)](#).

3. Dans `iot::Receive`, mettez à jour la chaîne de l'objet Resource pour la remplacez `subtopic` avec `*` afin que l'adresse se présente ainsi.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:Receive"  
    ],  
    "Resource": [  
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*"
```

```
        ]  
    }
```

4. Enregistrez le document de stratégie mis à jour sous~/policies/pubsub_wild_test_thing_policy.jsonet quittez l'éditeur.
5. Entrez cette commande pour mettre à jour la stratégie de ce didacticiel afin d'utiliser les nouvelles définitions de ressources.

```
aws iot create-policy-version \  
--set-as-default \  
--policy-name "PubSubTestThingPolicy" \  
--policy-document "file://~/policies/pubsub_wild_test_thing_policy.json"
```

Si la commande réussit, elle renvoie une réponse telle que celle-ci. Remarquez quepolicyVersionIdest maintenant2, indiquant qu'il s'agit de la deuxième version de cette politique.

Si vous avez mis à jour la stratégie avec succès, vous pouvez passer à la procédure suivante.

```
{  
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/PubSubTestThingPolicy",  
    "policyDocument": "{\n        \"Version\": \"2012-10-17\",  
        \"Statement\": [\n            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Connect\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Publish\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Subscribe\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/*\"  
            },  
            {\n                \"Effect\": \"Allow\",  
                \"Action\": \"iot:Receive\",  
                \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*\"  
            }  
        ],  
        \"policyVersionId\": \"2\",  
        \"isDefaultVersion\": true  
}
```

Si vous obtenez une erreur indiquant qu'il existe trop de versions de stratégie pour en enregistrer une nouvelle, entrez cette commande pour répertorier les versions actuelles de la stratégie. Consultez la liste renvoyée par cette commande pour trouver une version de stratégie que vous pouvez supprimer.

```
aws iot list-policy-versions --policy-name "PubSubTestThingPolicy"
```

Entrez cette commande pour supprimer une version dont vous n'avez plus besoin. Notez que vous ne pouvez pas supprimer la version de la stratégie par défaut. La version de stratégie par défaut est celle avec unisDefaultVersionvaleur detruue.

```
aws iot delete-policy-version \  
--policy-name "PubSubTestThingPolicy" \  
--policy-version-id policyId
```

Après avoir supprimé une version de stratégie, réessayez cette étape.

Avec le fichier de configuration et la stratégie mis à jour, vous êtes prêt à démontrer les abonnements aux caractères génériques avec leAWS IoTClient de l'appareil.

Pour démontrer comment leAWS IoTDevice Client s'abonne à plusieurs rubriques de message MQTT et les reçoit

1. DansClient MQTT, vérifiez les abonnements. Si l'icôneClient MQTTTest abonné à la#Filtre de rubriques, passez à l'étape suivante. Dans le cas contraire, dans leClient MQTT, dansS'abonner à une rubrique, dansFiltre de rubriques, saisissez#(un signe de livre), puis choisissezS'abonnerpour y souscrire.
2. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, entrez ces commandes pour démarrer leAWS IoTClient de l'appareil.

```
cd ~/aws-iot-device-client/build
./aws-iot-device-client --config-file ~/dc-configs/dc-pubsub-wild-config.json
```

3. En regardant leAWS IoTSortie Device Client dans la fenêtre du terminal sur l'ordinateur hôte local, revenez à laClient MQTT. DansPublier dans une rubrique, dansNom de la rubrique, saisissez**test/dc/subtopic**ChoisissezPublier.
4. Dans la fenêtre du terminal, confirmez que le message a été reçu en recherchant un message tel que :

```
2021-11-10T16:34:20.101Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
subscribe topic, size: 76 bytes
```

5. En regardant leAWS IoTSortie Device Client dans la fenêtre du terminal de l'ordinateur hôte local, revenez à laTest client MQTT. DansPublier dans une rubrique, dansNom de la rubrique, saisissez**test/dc/subtopic2**ChoisissezPublier.
6. Dans la fenêtre du terminal, confirmez que le message a été reçu en recherchant un message tel que :

```
2021-11-10T16:34:32.078Z [DEBUG] {samples/PubSubFeature.cpp}: Message received on
subscribe topic, size: 77 bytes
```

7. Après avoir vu les messages confirmant que les deux messages ont été reçus, entrez^C(Ctrl+C) afin d'arrêter leAWS IoTClient de l'appareil.
8. Saisissez cette commande pour afficher la fin du fichier journal des messages et voir le message que vous avez publié à partir duTest client MQTT.

```
tail -n 20 ~/.aws-iot-device-client/log/pubsub_rx_msgs.log
```

Note

Le fichier journal contient uniquement des charges utiles de messages. Les rubriques de message ne sont pas enregistrées dans le fichier journal des messages reçus.

Il se peut que vous voyiez également le message publié par leAWS IoTDevice Client dans le journal reçu. En effet, le filtre de rubrique générique inclut cette rubrique de message et, parfois, la demande d'abonnement peut être traitée par le courtier de messages avant que le message publié ne soit envoyé aux abonnés.

Les entrées du fichier journal montrent que les messages ont été reçus. Vous pouvez répéter cette procédure en utilisant d'autres noms de rubriques. Tous les messages dont le nom de sujet commence par**test/dc/doivent** être reçus et enregistrés. Les messages dont les noms de rubriques commencent par un autre texte sont ignorés.

Après avoir démontré comment leAWS IoTDevice Client peut publier et s'abonner aux messages MQTT, continuer à[Didacticiel : Démontrez des actions à distance \(tâches\) avecAWS IoTPériphérique](#)
[Périphérique \(p. 179\)](#).

Didacticiel : Démontrez des actions à distance (tâches) avec AWS IoT Périphérique Périphérique

Dans ces didacticiels, vous allez configurer et déployer des tâches sur votre Raspberry Pi pour montrer comment envoyer des opérations à distance à vos appareils IoT.

Pour démarrer ce didacticiel :

- Configurez votre ordinateur hôte local, un Raspberry Pi, tel qu'il est utilisé dans [la section précédente \(p. 165\)](#).
- Si vous n'avez pas terminé le didacticiel de la section précédente, vous pouvez essayer ce didacticiel en utilisant le Raspberry Pi avec une carte microSD contenant l'image que vous avez enregistrée après avoir installé le AWS IoT Périphérique : Périphérique ([Facultatif](#)) [Enregistrez l'image de la carte microSD \(p. 158\)](#).
- Si vous avez déjà exécuté cette démo, consultez [??? \(p. 190\)](#) pour tout supprimer AWS IoT Resources que vous avez créées lors d'exécutions précédentes pour éviter les erreurs de duplication des ressources.

Ce didacticiel vous prendra environ 45 minutes.

Lorsque vous avez terminé avec cette rubrique :

- Vous aurez démontré différentes manières dont votre appareil IoT peut utiliser AWS IoT Core pour exécuter des opérations à distance gérées par AWS IoT.

Périphérique requise :

- Votre environnement de développement et de test local dans lequel vous avez effectué les tests [une section précédente \(p. 156\)](#)
- Le Raspberry Pi que vous avez testé dans [une section précédente \(p. 156\)](#)
- La carte mémoire microSD du Raspberry Pi que vous avez testée dans [une section précédente \(p. 156\)](#)

Procédures de ce didacticiel

- [Étape 1 : Préparez le Raspberry Pi pour exécuter des tâches \(p. 179\)](#)
- [Étape 2 : Création et exécution de la tâche dans AWS IoT \(p. 185\)](#)

Étape 1 : Préparez le Raspberry Pi pour exécuter des tâches

Les procédures de cette section expliquent comment préparer votre Raspberry Pi à exécuter des tâches à l'aide de AWS IoT Périphérique : Périphérique

Note

Ces procédures sont spécifiques à l'appareil. Si vous souhaitez exécuter les procédures décrites dans cette section avec plusieurs appareils à la fois, chaque appareil aura besoin de sa propre politique, d'un certificat et d'un nom d'objet uniques et spécifiques à l'appareil. Pour attribuer à chaque appareil ses ressources uniques, effectuez cette procédure une fois pour chaque appareil tout en modifiant les éléments spécifiques au périphérique, comme décrit dans les procédures.

Procédures de ce didacticiel

- [Provisionnez votre Raspberry Pi pour démontrer \(p. 180\)](#)

- [Configuration de AWS IoT Device Client pour exécuter l'agent de tâches \(p. 184\)](#)

Provisionnez votre Raspberry Pi pour démontrer

Les procédures décrites dans cette section approvisionnent votre Raspberry Pi AWS IoT créant AWS IoT des ressources et des certificats d'appareils pour cela.

Créez et téléchargez des fichiers de certificat d'appareil pour démontrer AWS IoT emplois

Cette procédure permet de créer les fichiers de certificat de l'appareil pour cette démo.

Si vous préparez plusieurs appareils, cette procédure doit être effectuée sur chaque appareil.

Pour créer et télécharger les fichiers de certificat de l'appareil pour votre Raspberry Pi :

Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, entrez ces commandes.

1. Entrez la commande suivante pour créer les fichiers de certificat de votre appareil.

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

La commande renvoie une réponse telle que la suivante. Enregistrer *certificateArn* valeur pour une utilisation ultérieure.

```
{
"certificateArn": "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/76e7e4edb3e52f52334be2f387a06145b2aa4c7fc810f3aea2d92abc227d269",
"certificateId": "76e7e4edb3e52f5233EXAMPLE7a06145b2aa4c7fc810f3aea2d92abc227d269",
"certificatePem": "-----BEGIN CERTIFICATE-----
\nMIIDWTCCAkGgAwIBAgI_SHORTENED_FOR_EXAMPLE_Lgn4jfjgtS\n-----END CERTIFICATE-----\n",
"keyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBkgkhkiG9w0BA_SHORTENED_FOR_EXAMPLE_ImwIDAQAB\n-----END PUBLIC KEY-----\n",
    "PrivateKey": "-----BEGIN RSA PRIVATE KEY-----
\nMIIEowIBAAKCAQE_SHORTENED_FOR_EXAMPLE_T9RoDiukY\n-----END RSA PRIVATE KEY-----\n"
}
}
```

2. Entrez les commandes suivantes pour définir les autorisations sur le répertoire des certificats et ses fichiers.

```
chmod 700 ~/certs/jobs
chmod 644 ~/certs/jobs/*
chmod 600 ~/certs/jobs/private.pem.key
```

3. Exécutez cette commande pour vérifier les autorisations sur vos répertoires et fichiers de certificats.

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

Le résultat de la commande doit être identique à ce que vous voyez ici, sauf que les dates et heures du fichier seront différentes.

```
-rw-r--r-- 1 pi pi 1220 Oct 28 13:02 device.pem.crt
```

```
-rw----- 1 pi pi 1675 Oct 28 13:02 private.pem.key
-rw-r--r-- 1 pi pi 451 Oct 28 13:02 public.pem.key
```

Après avoir téléchargé les fichiers de certificat de l'appareil sur votre Raspberry Pi, vous êtes prêt à continuer [the section called "Provisionnez votre Raspberry Pi pour démontrer" \(p. 180\)](#).

Création AWS IoT des ressources pour démontrer AWS IoT emplois

Créer le AWS IoT resources pour cet appareil.

Si vous préparez plusieurs appareils, cette procédure doit être effectuée pour chaque appareil.

Pour approvisionner votre appareil dans AWS IoT :

Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi :

1. Saisissez la commande suivante pour obtenir l'adresse du point de terminaison de données de l'Périphérique pour votre Compte AWS.

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

La valeur de point de terminaison n'a pas changé depuis la dernière exécution de cette commande. La réexécution de la commande ici permet de trouver et de coller facilement la valeur du point de terminaison des données dans le fichier de configuration utilisé dans ce didacticiel.

Dans la `describe-endpoint` La commande renvoie une réponse telle que la suivante. Enregistrez la `endpointAddress` valeur pour une utilisation ultérieure.

```
{  
    "endpointAddress": "a3qjEXAMPLEffp-ats.iot.us-west-2.amazonaws.com"  
}
```

2. Remplacer `uniqueThingName` avec un nom unique pour votre Périphérique. Si vous souhaitez exécuter ce didacticiel sur plusieurs appareils, donnez à chaque appareil son propre nom. Par exemple, `TestDevice01`, `TestDevice02`, et ainsi de suite.

Saisissez cette commande pour en créer une AWS IoT resource d'objets pour votre Raspberry Pi.

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

Parce qu'un AWS IoT une ressource est une représentation virtuelle de votre appareil dans le cloud, nous pouvons créer plusieurs ressources dans AWS IoT à utiliser à des fins différentes. Ils peuvent tous être utilisés par le même appareil IoT physique pour représenter différents aspects de l'appareil.

Note

Lorsque vous souhaitez sécuriser la politique pour plusieurs appareils, vous pouvez utiliser `${iot:Thing.ThingName}` au lieu du nom statique de l'objet, `uniqueThingName`.

Ces didacticiels n'utiliseront qu'une seule ressource à la fois par appareil. De cette façon, dans ces tutoriels, ils représentent les différentes démos de sorte qu'après avoir créé le AWS IoT resources pour une démo, vous pouvez revenir en arrière et répéter les démos en utilisant les ressources que vous avez créées spécifiquement pour chacune d'elles.

Si votre AWS IoT une ressource de l'objet a été créée, la commande renvoie une réponse telle que celle ci-après. Enregistrez la `thingArn` valeur à utiliser ultérieurement lors de la création de la tâche à exécuter sur cet appareil.

```
{  
    "thingName": "uniqueThingName",  
    "thingArn": "arn:aws:iot:us-west-2:57EXAMPLE833:thing/uniqueThingName",  
    "thingId": "8ea78707-32c3-4f8a-9232-14bEXAMPLEfd"  
}
```

3. Dans la fenêtre du terminal :

- Ouvrez un éditeur de texte tel que nano.
- Copiez ce document JSON et collez-le dans votre éditeur de texte ouvert.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:client/uniqueThingName"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/job/*",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/$aws/things/uniqueThingName/jobs/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic",  
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:DescribeJobExecution",  
            ]  
        }  
    ]  
}
```

```

        "iot:GetPendingJobExecutions",
        "iot:StartNextPendingJobExecution",
        "iot:UpdateJobExecution"
    ],
    "Resource": [
        "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName"
    ]
}
]
}
}

```

- Dans l'éditeur, dans `Resource` de chaque déclaration de politique, remplacez `us-west-west-2:57 EXAMPLE 833` avec votre Région AWS, un caractère deux-points (:) et vos 12 chiffres Compte AWS Numéro.
 - Dans l'éditeur, dans chaque déclaration de politique, remplacez `uniqueThingName` avec le nom de l'objet que vous avez donné à cette ressource.
 - Enregistrez le fichier dans votre éditeur de texte sous `~/policies/jobs_test_thing_policy.json`.
- Si vous exécutez cette procédure sur plusieurs appareils, enregistrez le fichier sous ce nom sur chaque appareil.
- Remplacer `uniqueThingName` avec le nom de l'objet pour le périphérique, puis exécutez cette commande pour créer une AWS IoT politique adaptée à cet appareil.

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

Si la politique est créée, la commande renvoie une réponse comme celle-ci.

```
{
    "policyName": "JobTestPolicyForuniqueThingName",
    "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/JobTestPolicyForuniqueThingName",
    "policyDocument": "{\n    \"Version\": \"2012-10-17\", \"Statement\": [\n        {\n            \"Effect\": \"Allow\", \"Action\": \"iot:Connect\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:client/PubSubTestThing\"\n        },\n        {\n            \"Effect\": \"Allow\", \"Action\": \"iot:Publish\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic\"\n        },\n        {\n            \"Effect\": \"Allow\", \"Action\": \"iot:Subscribe\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic\"\n        },\n        {\n            \"Effect\": \"Allow\", \"Action\": \"iot:Receive\", \"Resource\": \"arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/*\"\n        }\n    ],\n    \"PolicyVersionId\": \"1\"\n}
```

- Remplacer `uniqueThingName` avec le nom de l'objet pour l'appareil et `certificateArn` avec le `certificateArn` valeur que vous avez enregistrée précédemment dans cette section pour cet appareil, puis exécutez cette commande pour associer la politique au certificat de l'appareil.

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

Si elle aboutit, cette commande ne renvoie rien.

- Remplacer `uniqueThingName` par le nom de l'appareil, remplacez `certificateArn` avec le `certificateArn` valeur que vous avez enregistrée précédemment dans cette section, puis exécutez cette commande pour joindre le certificat de l'appareil au AWS IoT ressource d'objet.

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

Si elle aboutit, cette commande ne renvoie rien.

Après avoir approvisionné avec succès votre Raspberry Pi, vous êtes prêt à répéter cette section pour un autre Raspberry Pi lors de votre test ou, si tous les appareils ont été approvisionnés, à continuer [the section called "Configuration de AWS IoTDevice Client pour exécuter l'agent de tâches" \(p. 184\)](#).

Configuration de AWS IoTDevice Client pour exécuter l'agent de tâches

Cette procédure crée un fichier de configuration pour AWS IoTDevice Client pour exécuter l'agent de tâches :

Remarque : si vous préparez plusieurs appareils, cette procédure doit être effectuée sur chaque appareil.

Pour créer le fichier de configuration afin de tester AWS IoTPériphérique :

1. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi :
 - a. Ouvrez un éditeur de texte tel que nano.
 - b. Copiez ce document JSON et collez-le dans votre éditeur de texte ouvert.

```
{
  "endpoint": "a3qEXAMPLEaffp-ats.iot.us-west-2.amazonaws.com",
  "cert": "~/certs/jobs/device.pem.crt",
  "key": "~/certs/jobs/private.pem.key",
  "root-ca": "~/certs/AmazonRootCA1.pem",
  "thing-name": "uniqueThingName",
  "logging": {
    "enable-sdk-logging": true,
    "level": "DEBUG",
    "type": "STDOUT",
    "file": ""
  },
  "jobs": {
    "enabled": true,
    "handler-directory": ""
  },
  "tunneling": {
    "enabled": false
  },
  "device-defender": {
    "enabled": false,
    "interval": 300
  },
  "fleet-provisioning": {
    "enabled": false,
    "template-name": "",
    "template-parameters": "",
    "csr-file": "",
    "device-key": ""
  },
  "samples": {
    "pub-sub": {
      "enabled": false,
      "publish-topic": "",
      "publish-file": "",
      "subscribe-topic": ""
    }
  }
}
```

```
        "subscribe-file": ""  
    },  
    "config-shadow": {  
        "enabled": false  
    },  
    "sample-shadow": {  
        "enabled": false,  
        "shadow-name": "",  
        "shadow-input-file": "",  
        "shadow-output-file": ""  
    }  
}
```

- c. Remplacez le *point final* valeur avec valeur de point de terminaison des données de votre appareilCompte AWSque vous avez trouvé dans[the section called “Provisionnez votre appareil dansAWS IoT Core” \(p. 159\)](#).
 - d. Remplacer *uniqueThingName* avec le nom de l'objet que vous avez utilisé pour cet appareil.
 - e. Enregistrez le fichier dans votre éditeur de texte sous~/dc-configs/dc-jobs-config.json.
2. Exécutez cette commande pour définir les autorisations du nouveau fichier de configuration.

```
chmod 644 ~/dc-configs/dc-jobs-config.json
```

Vous n'utiliserez pas leClient de test MQTTpour ce test. Alors que l'appareil échangera des messages MQTT liés aux tâches avecAWS IoT, les messages de progression de la tâche ne sont échangés qu'avec l'appareil qui exécute la tâche. Comme les messages de progression de la tâche ne sont échangés qu'avec l'appareil qui exécute la tâche, vous ne pouvez pas vous y abonner depuis un autre appareil, tel que leAWS IoTConsole.

Après avoir enregistré le fichier de configuration, vous êtes prêt à continuer[the section called “Étape 2 : Crédit et exécution de la tâche dansAWS IoT” \(p. 185\)](#).

Étape 2 : Crédit et exécution de la tâche dansAWS IoT

Les procédures décrites dans cette section permettent de créer un document de travail et unAWS IoTressource d'emploi. Après avoir créé la ressource d'emploi,AWS IoTvoe le document de travail aux cibles de travail spécifiées sur lesquelles un agent de tâches applique le document de travail au terminal ou au client.

Procédures décrites dans cette section

- [Créez et stockez le document de travail de la tâche \(p. 185\)](#)
- [Exécuter une tâche dansAWS IoTpour un appareil IoT \(p. 186\)](#)

Créez et stockez le document de travail de la tâche

Cette procédure crée un document de travail simple à inclure dansAWS IoTressource d'emploi. Ce document de travail affiche « Bonjour tout le monde ! » sur l'objectif du poste.

Pour créer et stocker un document de travail :

1. Sélectionnez le compartiment Amazon S3 dans lequel vous allez enregistrer votre document de travail. Si vous ne disposez pas d'un compartiment Amazon S3 à utiliser à cette fin, vous aurez besoin d'en créer un. Pour plus d'informations sur la création de compartiments Amazon S3, consultez les rubriques dans[Premiers pas avec Amazon S3](#).
2. Création et enregistrement du document de tâche pour cette tâche

- a. Sur votre ordinateur hôte local, ouvrez un éditeur de texte.
- b. Copiez-collez ce texte dans l'éditeur.

```
{  
    "operation": "echo",  
    "args": ["Hello world!"]  
}
```

- c. Sur l'ordinateur hôte local, enregistrez le contenu de l'éditeur dans un fichier nommé **hello-world-job.json**.
- d. Vérifiez que le fichier a été correctement enregistré. Certains éditeurs de texte ajoutent automatiquement .txt au nom du fichier lorsqu'ils enregistrent un fichier texte. Si votre éditeur a ajouté .txt au nom du fichier, corrigez le nom du fichier avant de continuer.
3. Remplacez le **chemin_to_fichier** avec le chemin vers **hello-world-job.json**, s'il ne se trouve pas dans votre répertoire actuel, remplacez **s3_bucket_name** avec le chemin du compartiment Amazon S3 vers le compartiment que vous avez sélectionné, puis exécutez cette commande pour placer votre document de travail dans le compartiment Amazon S3.

```
aws s3api put-object \  
--key hello-world-job.json \  
--body path_to_file/hello-world-job.json --bucket s3_bucket_name
```

L'URL du document de travail qui identifie le document de travail que vous avez stocké dans Amazon S3 est déterminée en remplaçant les **s3_bucket_name** et **Région AWS** dans l'URL suivante. Enregistrez l'URL résultante pour l'utiliser ultérieurement en tant que **job_document_path**

```
https://s3\_bucket\_name.s3.AWS\_Region.amazonaws.com/hello-world-job.json
```

Note

AWS la sécurité vous empêche d'ouvrir cette URL en dehors de votre Compte AWS, par exemple à l'aide d'un navigateur. L'URL est utilisée par AWS IoT moteur de tâches, qui a accès au fichier, par défaut. Dans un environnement de production, vous devez vous assurer que votre AWS IoT les services ont l'autorisation d'accéder aux documents de tâche stockés dans Simple Storage Service (Amazon S3).

Après avoir enregistré l'URL du document de travail, passez à [the section called “Exécuter une tâche dans AWS IoT pour un appareil IoT” \(p. 186\)](#).

Exécuter une tâche dans AWS IoT pour un appareil IoT

Les procédures décrites dans cette section lancent AWS IoT Device Client sur votre Raspberry Pi pour exécuter l'agent de tâches sur l'appareil afin d'attendre l'exécution des tâches. Il crée également une ressource d'emploi dans AWS IoT, qui enverra la tâche à votre appareil IoT et s'exécutera sur celui-ci.

Note

Cette procédure exécute une tâche sur un seul appareil.

Pour démarrer l'agent de jobs sur votre Raspberry Pi :

1. Dans la fenêtre du terminal de votre ordinateur hôte local connecté à votre Raspberry Pi, exécutez cette commande pour démarrer AWS IoT Périphérique : Périphérique

```
cd ~/aws-iot-device-client/build
```

```
./aws-iot-device-client --config-file ~/dc-configs/dc-jobs-config.json
```

2. Dans une fenêtre de terminal, vérifiez que AWS IoT Device Client et affiche ces messages

```
2021-11-15T18:45:56.708Z [INFO] {Main.cpp}: Jobs is enabled
:
:
2021-11-15T18:45:56.708Z [INFO] {Main.cpp}: Client base has been notified that Jobs has started
2021-11-15T18:45:56.708Z [INFO] {JobsFeature.cpp}: Running Jobs!
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to startNextPendingJobExecution accepted and rejected
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to nextJobChanged events
2021-11-15T18:45:56.708Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to updateJobExecutionStatusAccepted for jobId +
2021-11-15T18:45:56.738Z [DEBUG] {JobsFeature.cpp}: Ack received for SubscribeToUpdateJobExecutionAccepted with code {0}
2021-11-15T18:45:56.739Z [DEBUG] {JobsFeature.cpp}: Attempting to subscribe to updateJobExecutionStatusRejected for jobId +
2021-11-15T18:45:56.753Z [DEBUG] {JobsFeature.cpp}: Ack received for SubscribeToNextJobChanged with code {0}
2021-11-15T18:45:56.760Z [DEBUG] {JobsFeature.cpp}: Ack received for SubscribeToStartNextJobRejected with code {0}
2021-11-15T18:45:56.776Z [DEBUG] {JobsFeature.cpp}: Ack received for SubscribeToStartNextJobAccepted with code {0}
2021-11-15T18:45:56.776Z [DEBUG] {JobsFeature.cpp}: Ack received for SubscribeToUpdateJobExecutionRejected with code {0}
2021-11-15T18:45:56.777Z [DEBUG] {JobsFeature.cpp}: Publishing startNextPendingJobExecutionRequest
2021-11-15T18:45:56.785Z [DEBUG] {JobsFeature.cpp}: Ack received for StartNextPendingJobPub with code {0}
2021-11-15T18:45:56.785Z [INFO] {JobsFeature.cpp}: No pending jobs are scheduled, waiting for the next incoming job
```

3. Dans la fenêtre du terminal, après avoir vu ce message, passez à la procédure suivante et créez la ressource de travail. Notez qu'il ne s'agit peut-être pas de la dernière entrée de la liste.

```
2021-11-15T18:45:56.785Z [INFO] {JobsFeature.cpp}: No pending jobs are scheduled, waiting for the next incoming job
```

Pour créer un AWS IoT ressource d'emploi

1. Sur votre ordinateur hôte local :

- a. Remplacer *job_document_url* avec l'URL du document de travail à partir de [the section called "Créez et stockez le document de travail de la tâche" \(p. 185\)](#).
- b. Remplacer *thing_arn* avec l'ARN de la ressource objet que vous avez créée pour votre appareil, puis exécutez cette commande.

```
aws iot create-job \
--job-id hello-world-job-1 \
--document-source "job_document_url" \
--targets "thing_arn" \
--target-selection SNAPSHOT
```

En cas de succès, la commande renvoie un résultat tel que celui-ci.

```
{  
    "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-1",  
    "jobId": "hello-world-job-1"  
}
```

2. Dans une fenêtre de terminal, vous devriez voir une sortie de AWS IoTDevice Client comme celui-ci.

```
2021-11-15T18:02:26.688Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,  
waiting for the next incoming job  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Job ids differ  
2021-11-15T18:10:24.890Z [INFO]  {JobsFeature.cpp}: Executing job: hello-world-job-1  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Attempting to update job execution  
status!  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Not including stdout with the  
status details  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Not including stderr with the  
status details  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Assuming executable is in PATH  
2021-11-15T18:10:24.890Z [INFO]  {JobsFeature.cpp}: About to execute: echo Hello world!  
2021-11-15T18:10:24.890Z [DEBUG] {Retry.cpp}: Retryable function starting, it will  
retry until success  
2021-11-15T18:10:24.890Z [DEBUG] {JobsFeature.cpp}: Created EphemeralPromise for  
ClientToken 3TEWba9Xj6 in the updateJobExecution promises map  
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Child process now running  
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Child process about to call execvp  
2021-11-15T18:10:24.890Z [DEBUG] {JobEngine.cpp}: Parent process now running, child PID  
is 16737  
2021-11-15T18:10:24.891Z [DEBUG] {16737}: Hello world!  
2021-11-15T18:10:24.891Z [DEBUG] {JobEngine.cpp}: JobEngine finished waiting for child  
process, returning 0  
2021-11-15T18:10:24.891Z [INFO]  {JobsFeature.cpp}: Job exited with status: 0  
2021-11-15T18:10:24.891Z [INFO]  {JobsFeature.cpp}: Job executed successfully!  
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Attempting to update job execution  
status!  
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Not including stdout with the  
status details  
2021-11-15T18:10:24.891Z [DEBUG] {JobsFeature.cpp}: Not including stderr with the  
status details  
2021-11-15T18:10:24.892Z [DEBUG] {Retry.cpp}: Retryable function starting, it will  
retry until success  
2021-11-15T18:10:24.892Z [DEBUG] {JobsFeature.cpp}: Created EphemeralPromise for  
ClientToken GmQ0HTzWGg in the updateJobExecution promises map  
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Ack received for  
PublishUpdateJobExecutionStatus with code {0}  
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Removing ClientToken 3TEWba9Xj6  
from the updateJobExecution promises map  
2021-11-15T18:10:24.905Z [DEBUG] {JobsFeature.cpp}: Success response after  
UpdateJobExecution for job hello-world-job-1  
2021-11-15T18:10:24.917Z [DEBUG] {JobsFeature.cpp}: Ack received for  
PublishUpdateJobExecutionStatus with code {0}  
2021-11-15T18:10:24.918Z [DEBUG] {JobsFeature.cpp}: Removing ClientToken GmQ0HTzWGg  
from the updateJobExecution promises map  
2021-11-15T18:10:24.918Z [DEBUG] {JobsFeature.cpp}: Success response after  
UpdateJobExecution for job hello-world-job-1  
2021-11-15T18:10:25.861Z [INFO]  {JobsFeature.cpp}: No pending jobs are scheduled,  
waiting for the next incoming job
```

3. Alors que AWS IoT Le client de l'appareil est en cours d'exécution et attend une tâche. Vous pouvez soumettre une autre tâche en modifiant le job-id valeur et réexécution de create-job à partir de l'étape 1.

Lorsque vous avez terminé d'exécuter des tâches, dans la fenêtre du terminal, entrez^C(Control-C) pour arrêter AWS IoT Périphérique : Périphérique

Didacticiel : Nettoyage après l'exécution AWS IoTTutoriels Device Client

Les procédures de ce didacticiel vous expliquent comment supprimer les fichiers et les ressources que vous avez créés tout en terminant les didacticiels de ce parcours d'apprentissage.

Procédures de ce didacticiel

- [Étape 1 : Nettoyez vos appareils après la création de démonstrations avec leAWS IoTClient d'appareil \(p. 189\)](#)
- [Étape 2 : Nettoyage de votreCompte AWSaprès la création de démonstrations avec leAWS IoTClient d'appareil \(p. 190\)](#)

Étape 1 : Nettoyez vos appareils après la création de démonstrations avec leAWS IoTClient d'appareil

Ce didacticiel décrit deux options pour nettoyer la carte microSD après avoir créé les démos dans ce parcours d'apprentissage. Choisissez l'option qui fournit le niveau de sécurité dont vous avez besoin.

Notez que le nettoyage de la carte microSD de l'appareil ne supprime aucun AWS IoT ressources que vous avez créées. Pour nettoyer AWS IoT après avoir nettoyé la carte microSD de l'appareil, vous devriez consulter le tutoriel sur [the section called "Nettoyage après les démonstrations de construction avec leAWS IoTClient d'appareil" \(p. 190\)](#).

Option 1 : Nettoyage en réécrivant la carte microSD

Le moyen le plus simple et le plus complet de nettoyer la carte microSD après avoir terminé les didacticiels de ce parcours d'apprentissage consiste à remplacer la carte microSD par un fichier image enregistré que vous avez créé lors de la préparation de votre appareil la première fois.

Cette procédure utilise l'ordinateur hôte local pour écrire une image de carte microSD enregistrée sur une carte microSD.

Note

Si votre appareil n'utilise pas de support de stockage amovible pour son système d'exploitation, reportez-vous à la procédure applicable à cet appareil.

Pour écrire une nouvelle image sur la carte microSD

1. Sur votre ordinateur hôte local, localisez l'image de la carte microSD enregistrée que vous souhaitez écrire sur votre carte microSD.
2. Insérez votre carte microSD dans l'ordinateur hôte local.
3. À l'aide d'un outil d'imagerie de carte SD, écrivez le fichier image sélectionné sur la carte microSD.
4. Après avoir écrit l'image Raspberry Pi OS sur la carte microSD, éjectez la carte microSD et retirez-la en toute sécurité de l'ordinateur hôte local.

Votre carte microSD est prête à être utilisée.

Option 2 : Nettoyage en supprimant les répertoires d'utilisateurs

Pour nettoyer la carte microSD après avoir terminé les didacticiels sans réécrire l'image de la carte microSD, vous pouvez supprimer les répertoires utilisateur individuellement. Cela n'est pas aussi complet

que la réécriture de la carte microSD à partir d'une image enregistrée, car elle ne supprime aucun fichier système qui aurait pu être installé.

Si la suppression des répertoires d'utilisateurs est suffisamment approfondie pour répondre à vos besoins, vous pouvez suivre cette procédure.

Pour supprimer les répertoires d'utilisateurs de ce parcours d'apprentissage de votre appareil

1. Exécutez ces commandes pour supprimer les répertoires d'utilisateurs, les sous-répertoires et tous leurs fichiers créés dans ce parcours d'apprentissage, dans la fenêtre du terminal connecté à votre appareil.

Note

Une fois que vous aurez supprimé ces répertoires et fichiers, vous ne pourrez pas exécuter les démonstrations sans avoir terminé à nouveau les tutoriels.

```
rm -Rf ~/dc-configs
rm -Rf ~/policies
rm -Rf ~/messages
rm -Rf ~/certs
rm -Rf ~/.aws-iot-device-client
```

2. Exécutez ces commandes pour supprimer les répertoires et fichiers sources de l'application, dans la fenêtre du terminal connecté à votre appareil.

Note

Ces commandes ne désinstallent aucun programme. Ils suppriment uniquement les fichiers sources utilisés pour les construire et les installer. Après avoir supprimé ces fichiers, le AWS CLet l'AWS IoTDevice Client peut ne pas fonctionner.

```
rm -Rf ~/aws-cli
rm -Rf ~/aws
rm -Rf ~/.aws-iot-device-client
```

Étape 2 : Nettoyage de votre Compte AWS après la création de démonstrations avec le AWS IoT Client d'appareil

Ces procédures vous aident à identifier et supprimer AWSressources que vous avez créées en terminant les didacticiels de ce parcours d'apprentissage.

Nettoyage AWS IoTressources

Cette procédure vous aide à identifier et à supprimer le AWS IoTressources que vous avez créées en terminant les didacticiels de ce parcours d'apprentissage.

AWS IoTressources créées dans ce parcours d'apprentissage

Didacticiel	Ressources d'objet	Ressources de stratégie
the section called "Installation et configuration de l'AWS IoTClient d'appareil" (p. 156)	Chose DevCliest	Stratégie DevClitest Thing
the section called "Démontrer la communication des messages	Subtest Pub	Politique de sous-test Pub

Didacticiel	Ressources d'objet	Ressources de stratégie
MQTT avec leAWS IoTAppareil client" (p. 165)		
the section called "Démontrez des actions à distance (tâches) avecAWS IoTPériphérique Périphérique" (p. 179)	défini par l'utilisateur(il peut y en avoir plus d'un)	défini par l'utilisateur(il peut y en avoir plus d'un)

Pour supprimer leAWS IoT, suivez cette procédure pour chaque ressource objet que vous avez créée

1. Remplacez **thing_name** avec le nom de la ressource objet que vous souhaitez supprimer, puis exécutez cette commande pour répertorier les certificats attachés à la ressource objet, à partir de l'ordinateur hôte local.

```
aws iot list-thing-principals --thing-name thing_name
```

Cette commande renvoie une réponse comme celle-ci qui répertorie les certificats attachés à **thing_name**. Dans la plupart des cas, il n'y aura qu'un seul certificat dans la liste.

```
{
  "principals": [
    "arn:aws:iot:us-
west-2:57EXAMPLE833:cert/23853eea3cf0edc7f8a69c74abeafa27b2b52823cab5b3e156295e94b26ae8ac"
  ]
}
```

2. Pour chaque certificat répertorié par la commande précédente :

- a. Remplacez **certificate_ID** avec l'ID de certificat de la commande précédente. L'ID du certificat est les caractères alphanumériques qui suivent cert/ dans l'ARN renvoyé par la commande précédente. Exécutez ensuite cette commande pour désactiver le certificat.

```
aws iot update-certificate --new-status INACTIVE --certificate-id certificate_ID
```

Si elle aboutit, cette commande ne renvoie rien.

- b. Remplacez **certificate_ARN** avec l'ARN du certificat dans la liste des certificats renvoyée précédemment, puis exécutez cette commande pour répertorier les stratégies attachées à ce certificat.

```
aws iot list-attached-policies --target certificate_ARN
```

Cette commande renvoie une réponse comme celle-ci qui répertorie les stratégies attachées au certificat. Dans la plupart des cas, il n'y aura qu'une seule politique dans la liste.

```
{
  "policies": [
    {
      "policyName": "DevCliTestThingPolicy",
      "policyArn": "arn:aws:iot:us-west-2:57EXAMPLE833:policy/
DevCliTestThingPolicy"
    }
  ]
}
```

```
}
```

- c. Pour chaque stratégie attachée au certificat :

- i. Remplacez *policy_name* avec le policyName valeur de la commande précédente, remplacer *certificate_ARN* avec l'ARN du certificat, puis exécutez cette commande pour détacher la stratégie du certificat.

```
aws iot detach-policy --policy-name policy_name --target certificate_ARN
```

Si elle aboutit, cette commande ne renvoie rien.

- ii. Remplacez *policy_name* avec le policyName, puis exécutez cette commande pour voir si la stratégie est attachée à d'autres certificats.

```
aws iot list-targets-for-policy --policy-name policy_name
```

Si la commande renvoie une liste vide comme celle-ci, la stratégie n'est attachée à aucun certificat et vous continuez à répertorier les versions de stratégie. S'il y a toujours des certificats attachés à la stratégie, continuez avec ledetach-thing-principal étape.

```
{
  "targets": []
}
```

- iii. Remplacez *policy_name* avec le policyName, puis exécutez cette commande pour vérifier les versions de stratégie. Pour supprimer la stratégie, elle ne doit comporter qu'une seule version.

```
aws iot list-policy-versions --policy-name policy_name
```

Si la stratégie ne comporte qu'une seule version, comme cet exemple, vous pouvez passer à la pagedelete-policy puis supprimez la stratégie maintenant.

```
{
  "policyVersions": [
    {
      "versionId": "1",
      "isDefaultVersion": true,
      "createDate": "2021-11-18T01:02:46.778000+00:00"
    }
  ]
}
```

Si la stratégie comporte plusieurs versions, comme dans cet exemple, les versions de stratégie avec un isDefaultVersion valeur de false doit être supprimé avant que la stratégie puisse être supprimée.

```
{
  "policyVersions": [
    {
      "versionId": "2",
      "isDefaultVersion": true,
      "createDate": "2021-11-18T01:52:04.423000+00:00"
    },
    {
      "versionId": "1",
      "isDefaultVersion": false,
```

```
        "createDate": "2021-11-18T01:30:18.083000+00:00"
    ]
}
```

Si vous devez supprimer une version de stratégie, remplacez **policy_name** avec le **policyName** valeur, remplacer **version_ID** avec la **versionId** valeur de la commande précédente, puis exécutez cette commande pour supprimer une version de stratégie.

```
aws iot delete-policy-version --policy-name policy_name --policy-version-id version_ID
```

Si elle aboutit, cette commande ne renvoie rien.

Après avoir supprimé une version de stratégie, répétez cette étape jusqu'à ce que la stratégie ne comporte qu'une seule version de stratégie.

- iv. Remplacez **policy_name** avec **policyName**, puis exécutez cette commande pour supprimer la stratégie.

```
aws iot delete-policy --policy-name policy_name
```

- d. Remplacez **thing_name** par le nom de la chose, remplacez **certificate_ARN** avec l'ARN du certificat, puis exécutez cette commande pour détacher le certificat de la ressource objet.

```
aws iot detach-thing-principal --thing-name thing_name --principal certificate_ARN
```

Si elle aboutit, cette commande ne renvoie rien.

- e. Remplacez **certificate_ID** avec l'ID de certificat de la commande précédente. L'ID du certificat est les caractères alphanumériques qui suivent cert/ dans l'ARN renvoyé par la commande précédente. Exécutez ensuite cette commande pour supprimer la ressource de certificat.

```
aws iot delete-certificate --certificate-id certificate_ID
```

Si elle aboutit, cette commande ne renvoie rien.

3. Remplacez **thing_name** avec le nom de l'objet, puis exécutez cette commande pour supprimer l'objet.

```
aws iot delete-thing --thing-name thing_name
```

Si elle aboutit, cette commande ne renvoie rien.

Nettoyage AWSressources

Cette procédure vous aide à identifier et à supprimer d'autres AWSressources que vous avez créées en terminant les didacticiels de ce parcours d'apprentissage.

Autre AWSressources créées dans ce parcours d'apprentissage

Didacticiel	Type de ressource	Nom ou ID de la ressource
the section called “Démontrez des actions à distance (tâches) avec AWS IoT Périphérique Périphérique” (p. 179)	Objet Amazon S3	hello-world-job.json

Didacticiel	Type de ressource	Nom ou ID de la ressource
<u>the section called “Démontrez des actions à distance (tâches) avec AWS IoT Périphérique” (p. 179)</u>	AWS IoT Ressources pour l'emploi	défini par l'utilisateur

Pour supprimer les AWS Ressources créées dans ce parcours d'apprentissage

1. Pour supprimer les tâches créées dans ce parcours d'apprentissage
 - a. Exécutez cette commande pour répertorier les tâches de votre Compte AWS.

```
aws iot list-jobs
```

La commande renvoie une liste des AWS IoT Tâches dans votre Compte AWS. Elle se présente ainsi.

```
{
    "jobs": [
        {
            "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-2",
            "jobId": "hello-world-job-2",
            "targetSelection": "SNAPSHOT",
            "status": "COMPLETED",
            "createdAt": "2021-11-16T23:40:36.825000+00:00",
            "lastUpdatedAt": "2021-11-16T23:40:41.375000+00:00",
            "completedAt": "2021-11-16T23:40:41.375000+00:00"
        },
        {
            "jobArn": "arn:aws:iot:us-west-2:57EXAMPLE833:job/hello-world-job-1",
            "jobId": "hello-world-job-1",
            "targetSelection": "SNAPSHOT",
            "status": "COMPLETED",
            "createdAt": "2021-11-16T23:35:26.381000+00:00",
            "lastUpdatedAt": "2021-11-16T23:35:29.239000+00:00",
            "completedAt": "2021-11-16T23:35:29.239000+00:00"
        }
    ]
}
```

- b. Pour chaque tâche que vous reconnaissiez dans la liste comme une tâche que vous avez créée dans ce parcours d'apprentissage, remplacez **jobID** avec le jobId de la tâche à supprimer, puis exécutez cette commande pour supprimer un AWS IoT tâche.

```
aws iot delete-job --job-id jobID
```

Si la commande aboutit, elle ne renvoie rien.

2. Pour supprimer les documents de tâche que vous avez stockés dans un compartiment Amazon S3 dans ce parcours d'apprentissage.
 - a. Remplacez **bucket** avec le nom du compartiment que vous avez utilisé, puis exécutez cette commande pour répertorier les objets du compartiment Amazon S3 que vous avez utilisé.

```
aws s3api list-objects --bucket bucket
```

La commande renvoie une liste des objets Amazon S3 dans le compartiment qui ressemble à ceci.

```
{  
    "Contents": [  
        {  
            "Key": "hello-world-job.json",  
            "LastModified": "2021-11-18T03:02:12+00:00",  
            "ETag": "\"868c8bc3f56b5787964764d4b18ed5ef\"",  
            "Size": 54,  
            "StorageClass": "STANDARD",  
            "Owner": {  
                "DisplayName": "EXAMPLE",  
                "ID": "  
e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"  
            }  
        },  
        {  
            "Key": "iot_job_firmware_update.json",  
            "LastModified": "2021-04-13T21:57:07+00:00",  
            "ETag": "\"7c68c591949391791ecf625253658c61\"",  
            "Size": 66,  
            "StorageClass": "STANDARD",  
            "Owner": {  
                "DisplayName": "EXAMPLE",  
                "ID": "  
e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"  
            }  
        },  
        {  
            "Key": "order66.json",  
            "LastModified": "2021-04-13T21:57:07+00:00",  
            "ETag": "\"bca60d5380b88e1a70cc27d321cab72\"",  
            "Size": 29,  
            "StorageClass": "STANDARD",  
            "Owner": {  
                "DisplayName": "EXAMPLE",  
                "ID": "  
e9e3d6ec1EXAMPLEf5bfb5e6bd0a2b6ed03884d1ed392a82ad011c144736a4ee"  
            }  
        }  
    ]  
}
```

- b. Pour chaque objet que vous reconnaissiez dans la liste comme un objet que vous avez créé dans ce parcours d'apprentissage, remplacez **bucket** avec le nom de compartiment et **key** avec la valeur de clé de l'objet à supprimer, puis exécutez cette commande pour supprimer un objet Amazon S3.

```
aws s3api delete-object --bucket bucket --key key
```

Si la commande aboutit, elle ne renvoie rien.

Une fois que vous avez supprimé tous les AWSressources et objets que vous avez créés lorsque vous terminez ce parcours d'apprentissage, vous pouvez recommencer et répéter les didacticiels.

Construire des solutions avec leAWS IoTKits SDK pour les appareils

Les didacticiels de cette section vous expliquent les étapes à suivre pour développer une solution IoT pouvant être déployée dans un environnement de production à l'aide de AWS IoT.

Ces tutoriels peuvent prendre plus de temps que ceux de la section sur [the section called “Construire des démonstrations avec leAWS IoTClient d'appareil” \(p. 143\)](#) parce qu'ils utilisent le AWS IoT Dispositif SDK et expliquez plus en détail les concepts appliqués pour vous aider à créer des solutions sécurisées et fiables.

Commencez à créer des solutions avec leAWS IoTKits SDK pour les appareils

Ces tutoriels vous guident à travers différents AWS IoT hypothétiques. Le cas échéant, les didacticiels utilisent l' AWS IoTKits SDK pour les appareils.

Rubriques

- [Tutorial : Connexion d'un appareil à AWS IoT Core l'aide du SDK de l'AWS IoT appareil \(p. 196\)](#)
- [Création de AWS IoT règles pour acheminer les données de l'appareil vers d'autres services \(p. 213\)](#)
- [Conservation de l'état de l'appareil lorsque l'appareil est hors connexion avec Device Shadows \(p. 244\)](#)
- [Didacticiel : Création d'un outil d'autorisation personnalisé pour AWS IoT Core \(p. 265\)](#)
- [Tutorial : Surveillance de l'humidité du sol avec un AWS IoT Raspberry Pi \(p. 277\)](#)

Tutoriel : Connexion d'un appareil à AWS IoT Core l'aide du SDK de l'AWS IoT appareil

Ce didacticiel explique comment connecter un appareil AWS IoT Core afin qu'il puisse envoyer et recevoir des données depuis et vers AWS IoT. Une fois ce didacticiel terminé, votre appareil sera configuré pour se connecter à AWS IoT Core et vous comprendrez comment les appareils communiquent avec AWS IoT.

Dans ce didacticiel, vous allez :

1. [the section called “Préparez votre appareil pour AWS IoT” \(p. 197\)](#)
2. [the section called “Vérifier le protocole MSON” \(p. 197\)](#)
3. [the section called “Consultez l'exemple d'application pubsub.py Device SDK” \(p. 198\)](#)
4. [the section called “Connectez votre appareil et communiquez avec AWS IoT Core” \(p. 204\)](#)
5. [the section called “Vérifier les résultats” \(p. 209\)](#)

Ce tutoriel vous prendra environ une heure.

Avant de commencer ce tutoriel, assurez-vous de disposer des éléments suivants :

- Terminé [Démarrer avec AWS IoT Core \(p. 18\)](#)

Dans la section de ce didacticiel où vous devez le faire [the section called “Configurer votre appareil” \(p. 43\)](#), sélectionnez l'[the section called “Connect un Raspberry Pi ou un autre](#)

[appareil](#) (p. 61) option correspondant à votre appareil et utilisez les options du langage Python pour configurer votre appareil.

Gardez ouverte la fenêtre de terminal que vous utilisez dans ce didacticiel, car vous l'utiliserez également dans ce didacticiel.

- Un appareil capable d'exécuter le AWS IoT Device SDK v2 pour Python.

Ce didacticiel explique comment connecter un appareil à l'aide AWS IoT Core d'exemples de code Python, qui nécessitent un appareil relativement puissant.

Si vous travaillez avec des appareils dont les ressources sont limitées, ces exemples de code risquent de ne pas fonctionner sur eux. Dans ce cas, vous aurez peut-être plus de succès grâce au [the section called "Utilisation de l'Kit SDK des appareils AWS IoT pour Embedded C" \(p. 210\)](#) tutoriel.

Préparez votre appareil pour AWS IoT

Dans [Démarrer avec AWS IoT Core \(p. 18\)](#), vous avez préparé votre appareil et votre AWS compte afin qu'ils puissent communiquer. Cette section passe en revue les aspects de cette préparation qui s'appliquent à tout appareil connecté à AWS IoT Core.

Pour connecter un appareil à AWS IoT Core :

1. Vous devez avoir un Compte AWS.

La procédure décrite [Configurez votre Compte AWS \(p. 19\)](#) décrit comment créer un Compte AWS si vous n'en avez pas déjà un.

2. Dans ce compte, les AWS IoT ressources suivantes doivent être définies pour l'appareil de votre région Compte AWS et.

La procédure décrite [Création de AWS IoT ressources \(p. 40\)](#) décrit comment créer ces ressources pour l'appareil de votre région Compte AWS et de votre région.

- Un certificat d'appareil enregistré AWS IoT et activé pour authentifier l'appareil.

Le certificat est souvent créé avec et attaché à un AWS IoT objet quelconque. Bien qu'un objet ne soit pas nécessaire à la connexion d'un appareil AWS IoT, il met des AWS IoT fonctionnalités supplémentaires à la disposition de l'appareil.

- Une politique attachée au certificat de l'appareil qui l'autorise à se connecter AWS IoT Core et à effectuer toutes les actions que vous souhaitez.

3. Une connexion Internet qui permet d'accéder aux terminaux Compte AWS de votre appareil.

Les points de terminaison de l'appareil sont décrits [AWS IoT données de l'appareil et points de terminaison de service \(p. 85\)](#) et peuvent être consultés sur la [page des paramètres de la AWS IoT console](#).

4. Des logiciels de communication tels que les SDK de l'AWS IoT appareil sont fournis. Ce didacticiel utilise le [AWS IoT Device SDK v2 pour Python](#).

Vérifier le protocole MSON

Avant de parler de l'exemple d'application, il est utile de 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 entre MQTT et HTTP, consultez [Choix d'un protocole pour la communication de votre appareil \(p. 90\)](#).

MQTT utilise un modèle de communication publié/abonnement

Le protocole MQTT utilise un modèle de communication publié/abonnement avec son hôte. Ce modèle est différent du modèle de demande/réponse utilisé par HTTP. Avec MQTT, les appareils établissent une session avec l'hôte identifié par un identifiant client unique. Pour envoyer des données, les appareils publient des messages identifiés par sujets vers un courtier de messages de l'hôte. Pour recevoir des messages du courtier de messages, les appareils s'abonnent à des sujets en envoyant des filtres de sujet dans les demandes d'abonnement au courtier de messages.

MQTT prend en charge les sessions persistantes

Le courtier de messages reçoit les messages des appareils et publie des messages vers les appareils qui y sont abonnés. Grâce aux [sessions persistantes \(p. 94\)](#), c'est-à-dire des sessions qui restent actives même lorsque l'appareil initiateur est déconnecté, les appareils peuvent récupérer les messages publiés alors qu'ils étaient déconnectés. Du côté de l'appareil, MQTT prend en charge les niveaux de qualité de service ([QoS \(p. 94\)](#)) qui garantissent que l'hôte reçoit les messages envoyés par l'appareil.

Consultez l'exemple d'application pubsub.py Device SDK

Cette section passe en revue l'pubsub.py exemple d'application du AWS IoTDevice SDK v2 pour Python utilisé dans ce didacticiel. Ici, nous allons voir comment il se connecteAWS IoT Core pour publier des messages MQTT et s'y abonner. La section suivante présente des exercices qui vous aideront à découvrir comment un appareil se connecte et communique avecAWS IoT Core.

L'pubsub.py exemple d'application montre les aspects suivants d'une connexion MQTT avecAWS IoT Core :

- [Protocoles de communication \(p. 198\)](#)
- [Sessions persistantes \(p. 201\)](#)
- [Qualité de service \(p. 201\)](#)
- [Publier un message \(p. 202\)](#)
- [Abonnement aux messages \(p. 203\)](#)
- [Déconnexion et reconnexion de l'appareil \(p. 204\)](#)

Protocoles de communication

L'pubsub.py exemple montre une connexion MQTT utilisant les protocoles MQTT et MQTT over WSS. La bibliothèque [AWSCRT \(AWSCommon Runtime\)](#) prend en charge les protocoles de communication de bas niveau et est incluse dans le AWS IoT Device SDK v2 pour Python.

MQTT

Lespubsub.py exemples d'appelsmtls_from_path (présentés ici) dans le [mqtt_connection_builder](#) pour établir une connexion à AWS IoT Core l'aide du protocole MQTT. mtls_from_path utilise des certificats X.509 et TLS v1.2 pour authentifier l'appareil. La bibliothèque AWS 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.ca_file,  
    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

Le pointCompte AWS de terminaison de votre appareil IoT

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

cert_filepath

Chemin d'accès au fichier de certificat de l'appareil

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

pri_key_filepath

Le chemin d'accès au fichier de clé privée de l'appareil qui a été créé avec son fichier de certificat

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

ca_filepath

Le chemin d'accès au fichier Root CA. Obligatoire uniquement si le serveur MQTT utilise un certificat qui ne figure pas déjà dans votre trust store.

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

client_bootstrap

L'objet d'exécution commun qui gère les activités de communication avec les sockets

Dans l'exemple d'application, cet objet est instancié avant l'appel
`à mqtt_connection_builder.mtls_from_path`.

on_connection_interrupted, on_connection_resumed

Les fonctions de rappel pour appeler lorsque la connexion de l'appareil est interrompue et reprise

client_id

L'identifiant qui identifie de manière unique cet appareil dans la région AWS

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

clean_session

S'il faut démarrer une nouvelle session persistante ou, s'il y en a une, se reconnecter à une session existante

keep_alive_secs

La valeur Keep Alive, en secondes, à envoyer dans la CONNECT demande. Un ping sera automatiquement envoyé à cet intervalle. Si le serveur ne reçoit pas de ping après 1,5 fois cette valeur, il suppose que la connexion est perdue.

MSON sur WSON

Les `pubsub.py` exemples d'appels `websockets_with_default_aws_signing` (présentés ici) dans le [mqtt_connection_builder](#) pour établir une connexion à AWS IoT Core l'aide du protocole MQTT via WSS. `websockets_with_default_aws_signing` crée une connexion MQTT via WSS à l'aide de [Signature V4](#) pour 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.ca_file,
        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

Le pointCompte AWS de terminaison de votre appareil IoT

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

client_bootstrap

L'objet d'exécution commun qui gère les activités de communication avec les sockets

Dans l'exemple d'application, cet objet est instancié avant l'appel
à `mqtt_connection_builder.websockets_with_default_aws_signing`.

region

Région deAWS signature utilisée par l'authentification Signature V4. Danspubsub .py, il transmet le paramètre saisi dans la ligne de commande.

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

credentials_provider

LesAWS informations d'identification fournies à utiliser pour l'authentification

Dans l'exemple d'application, cet objet est instancié 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 avant l'appel
à `mqtt_connection_builder.websockets_with_default_aws_signing`.

ca_filepath

Le chemin d'accès au fichier Root CA. Obligatoire uniquement si le serveur MQTT utilise un certificat qui ne figure pas déjà dans votre trust store.

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

on_connection_interrupted, on_connection_resumed

Les fonctions de rappel pour appeler lorsque la connexion de l'appareil est interrompue et reprise

client_id

L'identifiant qui identifie de manière unique cet appareil dans leRégion AWS.

Dans l'exemple d'application, cette valeur est transmise depuis la ligne de commande.

clean_session

S'il faut démarrer une nouvelle session persistante ou, s'il y en a une, se reconnecter à une session existante

keep_alive_secs

La valeur Keep Alive, en secondes, à envoyer dans laCONNECT demande. Un ping sera automatiquement envoyé à cet intervalle. Si le serveur ne reçoit pas de ping après 1,5 fois cette valeur, il suppose que la connexion est perdue.

HTTPS

Qu'en est-il du protocole HTTPS ? AWS IoT Core prend en charge les appareils qui publient des requêtes HTTPS. Du point de vue de la programmation, les appareils envoient des requêtes HTTPSAWS IoT Core comme n'importe quelle autre application. Pour un exemple de programme Python qui envoie un message HTTP depuis un appareil, consultez [l'exemple de code HTTPS \(p. 112\)](#) utilisant la `requests` bibliothèque Python. Cet exemple envoie un message à AWS IoT Core l'aide du protocole HTTPS afin qu'il soit AWS IoT Core interprété comme un message MQTT.

Bien que AWS IoT Core les demandes HTTPS soient prises en charge depuis des appareils, assurez-vous de consulter les informations vous concernant [Choix d'un protocole pour la communication de votre appareil \(p. 90\)](#) afin de pouvoir prendre une décision éclairée sur le protocole à utiliser pour les communications de votre appareil.

Sessions persistantes

Dans l'exemple d'application, définir le `clean_session` paramètre sur `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, s'il en existe une. Sinon, il crée une nouvelle session persistante et s'y connecte.

Dans le cas d'une session persistante, les messages envoyés à l'appareil sont stockés par le courtier de messages lorsque l'appareil n'est pas connecté. Lorsqu'un appareil 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 permanente, l'appareil ne recevra pas les messages envoyés alors qu'il n'est pas connecté. L'option à utiliser dépend de votre application et de la nécessité ou non de communiquer les messages qui apparaissent alors qu'un appareil n'est pas connecté. Pour plus d'informations, veuillez consulter [Sessions permanentes MQTT \(p. 94\)](#).

Qualité de service

Lorsque l'appareil publie des messages et s'y abonne, la qualité de service (QoS) préférée peut être définie. AWS IoT prend en charge les niveaux de QoS 0 et 1 pour les opérations de publication et d'abonnement. Pour plus d'informations sur les niveaux de QoS dans AWS IoT, consultez [Options de qualité de service \(QoS\) MQTT \(p. 94\)](#).

Le moteur d'exécution AWS CRT pour Python définit ces constantes pour les niveaux de QoS qu'il prend en charge :

Niveaux de qualité de service en Python

Niveau QoS MQTT	Valeur symbolique Python utilisée par le SDK	Description
Niveau QoS service utile 0	<code>mqtt.QoS.AT_MOST_ONCE</code>	Une seule tentative d'envoi du message sera effectuée, qu'il soit

Niveau QoS MQTT	Valeur symbolique Python utilisée par le SDK	Description
		reçu ou non. Le message peut ne pas être envoyé du tout, par exemple, si l'appareil 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 accusé de réception soit reçu.

Dans l'exemple d'application, les demandes de publication et d'abonnement sont effectuées avec un niveau de QoS de 1 (mqtt.QoS.AT_LEAST_ONCE).

- **QoS de la publication**

Lorsqu'un appareil publie un message avec un niveau de QoS 1, il envoie le message à plusieurs reprises jusqu'à ce qu'il reçoive une PUBACK réponse du courtier de messages. Si l'appareil n'est pas connecté, le message est mis en file d'attente pour être envoyé après sa reconnexion.

- **QoS de l'abonnement**

Lorsqu'un appareil s'abonne à un message de niveau de QoS 1, le courtier de messages enregistre les messages auxquels l'appareil est abonné jusqu'à ce qu'ils puissent lui être envoyés. Le courtier de messages renvoie les messages jusqu'à ce qu'il reçoive une PUBACK réponse du terminal.

Publier un message

Une fois la connexion établie avec succès AWS IoT Core, les appareils peuvent publier des messages. Pour ce faire, l'`pubsub.py` exemple appelle le `publish` fonctionnement de l'`mqtt_connection` objet.

```
mqtt_connection.publish(
    topic=args.topic,
    payload=message,
    qos= mqtt.QoS.AT_LEAST_ONCE
)
```

topic

Le nom du sujet du message qui identifie le message

Dans l'exemple d'application, cela est transmis depuis la ligne de commande.

payload

La charge utile du message formatée sous forme de chaîne (par exemple, un document JSON)

Dans l'exemple d'application, cela est transmis depuis la ligne de commande.

Un document JSON est un format de charge utile courant, reconnu par d'autres AWS IoT services ; toutefois, le format de données de la charge utile du message peut être celui sur lequel les éditeurs et les abonnés se mettent d'accord. D'autres AWS IoT services, cependant, ne reconnaissent que le JSON et le CBOR, dans certains cas, pour la plupart des opérations.

qos

Le niveau de QoS de ce message

Abonnement aux messages

Pour recevoir des messages provenant AWS IoT d'autres services et appareils, les appareils s'abonnent à ces messages par le nom de leur sujet. Les appareils peuvent s'abonner à des messages individuels en spécifiant un [nom de sujet \(p. 115\)](#), et à un groupe de messages en spécifiant un [filtre de sujet \(p. 116\)](#), qui peut inclure des caractères génériques. L'pubsub.py exemple utilise le code affiché ici pour s'abonner aux messages et enregistrer les fonctions de rappel afin de traiter le message une fois qu'il a été reçu.

```
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 depuis la ligne de commande.

qos

Si le courtier de messages doit stocker ces messages lorsque l'appareil est déconnecté.

Une valeur de `mqtt.QoS.AT_LEAST_ONCE` (niveau de QoS 1) nécessite la spécification d'une session persistante (`clean_session=False`) lors de la création de la connexion.

callback

La fonction à appeler pour traiter le message souscrit.

La `mqtt_connection.subscribe` fonction renvoie un future et un identifiant 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 callback de l'pubsub.py exemple traite les messages souscrits au fur et à mesure que l'appareil 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 êtes abonné à un filtre de sujet.

payload

Charge utile des messages

Le format utilisé est spécifique à l'application.

kwargs

Arguments supplémentaires possibles, tels que décrits dans [mqtt.Connection.subscribe](#).

Dans l'`pubsub.py` exemple, affiche `_message_received` uniquement la rubrique et sa charge utile. Il compte également les messages reçus pour terminer le programme une fois la limite atteinte.

Votre application évaluera le sujet et la charge utile afin de déterminer les actions à effectuer.

Déconnexion et reconnexion de l'appareil

L'`pubsub.py` exemple inclut des fonctions de rappel qui sont appelées lorsque le périphérique est déconnecté et lorsque la connexion est rétablie. Les actions entreprises par votre appareil lors de ces événements sont spécifiques à l'application.

Lorsqu'un appareil se connecte pour la première fois, il doit s'abonner à des sujets pour les 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 l'appareil est déconnecté trop longtemps.

Connectez votre appareil et communiquez avec AWS IoT Core

Cette section présente des exercices qui vous aideront à explorer les différents aspects de la connexion de votre appareil à AWS IoT Core. Pour ces exercices, vous allez utiliser le [client de test MQTT](#) de la AWS IoT console pour voir ce que votre appareil publie et pour publier des messages sur votre appareil. Ces exercices utilisent l'`pubsub.py` exemple du [AWS IoT Device SDK v2 pour Python](#) et s'appuient sur votre expérience à l'aide de [Démarrer avec AWS IoT Core \(p. 18\)](#) didacticiels.

Dans cette section, vous allez :

- [Abonnez-vous aux filtres thématiques génériques \(p. 204\)](#)
- [Traiter les abonnements aux filtres thématiques \(p. 206\)](#)
- [Publier des messages depuis votre appareil \(p. 207\)](#)

Pour ces exercices, vous allez partir de l'`pubsub.py` exemple de programme.

Note

Ces exercices supposent que vous avez suivi les [Démarrer avec AWS IoT Core \(p. 18\)](#) didacticiels et que vous avez utilisé la fenêtre de terminal de votre appareil à partir de ce didacticiel.

Abonnez-vous aux filtres thématiques génériques

Dans cet exercice, vous allez modifier la ligne de commande utilisée pour appeler `pubsub.py` afin de vous abonner à un filtre thématique générique et traiter les messages reçus en fonction du sujet du message.

Procédure d'

Pour cet exercice, imaginez que votre appareil contient un contrôle de température et un contrôle de lumière. Il utilise ces noms de rubriques pour identifier les messages les concernant.

1. Avant de commencer l'exercice, essayez d'exécuter cette commande à partir des [Démarrer avec AWS IoT Core \(p. 18\)](#) didacticiels de 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 --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Vous devriez obtenir le même résultat que celui que vous avez vu dans le [didacticiel de démarrage \(p. 65\)](#).

2. Pour cet exercice, modifiez ces paramètres de ligne de commande.

Action	Paramètre de ligne de commande	Effet
ajouter	--message ""	Configurer pubsub.py pour écouter uniquement
ajouter	--count 2	Terminez le programme après avoir reçu deux messages
modification	--topic device/+/details	Définissez le filtre de sujet auquel vous souhaitez vous abonner

L'application de ces modifications à la ligne de commande initiale entraîne cette ligne de commande. Entrez cette commande dans la fenêtre du terminal de votre appareil.

```
python3 pubsub.py --message "" --count 2 --topic device/+/details --ca_file ~/certs/Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key --endpoint your-iot-endpoint
```

Le programme doit afficher ceci :

```
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 ça sur votre terminal, cela signifie que votre appareil est prêt et écoute les messages dont les noms des sujets commencent par device et se terminent par /detail. Alors, testons cela.

3. Voici quelques messages que votre appareil est susceptible de recevoir.

Nom de la rubrique	Charge utile des messages
device/temp/details	{ "desiredTemp": 20, "currentTemp": 15 }
device/light/details	{ "desiredLight": 100, "currentLight": 50 }

4. À l'aide du client de test MQTT de la AWS IoT console, envoyez les messages décrits à l'étape précédente à votre appareil.

- a. Ouvrez le [client de test MQTT](#) dans la AWS IoT console.

- b. Dans S'abonner à une rubrique, dans le champ Rubrique d'abonnement, saisissez le filtre de rubrique :**device/+details**, puis choisissez S'abonner à la rubrique.
- c. Dans la colonne Abonnements du client de test MQTT, choisissez device/+details.
- d. Pour chacune des rubriques du tableau précédent, procédez comme suit dans le client de test MQTT :
 1. Dans Publier, entrez la valeur de la colonne Nom du sujet dans le tableau.
 2. Dans le champ de charge utile du message situé sous le nom de la rubrique, entrez la valeur de la colonne Charge utile du message du tableau.
 3. Regardez la fenêtre du terminal dans laquellepubsub.py s'exécute et, dans le client de test MQTT, choisissez Publier dans la rubrique.

Vous devriez voir que le message a été récupubsub.py dans la fenêtre du terminal.

Résultat de l'exercice

Ainsi pubsub.py s'est abonné aux messages à l'aide d'un filtre thématique générique, les a reçus et les a affichés dans la fenêtre du terminal. Remarquez que vous êtes abonné à un filtre de rubrique unique et que la fonction de rappel a été appelée pour traiter les messages comportant deux sujets distincts.

Traiter les abonnements aux filtres thématiques

Sur la base de l'exercice précédent, modifiez l'pubsub.py exemple d'application pour évaluer les sujets des messages et traiter les messages auxquels vous êtes abonné en fonction du sujet.

Procédure d'

Pour évaluer le sujet du message

1. Copiez pubsub.py dans pubsub2.py.
2. Commencez pubsub2.py par ouvrir dans votre éditeur de texte ou IDE favori.
3. Dans pubsub2.py, recherchez laon_message_received fonction.
4. Dans on_message_received, insérez le code suivant après la ligne commençant par print("Received message et avant la ligne commençant 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 --ca_file ~/certs/
Amazon-root-CA-1.pem --cert ~/certs/device.pem.crt --key ~/certs/private.pem.key --
endpoint your-iot-endpoint
```

6. Dans la AWS IoT console, ouvrez le [client de test MQTT](#).
7. Dans S'abonner à une rubrique, dans le champ Rubrique d'abonnement, saisissez le filtre de rubrique **:device/+/details**, puis choisissez S'abonner à la rubrique.
8. Dans la colonne Abonnements du client de test MQTT, choisissez device/+/details.
9. Pour chacune des rubriques de ce tableau, procédez comme suit dans le client de test MQTT :

Nom de la rubrique	Charge utile des messages
device/temp/details	{ "desiredTemp": 20, "currentTemp": 15 }
device/light/details	{ "desiredLight": 100, "currentLight": 50 }

1. Dans Publier, entrez la valeur de la colonne Nom du sujet dans le tableau.
2. Dans le champ de charge utile du message situé sous le nom de la rubrique, entrez la valeur de la colonne Charge utile du message du tableau.
3. Regardez la fenêtre du terminal dans laquellepubsub.py s'exécute et, dans le client de test MQTT, choisissez Publier dans la rubrique.

Vous devriez voir que le message a été récupubsub.py dans la fenêtre du terminal.

Vous devez voir ceci 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'exemple d'application reconnaissse et traite plusieurs messages dans la fonction de rappel. Ainsi, votre appareil peut recevoir des messages et agir en conséquence.

Un autre moyen pour votre appareil de recevoir et de traiter plusieurs messages consiste à s'abonner à différents messages séparément et à attribuer à chaque abonnement sa propre fonction de rappel.

Publier des messages depuis votre appareil

Vous pouvez utiliser l'exemple d'application pubsub.py pour publier des messages depuis 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 afin de pouvoir publier des documents JSON dans la charge utile des messages qui peuvent être lus par AWS IoT Core.

Procédure d'

Dans cet exercice, le message suivant sera envoyé avec le device/data sujet.

```
{  
    "timestamp": 1601048303,  
    "sensorId": 28,  
    "sensorData": [  
        {  
            "sensorName": "Wind speed",  
            "sensorValue": 34.2211224  
        }  
    ]  
}
```

Pour préparer votre client de test MQTT à surveiller les messages issus de cet exercice

1. Dans S'abonner à une rubrique, dans le champ Rubrique d'abonnement, saisissez le filtre de rubrique :**device/data**, puis choisissez S'abonner à la rubrique.
2. Dans la colonne Abonnements du client de test MQTT, choisissez appareil/données.
3. Gardez 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-`lepubsub.py` vers `pubsub3.py`.
2. Modifiez `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. Localisez cette ligne de code :

```
message = "{} [{}].format(message_string, publish_count)
```
 - c. Modifiez-le comme suit :

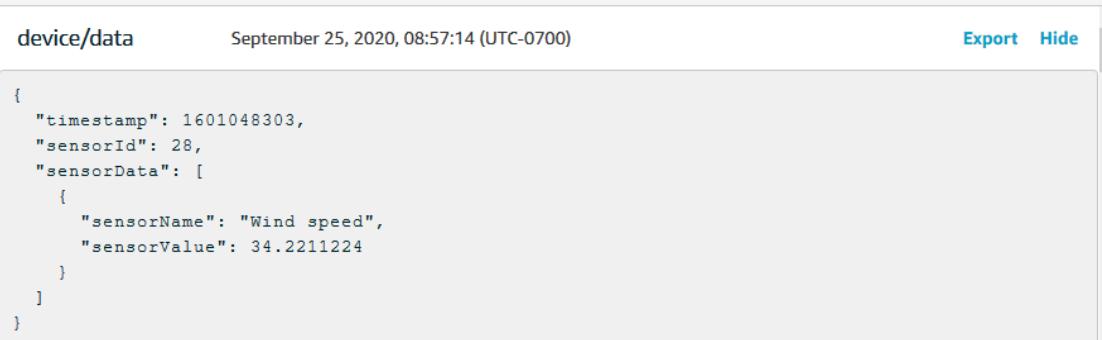
```
message = "{}".format(message_string)
```
 - d. Localisez cette ligne de code :

```
message_json = json.dumps(message)
```
 - e. Modifiez-le comme suit :

```
message = "{}.json.dumps(json.loads(message))
```
 - f. Enregistrez vos modifications.
3. Sur votre appareil, exécutez cette commande pour envoyer le message deux fois.

```
python3 pubsub3.py --ca_file ~/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 suit :



```
{  
    "timestamp": 1601048303,  
    "sensorId": 28,  
    "sensorData": [  
        {  
            "sensorName": "Wind speed",  
            "sensorValue": 34.2211224  
        }  
    ]  
}
```

Par défaut, il s'abonne `pubsub3.py` 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 doit se présenter comme suit.

```
Connecting to a3qEXAMPLEsffp-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

Ainsi, votre appareil peut générer des messages à envoyer AWS IoT Core pour tester la connectivité de base et fournir des messages AWS IoT Core à traiter. Par exemple, vous pouvez utiliser cette application pour envoyer des données de test depuis votre appareil afin de tester les actions des AWS règles.

Vérifier les résultats

Les exemples de ce didacticiel vous ont permis de découvrir les bases de la communication entre les appareils AWS IoT Core, un élément fondamental de votre AWS IoT solution. Lorsque vos appareils sont en mesure de communiquer avec eux AWS IoT Core, ils peuvent transmettre des messages à des AWS services et à d'autres appareils sur lesquels ils peuvent agir. De même, AWS les services et autres appareils peuvent traiter des informations qui se traduisent par des messages renvoyés à vos appareils.

Lorsque vous serez prêt à AWS IoT Core approfondir votre exploration, essayez ces didacticiels :

- [the section called “Envoi d'une notification Amazon SNS” \(p. 221\)](#)
- [the section called “Stockage des données de l'appareil dans une table DynamoDB” \(p. 229\)](#)
- [the section called “Formatage d'une notification à l'aide d'une AWS Lambda fonction” \(p. 235\)](#)

Didacticiel : Utilisation de l'Kit SDK des appareils AWS IoT pour Embedded C

Cette section décrit comment exécuter le Kit SDK des appareils AWS IoT pour Embedded C.

Procédures de cette section

- [Étape 1 : Installation de l'Kit SDK des appareils AWS IoT pour Embedded C \(p. 210\)](#)
- [Étape 2 : Configuration de l'exemple d'application \(p. 210\)](#)
- [Étape 3 : Créer et exécuter l'exemple d'application \(p. 212\)](#)

Étape 1 : Installation de l'Kit SDK des appareils AWS IoT pour Embedded C

Le Kit SDK des appareils AWS IoT pour 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 plus grands AWS IoT SDK pour appareils et mobiles (par exemple, C++, Java, JavaScript et Python).

En général, le Kit SDK des appareils AWS IoT pour 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 Kit SDK des appareils AWS IoT pour Embedded C vers votre appareil depuis [GitHub](#).

```
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 accédez à la dernière version. Veuillez consulter [github.com/aws/aws-iot-device-SDK-Embedded-C/Tags](#) pour la dernière version de la balise.

```
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
```

Étape 2 : Configuration de l'exemple d'application

Le Kit SDK des appareils AWS IoT pour Embedded C inclut des exemples d'applications que vous pouvez essayer. Pour des raisons de simplicité, ce didacticiel utilise `mqtt_demo_mutual_auth`, qui montre comment se connecter à l'AWS IoT Core Courtier de messages, abonnez-vous à des rubriques MQTT et y publiez.

1. Copiez le certificat et la clé privée que vous avez créés dans [Démarrer avec AWS IoT Core \(p. 18\)](#) dans `lebuild/bin/certificates.directory`.

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.

- 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 la AWS CLI installé, vous pouvez utiliser cette commande pour trouver l'URL du point de terminaison de votre compte.

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

Si vous n'avez pas installé la 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 choisissez Interagir. Votre point de terminaison s'affiche dans la section HTTPS de la page de détails de l'objet.

- Ouverture d'`demo_config.h` et mettez à jour les valeurs pour :

POINT DE TERMINAISON AWS_IOT_

Votre point de terminaison personnel.

CHEMIN CLIENT_CERT_PATH

Le chemin d'accès de votre fichier de certificat, par exemple `certificates/device.pem.crt`.

CHEMIN CLIENT_PRIVATE_KEY_PATH

Le nom de fichier de votre clé privée, par exemple `certificates/private.pem.key`.

Par exemple :

```
// 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"
// =====
```

- 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 --version` 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.

Étape 3 : Créer et exécuter l'exemple d'application

Pour exécuter les exemples d'applications du Kit SDK des appareils AWS IoT pour Embedded C

1. Accédez à aws-iot-device-sdk-embedded-c et créez un répertoire de génération.

```
mkdir build && cd build
```

2. Entrez la commande CMake suivante pour générer les Makefiles nécessaires à la création.

```
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 :

```
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:584] Establishing a TLS session to a2zk5tjv9x07ct-ats.iot.us-west-2.amazonaws.com:8883.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1264] Creating an MQTT connection to a2zk5tjv9x07ct-ats.iot.us-west-2.amazonaws.com.  
[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=2.  
[INFO] [MQTT] [core_mqtt_serializer.c:970] CONNACK session present bit not set.  
[INFO] [MQTT] [core_mqtt_serializer.c:912] Connection accepted.  
[INFO] [MQTT] [core_mqtt.c:1526] Received MQTT CONNACK successfully from broker.  
[INFO] [MQTT] [core_mqtt.c:1792] MQTT connection established with the broker.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1033] MQTT connection successfully established with broker.  
  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1296] A clean MQTT connection is established. Cleaning up all the stored outgoing publishes.  
  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1314] Subscribing to the MQTT topic testclient/example/topic.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1097] SUBSCRIBE sent for topic testclient/example/topic to broker.  
  
[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=3.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:921] Subscribed to the topic testclient/example/topic. with maximum QoS 1.  
  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1358] Sending Publish to the MQTT topic testclient/example/topic.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:1195] PUBLISH sent for topic testclient/example/topic to broker with packet ID 2.  
  
[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=2.  
[INFO] [MQTT] [core_mqtt.c:1126] Ack packet deserialized with result: MQTTSuccess.  
[INFO] [MQTT] [core_mqtt.c:1139] State record updated. New state=MQTTPublishDone.  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:946] PUBACK received for packet id 2.  
  
[INFO] [DEMO] [mqtt_demo_mutual_auth.c:672] Cleaned up outgoing publish packet with packet id 2.  
  
[INFO] [MQTT] [core_mqtt.c:855] Packet received. ReceivedBytes=40.  
[INFO] [MQTT] [core_mqtt.c:1015] De-serialized incoming PUBLISH packet: DeserializerResult=MQTTSuccess.
```

Votre appareil est maintenant connecté à AWS IoT Utilisation de l'Kit SDK des appareils AWS IoT pour Embedded C.

Vous pouvez également utiliser l'AWS IoT pour 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" \(p. 71\)](#).

Création de AWS IoT règles pour acheminer les données de l'appareil vers d'autres services

Ces didacticiels vous montrent comment créer et tester des AWS IoT règles à l'aide de certaines des actions les plus courantes.

AWS IoT Les règles envoient des données depuis vos appareils vers d'autres AWS services. Ils écoutent des messages MQTT spécifiques, formatent les données des charges utiles des messages et envoient le résultat à d'autres AWS services.

Nous vous recommandons de les essayer dans l'ordre dans lequel ils apparaissent ici, même si votre objectif est de créer une règle utilisant une fonction Lambda ou quelque chose de plus complexe. Les didacticiels sont présentés dans l'ordre du plus simple au plus complexe. Ils présentent de nouveaux concepts de manière progressive pour vous aider à apprendre les concepts que vous pouvez utiliser pour créer les actions de règles qui ne font pas l'objet d'un didacticiel spécifique.

Note

AWS IoT Les règles vous aident à envoyer les données de vos appareils IoT vers d'autres AWS services. Pour y parvenir, vous devez toutefois avoir une connaissance pratique des autres services auxquels vous souhaitez envoyer des données. Bien que ces didacticiels fournissent les informations nécessaires pour effectuer les tâches, vous trouverez peut-être utile d'en savoir plus sur les services auxquels vous souhaitez envoyer des données avant de les utiliser dans votre solution. Une explication détaillée des autres AWS services n'entre pas dans le cadre de ces didacticiels.

Présentation du scénario du didacticiel

Le scénario de ces didacticiels est celui d'un capteur météorologique qui publie périodiquement ses données. Il existe de nombreux capteurs de ce type dans ce système imaginaire. Les didacticiels de cette section se concentrent toutefois sur un seul appareil tout en montrant comment vous pouvez intégrer plusieurs capteurs.

Les didacticiels de cette section vous montrent comment utiliser AWS IoT les règles pour effectuer les tâches suivantes avec ce système imaginaire de capteurs météorologiques.

- [Tutoriel : Republication d'un message MQTT \(p. 215\)](#)

Ce didacticiel explique comment republier un message MQTT reçu des capteurs météorologiques sous la forme d'un message contenant uniquement l'identifiant du capteur et la valeur de température. Il utilise uniquement AWS IoT Core des services et montre une requête SQL simple et explique comment utiliser le client MQTT pour tester votre règle.

- [Tutoriel : Envoi d'une notification Amazon SNS \(p. 221\)](#)

Ce didacticiel explique comment envoyer un message SNS lorsqu'une valeur provenant d'un capteur météorologique dépasse une valeur spécifique. Il s'appuie sur les concepts présentés dans le didacticiel précédent et explique comment travailler avec un autre AWS service, [Amazon Simple Notification Service \(Amazon SNS\)](#).

Si vous utilisez Amazon SNS pour la première fois, consultez ses exercices de [mise en](#) route avant de commencer ce didacticiel.

- [Tutoriel : Stockage des données de l'appareil dans une table DynamoDB \(p. 229\)](#)

Ce didacticiel explique comment stocker les données des capteurs météorologiques dans une table de base de données. Il utilise l'instruction de requête de règles et les modèles de substitution pour formater les données des messages pour le service de destination, [Amazon DynamoDB](#).

Si vous utilisez DynamoDB pour la première fois, consultez ses exercices de [démarrage](#) avant de commencer ce didacticiel.

- [Tutoriel : Formatage d'une notification à l'aide d'une AWS Lambda fonction \(p. 235\)](#)

Ce didacticiel explique comment appeler une fonction Lambda pour reformater les données de l'appareil, puis les envoyer sous forme de message texte. Il ajoute un script Python et des fonctions AWS SDK dans une [AWS Lambda](#) fonction permettant de formater avec les données utiles du message provenant des capteurs météorologiques et d'envoyer un message texte.

Si vous utilisez Lambda pour la première fois, consultez ses exercices de [démarrage](#) avant de commencer ce didacticiel.

AWS IoT vue d'ensemble des règles

Tous ces didacticiels créent des AWS IoT règles.

Pour qu'une AWS IoT règle envoie les données d'un appareil à un autre AWS service, elle utilise :

- Une instruction de requête de règle composée des éléments suivants :
 - Clause SQL SELECT qui sélectionne et met en forme les données de la charge utile du message
 - Un filtre de rubrique (l'objet FROM dans l'instruction de requête de règle) qui identifie les messages à utiliser
 - Une instruction conditionnelle facultative (une clause SQL WHERE) qui spécifie les conditions spécifiques sur lesquelles agir
 - Au moins une action

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 provenant de la charge utile du message MQTT entrant à des fins d'utilisation par les actions de la règle. Pour obtenir la liste complète des actions de règle, consultez [Actions de règle AWS IoT \(p. 531\)](#).

Didacticiels dans cette section

- [Tutoriel : Republication d'un message MQTT \(p. 215\)](#)
- [Tutoriel : Envoi d'une notification Amazon SNS \(p. 221\)](#)
- [Tutoriel : Stockage des données de l'appareil dans une table DynamoDB \(p. 229\)](#)
- [Tutoriel : Formatage d'une notification à l'aide d'une AWS Lambda fonction \(p. 235\)](#)

Tutoriel : Republication d'un message MQTT

Ce didacticiel explique comment créer une AWS IoT règle qui publie un message MQTT lorsqu'un message MQTT spécifique est reçu. La charge utile des messages entrants peut être modifiée par la règle avant sa publication. Cela permet de créer des messages adaptés à des applications spécifiques sans avoir à modifier votre appareil ou son micrologiciel. 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 se comportent comme des messages envoyés par n'importe quel autre AWS IoT appareil ou client. Les appareils peuvent s'abonner aux messages republiés de la même manière qu'ils peuvent s'abonner à n'importe quel autre sujet de message MQTT.

Ce que vous allez apprendre dans ce didacticiel :

- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester une AWS IoT règle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel, vous effectuerez les tâches suivantes

- [Consultez les rubriques et les AWS IoT règles relatives au MQTT \(p. 215\)](#)
- [Étape 1 : Créer une AWS IoT règle pour republier un message MQTT \(p. 216\)](#)
- [Étape 2 : Tester votre nouvelle règle \(p. 218\)](#)
- [Étape 3 : Examiner les résultats et les étapes \(p. 221\)](#)

Avant de commencer ce didacticiel, vérifiez que vous respectez les conditions

- [Configurez votre Compte AWS \(p. 19\)](#)

Vous aurez besoin de votre AWS IoT console et de votre console pour suivre ce didacticiel.

- Révisé [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#)

Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner et publier sur un sujet. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans le cadre de cette procédure.

Consultez les rubriques et les AWS IoT règles relatives au MQTT

Avant de parler de AWS IoT règles, il est utile de 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 didacticiel. Pour plus d'informations

sur la comparaison entre MQTT et HTTP, consultez [Choix d'un protocole pour la communication de votre appareil \(p. 90\)](#).

Protocole MQTT

Le protocole MQTT utilise un modèle de communication publié/abonnement avec son hôte. Pour envoyer des données, les appareils publient des messages identifiés par sujet vers le courtier de AWS IoT messages. Pour recevoir des messages du courtier de messages, les appareils s'abonnent aux sujets qu'ils recevront en envoyant des filtres de sujet dans les demandes d'abonnement au courtier de messages. Le moteur de AWS IoT règles reçoit les messages MQTT du courtier de messages.

Règles AWS IoT

AWS IoTLes règles se composent d'une instruction de requête de règle et d'une ou de plusieurs actions de règle. Lorsque le moteur de AWS IoT règles reçoit un message MQTT, ces éléments agissent sur le message comme suit.

- Instruction de requête de règle

L'instruction de requête de la règle décrit les rubriques MQTT à utiliser, interprète les données issues de la charge utile du message et met en forme les données conformément à une instruction SQL similaire aux instructions utilisées par les bases de données SQL les plus courantes. Le résultat de l'instruction de requête est constitué des données envoyées aux actions de la règle.

- Action de la règle

Chaque action d'une règle agit sur les données résultant de l'instruction de requête de la règle. AWS IoT prend en charge [de nombreuses actions liées aux règles \(p. 531\)](#). Dans ce didacticiel, vous allez toutefois vous concentrer sur l'action de la [Republish \(p. 595\)](#) règle, qui publie le résultat de l'instruction de requête sous la forme d'un message MQTT avec un sujet spécifique.

Étape 1 : Créer uneAWS IoT règle pour republier un message MQTT

La AWS IoT règle que vous allez créer dans ce didacticiel s'inscrit dans les rubriques `device/device_id/data` MQTT où `device_id` est l'identifiant de l'appareil qui a envoyé le message. Ces rubriques sont décrites par un [filtre de rubrique \(p. 116\)](#) sous `device/+data` la+ forme d'un caractère générique qui correspond à n'importe quelle chaîne comprise entre les deux barres obliques.

Lorsque la règle reçoit un message provenant d'une rubrique correspondante, elle republie `latemperature valeur` sous `device_id` et sous la forme d'un nouveau message MQTT avec `ladevice/device/data/temp` rubrique.

Par exemple, la charge utile d'un message MQTT contenant `ladevice/22/data` sujet ressemble à ceci :

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

La règle prend `latemperature` valeur de la charge utile du message et `device_id` de la rubrique, et les republie sous la forme d'un message MQTT avec `ladevice/data/temp` sujet et une charge de message qui ressemble à ceci :

```
{
```

```
"device_id": "22",
"temperature": 28
}
```

Selon cette règle, les appareils qui n'ont besoin que de l'identifiant de l'appareil et des données de température s'abonnent à la device/data/temp rubrique pour ne recevoir que ces informations.

Pour créer une règle qui republie un message MQTT

1. Ouvrez [le hub Rules de la AWS IoT console](#).
2. Dans Règles, choisissez Créer et commencez à créer votre nouvelle règle.
3. Dans la partie supérieure de la section Créer une règle :

- a. Dans Nom, entrez le nom de la règle. Pour ce didacticiel, nommez-le **republish_temp**.

N'oubliez pas que le nom d'une règle doit être unique au sein de votre compte et de votre région, et qu'il ne peut pas contenir d'espaces. Nous avons utilisé un trait de soulignement dans ce nom pour séparer les deux mots du nom de la règle.

- b. Dans Description, décrivez la règle.

Une description significative vous permet de vous souvenir de la fonction de cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, donc soyez aussi détaillée que possible.

4. Dans l'instruction de requête Rule de Create a rule :

- a. Dans Utilisation de la version SQL, sélectionnez **2016-03-23**.

- b. Dans la zone d'édition de l'instruction de requête de règles, saisissez l'instruction :

```
SELECT topic(2) as device_id, temperature FROM 'device+/data'
```

Cette déclaration :

- Écoute les messages MQTT dont la rubrique correspond au filtre de device/+data rubrique.
- Sélectionne le deuxième élément de la chaîne de rubrique et l'affecte au device_id champ.
- Sélectionne le temperature champ de valeur dans la charge utile du message et l'affecte au temperature champ.

5. Dans Définir une ou plusieurs actions :

- a. Pour ouvrir la liste des actions de règle pour cette règle, choisissez Ajouter une action.
- b. Dans Sélectionner une action, choisissez Republier un message dans une AWS IoT rubrique.
- c. Au bas de la liste des actions, choisissez Configurer l'action pour ouvrir la page de configuration de l'action sélectionnée.

6. Dans l'action Configurer :

- a. Dans Rubrique, entrez **device/data/temp**. Il s'agit du sujet MQTT du message que cette règle va publier.
- b. Dans Qualité de service, choisissez 0 : le message est délivré zéro fois ou plus.
- c. Dans Choisissez ou créez un rôle pour autoriser l'AWS IoT accès à l'exécution de cette action :
 - i. Choisissez Create Role (Créer un rôle). La boîte de dialogue Créer un nouveau rôle s'ouvre.
 - ii. Entrez un nom qui décrit le nouveau rôle. Dans ce didacticiel, utilisez **republish_role**.

Lorsque vous créez un nouveau rôle, les politiques appropriées pour exécuter l'action de règle sont créées et associées au nouveau rôle. Si vous modifiez le sujet de cette action de règle ou si vous utilisez ce rôle dans une autre action de règle, vous devez mettre à jour la

- politique associée à ce rôle afin d'autoriser le nouveau sujet ou la nouvelle action. Pour mettre à jour un rôle existant, choisissez Mettre à jour le rôle dans cette section.
- iii. Choisissez Créer un rôle pour créer le rôle et fermer la boîte de dialogue.
 - d. Choisissez Ajouter une action pour ajouter l'action à la règle et revenir à la page Crée une règle.
 - 7. L'action Republier un message dans unAWS IoT sujet est désormais répertoriée dans Définir une ou plusieurs actions.

Dans la vignette de la nouvelle action, sous Republier un message dans uneAWS IoT rubrique, vous pouvez voir la rubrique dans laquelle votre action de republication sera publiée.

- Il s'agit de la seule action que vous allez ajouter à cette règle.
- 8. Dans Crée une règle, faites défiler l'écran vers le bas et choisissez Crée une règle pour créer la règle et terminer cette étape.

Étape 2 : Tester votre nouvelle règle

Pour tester votre nouvelle règle, vous allez utiliser le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouvrez le [client MQTT dans laAWS IoT console](#) dans une nouvelle fenêtre. Cela vous permettra de modifier la règle sans perdre la configuration de votre client MQTT. Le client MQTT ne conserve aucun abonnement ni journal de messages si vous le laissez pour accéder à une autre page de la console.

Pour utiliser le client MQTT pour tester votre règle

- 1. Dans le [client MQTT de laAWS IoT console](#), abonnez-vous aux rubriques d'entrée, dans ce cas, `device/+data`.
 - a. Dans le client MQTT, sous Abonnements, choisissez S'abonner à une rubrique.
 - b. Dans Rubrique d'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).

Dans la colonne Abonnements, sous Publier dans une rubrique, `device/+data` apparaît.

- 2. Abonnez-vous au sujet que votre règle publiera : `device/data/temp`.
 - a. Sous Abonnements, choisissez à nouveau S'abonner à une rubrique, puis dans Rubrique d'abonnement, entrez le sujet 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).

Dans la colonne Abonnements, sous `device/+data/device/data/temp` apparaît.

- 3. Publiez un message dans la rubrique d'entrée avec un identifiant d'appareil spécifique, `device/22/data`. Vous ne pouvez pas publier sur MQTT des rubriques contenant des caractères génériques.
 - a. Dans le client MQTT, sous Abonnements, choisissez Publier dans la rubrique.
 - b. Dans le champ Publier, entrez le nom de la rubrique saisie, `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, choisissez Publier dans le sujet.
4. Passez en revue les messages qui ont été envoyés.
 - a. Dans le client MQTT, sous Abonnements, un point vert apparaît à côté des deux rubriques auxquelles vous vous êtes abonné précédemment.

Les points verts indiquent qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.

- b. Sous Abonnements, choisissez device/+data pour 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. Sous Abonnements, choisissez device/data/temp pour vérifier que la charge utile de vos messages republiés ressemble à ceci :

```
{  
    "device_id": "22",  
    "temperature": 28  
}
```

Notez que la `device_id` valeur est une chaîne entre guillemets et que la `temperature` valeur est numérique. Cela est dû au fait que la [topic\(\)](#) fonction a extrait la chaîne du nom de rubrique du message d'entrée alors que la `temperature` valeur utilise la valeur numérique de la charge utile du message d'entrée.

Si vous souhaitez transformer la `device_id` valeur en valeur numérique, remplacez `topic(2)` dans l'instruction de requête de règle par :

```
cast(topic(2) AS DECIMAL)
```

Notez que la conversion de `topic(2)` valeur en valeur numérique ne fonctionnera que si cette partie de la rubrique ne contient que des caractères numériques.

5. Si vous constatez que le bon message a été publié dans la rubrique appareil/données/temp, cela signifie que votre règle a fonctionné. Pour en savoir plus sur l'action relative à la règle Republier, consultez la section suivante.

Si le message correct n'a pas été publié dans les rubriques Device/+data ou Device/Data/Temp, consultez les conseils de résolution des problèmes.

Résolution des problèmes liés à la règle de republication des messages

Voici quelques points à vérifier au cas où vous n'obtiendriez pas les résultats escomptés.

- Vous avez reçu une bannière d'erreur

Si une erreur est apparue lorsque vous avez publié le message d'entrée, corrigez-la d'abord. 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 la device/22/data rubrique, ce message doit apparaître dans le client MQTT si vous êtes abonné au filtre device/+data rubrique comme décrit dans la procédure.

Things to check

- Vérifiez le filtre de sujet 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 du sujet auquel vous êtes abonné et comparez-le au sujet dans lequel vous avez publié. Les noms des rubriques font la distinction majuscules/minuscules et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique dans laquelle vous avez publié le message.

- Vérifiez la fonction de publication des messages

Dans le client MQTT, sous Abonnements, choisissez device/+data, vérifiez le sujet du message de publication, puis choisissez Publier dans le sujet. Vous devriez voir apparaître la charge utile du message dans la zone d'édition située sous le sujet dans la liste des messages.

- Votre message republié ne s'affiche pas dans le client MQTT

Pour que votre règle fonctionne, elle doit disposer de la politique appropriée qui l'autorise à recevoir et à republier un message et elle doit recevoir le message.

Things to check

- Vérifiez le fonctionnementRégion AWS de votre client MQTT et la règle que vous avez créée

La console sur laquelle vous exécutez le client MQTT doit se trouver dans la mêmeAWS région que la règle que vous avez créée.

- Vérifiez le sujet du message d'entrée dans l'instruction de requête de règle

Pour que la règle fonctionne, elle doit recevoir un message dont le nom de rubrique correspond 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 font la distinction majuscules/minuscules et le sujet du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.

- Vérifiez le contenu de la charge utile du message

Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message déclarée dans l'instruction SELECT.

Vérifiez l'orthographe du temperature champ dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ distinguent les majuscules et minuscules et le temperature champ de l'instruction de requête de règle doit être identique à celui de la charge utile du message.temperature

Assurez-vous que le document JSON contenu dans la charge utile du message est correctement formaté. Si le fichier JSON contient des erreurs, par exemple une virgule manquante, la règle ne pourra pas le lire.

- Vérifiez le sujet du message republié dans l'action de la règle

La rubrique dans laquelle l'action Republier la règle publie le nouveau message doit correspondre à la rubrique à laquelle vous êtes abonné dans le client MQTT.

Ouvrez la règle que vous avez créée dans la console et vérifiez le sujet dans lequel l'action de la règle republiera le message.

- Vérifiez le rôle utilisé par la règle

L'action de règle doit être autorisée à recevoir le sujet d'origine et à publier le nouveau sujet.

Les règles qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez le sujet utilisé pour republier les données du message, vous devez mettre à jour le rôle de l'action de règle afin de mettre à jour sa politique en fonction du sujet actuel.

Si vous pensez 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.

Étape 3 : Examiner les résultats et les étapes

Dans le présent didacticiel

- Vous avez utilisé une simple requête SQL 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 votre AWS IoT règle.

Étapes suivantes

Après avoir republié quelques messages avec cette règle, essayez-la pour voir comment la modification de certains aspects du didacticiel affecte le message republié. Voici quelques idées pour vous aider à démarrer.

- Modifiez le **device_id** dans la rubrique du message d'entrée et observez l'effet sur la charge utile du message republié.
- Modifiez les champs sélectionnés dans l'instruction de requête de règle et observez l'effet sur la charge utile du message republié.
- Essayez le prochain didacticiel de cette série et découvrez comment [Tutorial : Envoi d'une notification Amazon SNS \(p. 221\)](#).

L'action Republier les règles utilisée dans ce didacticiel peut également vous aider à débugger les instructions de requête de règles. 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.

Tutoriel : Envoi d'une notification Amazon SNS

Ce didacticiel explique comment créer une AWS IoT règle qui envoie des données de message MQTT à une rubrique Amazon SNS afin qu'elles puissent être envoyées sous forme de SMS.

Dans ce didacticiel, vous allez créer une règle qui envoie des données de message provenant d'un capteur météo à tous les abonnés d'une rubrique Amazon SNS, chaque fois que la température dépasse la valeur définie dans la règle. La règle détecte lorsque la température signalée dépasse la valeur définie par

la règle, puis crée une nouvelle charge de message contenant uniquement l'identifiant de l'appareil, la température signalée et la limite de température dépassée. La règle envoie la charge utile du nouveau message sous la forme d'un document JSON à une rubrique SNS, qui informe tous les abonnés à la rubrique SNS.

Ce que vous allez apprendre dans ce didacticiel :

- Comment créer et tester une notification Amazon SNS
- Comment appeler une notification Amazon SNS à partir d'une AWS IoT règle
- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester une AWS IoT règle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel, vous effectuerez les tâches suivantes

- [Étape 1 : Créer une rubrique Amazon SNS qui envoie un SMS \(p. 222\)](#)
- [Étape 2 : Créer une AWS IoT règle pour envoyer le message \(p. 223\)](#)
- [Étape 3 : Test de la AWS IoT règle et de la notification Amazon SNS \(p. 225\)](#)
- [Étape 4 : Examiner les résultats et les étapes \(p. 228\)](#)

Avant de commencer ce didacticiel, vérifiez que vous respectez les conditions

- [Configurez votre Compte AWS \(p. 19\)](#)

Vous aurez besoin de votre AWS IoT console Compte AWS et de votre console pour suivre ce didacticiel.

- Révisé [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#)

Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner et publier sur un sujet. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans le cadre de cette procédure.

- A

Si vous n'avez jamais utilisé Amazon SNS auparavant, consultez la [section Configuration de l'accès à Amazon SNS](#). Si vous avez déjà suivi d'autres AWS IoT didacticiels, votre Compte AWS devriez déjà être correctement configuré.

Étape 1 : Créer une rubrique Amazon SNS qui envoie un SMS

Pour créer une rubrique Amazon SNS qui envoie un SMS

1. Créez une rubrique Amazon SNS.
 - a. Connectez-vous à la [console Amazon SNS](#).
 - b. Dans le panneau de navigation de gauche, choisissez Rubriques.
 - c. Sur la page Rubriques, choisissez Créer une rubrique.
 - d. Dans Détails, choisissez le type standard. Par défaut, la console crée une rubrique FIFO.
 - e. Dans Nom, 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 choisissez Créer une rubrique.

La console ouvre la page Détails de la nouvelle rubrique.

2. Créez un abonnement Amazon SNS.

Note

Le numéro de téléphone que vous utilisez dans le cadre de cet abonnement peut entraîner des frais de messagerie texte en raison des messages que vous enverrez dans ce didacticiel.

- a. Sur la page de détails de la rubrique `high_temp_notice`, sélectionnez **Créer un abonnement**.
 - b. Dans **Créer un abonnement**, dans la section **Détails**, dans la liste des protocoles, sélectionnez **SMS**.
 - c. Dans **Endpoint**, entrez le numéro d'un téléphone pouvant recevoir des SMS. Veillez à le saisir de telle sorte qu'il commence par un+, qu'il inclue le code du pays et de la région, et qu'il ne contienne aucun autre caractère de ponctuation.
 - d. Choisissez **Create subscription** (**Créer un abonnement**).
3. Testez la notification Amazon SNS.
 - a. Dans la [console Amazon SNS](#), dans le volet de navigation de gauche, sélectionnez **Rubriques**.
 - b. Pour ouvrir la page de détails du sujet, dans **Rubriques**, dans la liste des sujets, choisissez `high_temp_notice`.
 - c. Pour ouvrir la page **Publier le message** dans la rubrique, dans la page de détails de `high_temp_notice`, choisissez **Publier le message**.
 - d. Dans **Publier le message** dans le sujet, dans la section **Corps du message**, dans **Corps du message à envoyer au terminal**, entrez un court message.
 - e. Faites défiler la page vers le bas et choisissez **Publier 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 bien été reçu.

Si vous n'avez pas reçu le message de test, vérifiez le numéro de téléphone et les paramètres de votre téléphone.

Assurez-vous de pouvoir publier des messages de test depuis la [console Amazon SNS](#) avant de poursuivre le didacticiel.

Étape 2 : Créer uneAWS IoT règle pour envoyer le message

La AWS IoT règle que vous allez créer dans ce didacticiel s'inscrit dans les rubriques `device/device_id/data` MQTT où `device_id` trouve l'identifiant de l'appareil qui a envoyé le message. Ces rubriques sont décrites dans un filtre de rubrique sous la forme `device/+data`, où le+ est un caractère générique qui correspond à n'importe quelle chaîne comprise entre les deux barres obliques. Cette règle teste également la valeur du `temperature` champ dans la charge utile du message.

Lorsque la règle reçoit un message provenant d'une rubrique correspondante, elle prend le nom `device_id` de la rubrique, la `temperature` valeur de la charge utile du message, ajoute une valeur constante pour la limite qu'elle teste, et envoie ces valeurs sous forme de document JSON à une rubrique de notification Amazon SNS.

Par exemple, un message MQTT provenant du capteur météorologique numéro 32 utilise la `device/32/data` sujet et contient une charge utile qui ressemble à ceci :

```
{  
    "temperature": 38,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

}

L'instruction de requête de règle de la règle prend la `temperature` valeur de la charge utile du message, celle `device_id` du nom de la rubrique, et ajoute la `max_temperature` valeur constante pour envoyer une charge utile de message qui ressemble à ceci à la rubrique Amazon SNS :

```
{  
    "device_id": "32",  
    "reported_temperature": 38,  
    "max_temperature": 30  
}
```

Pour créer une AWS IoT règle permettant de détecter une valeur de température dépassée et de créer les données à envoyer à la rubrique Amazon SNS

1. Ouvrez [le hub Rules de la AWS IoT console](#).
2. S'il s'agit de votre première règle, choisissez Créer ou Créer une règle.
3. Dans Créer une règle :

- a. Pour Name (Nom), entrez `temp_limit_notify`.

N'oubliez pas que le nom d'une règle doit être unique au sein de votre régionCompte AWS et qu'il ne peut pas contenir d'espaces. Nous avons utilisé un trait de soulignement dans ce nom pour séparer les mots du nom de la règle.

- b. Dans Description, décrivez la règle.

Une description précise permet de se souvenir plus facilement du rôle de cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, donc soyez aussi détaillée que possible.

4. Dans l'instruction de requête Rule de Create a rule :

- a. Dans Utilisation de la version SQL, sélectionnez 2016-03-23.
- b. Dans la zone d'édition de l'instruction de requête de règles, saisissez 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 dont la rubrique correspond au filtre `device/+/data` rubrique et dont `temperature` la valeur est supérieure à 30.
 - Sélectionne le deuxième élément de la chaîne de rubrique et l'affecte au `device_id` champ.
 - Sélectionne le `temperature` champ de valeur dans la charge utile du message et l'affecte au `reported_temperature` champ.
 - Crée une valeur constante `30` pour représenter la valeur limite et l'affecte au `max_temperature` champ.
5. Pour ouvrir la liste des actions de règle pour cette règle, dans Définir une ou plusieurs actions, choisissez Ajouter une action.
 6. Dans Sélectionner une action, choisissez Envoyer un message sous forme de notification push SNS.
 7. Pour ouvrir la page de configuration de l'action sélectionnée, en bas de la liste des actions, choisissez Configurer l'action.

8. Dans l'action Configurer :
 - a. Dans SNS target, choisissez Select, recherchez votre rubrique SNS nommée high_temp_notice, puis sélectionnez Select.
 - b. Dans Format de message, choisissez RAW.
 - c. Dans Choisir ou créer un rôle pour autoriser AWS IoT l'accès à cette action, choisissez Créer un rôle.
 - d. Dans Créer un nouveau rôle, dans Nom, entrez un nom unique pour le nouveau rôle. Dans le cadre de ce tutoriel, utilisez **sns_rule_role**.
 - e. Sélectionnez Create role (Créer un rôle).

Si vous répétez ce didacticiel ou si vous réutilisez un rôle existant, choisissez Mettre à jour le rôle avant de continuer. Cela met à jour le document de politique du rôle pour qu'il fonctionne avec la cible SNS.

9. Choisissez Ajouter une action et revenez à la page Crée une règle.

Dans la vignette de la nouvelle action, sous Envoyer un message sous forme de notification push SNS, vous pouvez voir la rubrique SNS que votre règle appellera.

Il s'agit de la seule action que vous allez ajouter à cette règle.

10. Pour créer la règle et terminer cette étape, dans Crée une règle, faites défiler l'écran vers le bas et choisissez Crée une règle.

Étape 3 : Test de laAWS IoT règle et de la notification Amazon SNS

Pour tester votre nouvelle règle, vous allez utiliser le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouvrez le [client MQTT dans laAWS IoT console](#) dans 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 abonnement ni journal de messages.

Pour utiliser le client MQTT pour tester votre règle

1. Dans le [client MQTT de laAWS IoT console](#), abonnez-vous aux rubriques d'entrée, dans ce cas, `device/+data`.
 - a. Dans le client MQTT, sous Abonnements, choisissez S'abonner à une rubrique.
 - b. Dans Rubrique d'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).

Dans la colonne Abonnements, sous Publier dans une rubrique, `device/+data` apparaît.

2. Publiez un message dans la rubrique d'entrée avec un identifiant d'appareil spécifique, `device/32/data`. Vous ne pouvez pas publier sur MQTT des rubriques contenant des caractères génériques.
 - a. Dans le client MQTT, sous Abonnements, choisissez Publier dans la rubrique.
 - b. Dans le champ Publier, entrez le nom de la rubrique saisie, `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. Choisissez Publier dans le sujet pour publier votre message MQTT.
3. Confirmez que le message texte a été envoyé.
 - a. Dans le client MQTT, sous Abonnements, un point vert apparaît à côté du sujet auquel vous vous êtes abonné précédemment.

Le point vert indique qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.
 - b. Sous Abonnements, choisissez device/+data pour 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 assurez-vous que le contenu du message se présente comme suit :

```
{"device_id":"32","reported_temperature":38,"max_temperature":30}
```

Notez que la `device_id` valeur est une chaîne entre guillemets et que la `temperature` valeur est numérique. Cela est dû au fait que la [topic\(\)](#) fonction a extrait la chaîne du nom de rubrique du message d'entrée alors que la `temperature` valeur utilise la valeur numérique de la charge utile du message d'entrée.

Si vous souhaitez transformer la `device_id` valeur en valeur numérique, remplacez `topic(2)` dans l'instruction de requête de règle par :

```
cast(topic(2) AS DECIMAL)
```

Notez que la conversion de `topic(2)` valeur en une `DECIMAL` valeur numérique 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, sous Abonnements, choisissez Publier dans la rubrique.
 - b. Dans le champ Publier, entrez le nom de la rubrique saisi, **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, choisissez Publier dans le sujet.

Vous devriez voir le message que vous avez envoyé dans l'**device/+data** abonnement. Toutefois, étant donné que la valeur de température est inférieure à la température maximale indiquée dans l'instruction de requête de règle, vous ne devriez pas recevoir de message texte.

Si vous ne constatez pas le comportement correct, consultez les conseils de dépannage.

Résolution des problèmes liés à votre règle de message SNS

Voici quelques points à vérifier, au cas où vous n'obtiendriez pas les résultats escomptés.

- Vous avez reçu une bannière d'erreur

Si une erreur est apparue lorsque vous avez publié le message d'entrée, corrigez-la d'abord. 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 la **device/22/data** rubrique, ce message doit apparaître dans le client MQTT, si vous êtes abonné au filtre **device/+data** rubrique comme décrit dans la procédure.

Things to check

- Vérifiez le filtre de sujet 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 du sujet auquel vous êtes abonné et comparez-le au sujet dans lequel vous avez publié. Les noms des rubriques font la distinction majuscules/minuscules et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique dans laquelle vous avez publié le message.

- Vérifiez la fonction de publication des messages

Dans le client MQTT, sous Abonnements, choisissez **device/+data**, vérifiez le sujet du message de publication, puis choisissez Publier dans le sujet. Vous devriez voir apparaître la charge utile du message dans la zone d'édition située sous le sujet dans la liste des messages.

- Vous ne recevez pas de SMS

Pour que votre règle fonctionne, elle doit disposer de la politique appropriée qui l'autorise à recevoir un message et à envoyer une notification SNS, et elle doit recevoir le message.

Things to check

- Vérifiez le fonctionnement Région AWS de votre client MQTT et la règle que vous avez créée

La console sur laquelle vous exécutez le client MQTT doit se trouver dans la même AWS région que la règle que vous avez créée.

- Vérifiez 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 la règle, la règle n'exécutera aucune de ses actions.

- Vérifiez le sujet du message d'entrée dans l'instruction de requête de règle

Pour que la règle fonctionne, elle doit recevoir un message dont le nom de rubrique correspond 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 font la distinction majuscules/minuscules et le sujet du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.

- Vérifiez le contenu de la charge utile du message

Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message déclarée dans l'instruction SELECT.

Vérifiez l'orthographe du `temperature` champ dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ distinguent les majuscules et minuscules et le `temperature` champ de l'instruction de requête de règle doit être identique à celui de la charge utile du message `.temperature`.

Assurez-vous que le document JSON contenu dans la charge utile du message est correctement formaté. Si le fichier JSON contient des erreurs, par exemple une virgule manquante, la règle ne pourra pas le lire.

- Vérifiez le sujet du message republié dans l'action de la règle

La rubrique dans 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 le sujet dans lequel l'action de la règle republiera le message.

- Vérifiez le rôle utilisé par la règle

L'action de règle doit être autorisée à recevoir le sujet d'origine et à publier le nouveau sujet.

Les règles qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez le sujet utilisé pour republier les données du message, vous devez mettre à jour le rôle de l'action de règle afin de mettre à jour sa politique en fonction du sujet actuel.

Si vous pensez 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.

Étape 4 : Examiner les résultats et les étapes

Dans ce didacticiel :

- Vous avez créé et testé une rubrique et une souscription à Amazon SNS.
- Vous avez utilisé une simple requête SQL et des fonctions dans une instruction de requête de règle pour créer un nouveau message pour votre notification.
- Vous avez créé une AWS IoT règle pour envoyer une notification Amazon SNS utilisant votre charge de message personnalisée.
- Vous avez utilisé le client MQTT pour tester votre AWS IoT règle.

Étapes suivantes

Après avoir envoyé quelques SMS avec cette règle, essayez-la pour voir comment la modification de certains aspects du didacticiel affecte le message et le moment où il est envoyé. Voici quelques idées pour vous aider à démarrer.

- Modifiez le *device_id* dans le sujet 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 plutôt qu'une température maximale. N'oubliez pas de changer le nom *demax_temperature* !
- Ajoutez une action de règle de republication pour envoyer un message MQTT lorsqu'une notification SNS est envoyée.
- Essayez le prochain didacticiel de cette série et découvrez comment [Tutoriel : Stockage des données de l'appareil dans une table DynamoDB \(p. 229\)](#).

Tutoriel : Stockage des données de l'appareil dans une table DynamoDB

Ce didacticiel explique comment créer une AWS IoT règle qui envoie les données des messages à une table DynamoDB.

Dans ce didacticiel, vous allez créer une règle qui envoie les données d'un capteur météorologique imaginaire à une table DynamoDB. La règle met en forme les données de nombreux capteurs météorologiques de manière à ce qu'elles puissent être ajoutées à une seule table de base de données.

What you'll learn in this tutorial

- Comment créer une table DynamoDB
- Comment envoyer des données de message à une table DynamoDB à partir d'une AWS IoT règle
- Comment utiliser des modèles de substitution dans une AWS IoT règle
- Comment utiliser des requêtes et des fonctions SQL simples dans une instruction de requête de règle
- Comment utiliser le client MQTT pour tester une AWS IoT règle

Ce didacticiel vous prendra environ 30 minutes.

Dans ce didacticiel, vous effectuerez les tâches suivantes

- [Étape 1 : Créer la table DynamoDB \(p. 230\)](#)
- [Étape 2 : Crédit d'une AWS IoT règle pour envoyer des données à la table DynamoDB \(p. 230\)](#)
- [Étape 3 : Test de la AWS IoT règle et de la table DynamoDB \(p. 232\)](#)
- [Étape 4 : Examiner les résultats et les étapes \(p. 235\)](#)

Avant de commencer ce didacticiel, vérifiez que vous respectez les conditions

- [Configurez votre Compte AWS \(p. 19\)](#)

Vous aurez besoin de votre AWS IoT console Compte AWS et de votre console pour suivre ce didacticiel.

- Révisé [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#)

Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner et publier sur un sujet. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans le cadre de cette procédure.

- A consulté la présentation [d'Amazon DynamoDB](#)

Si vous n'avez jamais utilisé DynamoDB auparavant, consultez la [section Mise en route avec DynamoDB](#) pour vous familiariser avec les concepts et les opérations de base de DynamoDB.

Étape 1 : Créer la table DynamoDB

Dans ce didacticiel, vous allez créer une table DynamoDB avec les attributs suivants pour enregistrer les données provenant de capteurs météorologiques imaginaires :

- `sample_time` est une clé primaire qui décrit l'heure à laquelle l'échantillon a été enregistré.
- `device_id` est une clé de tri qui décrit le périphérique qui a fourni l'échantillon
- `device_data` sont les données reçues de l'appareil et formatées par l'instruction de requête de règle

Pour créer la table DynamoDB

1. Ouvrez la [console DynamoDB](#), puis choisissez Crée une table.
2. Dans Crée une table :
 - a. Dans Nom de la table, entrez le nom de la table :**wx_data**.
 - b. Dans la zone Clé de partition **sample_time**, entrez, puis choisissez dans la liste d'options à côté du champ**Number**.
 - c. Dans la zone Clé de tri **device_id**, entrez, puis choisissez dans la liste d'options à côté du champ**Number**.
 - d. Au bas de la page, cliquez sur Crée.

Vous le définirez `device_data` ultérieurement, lorsque vous configurerez l'action de la règle DynamoDB.

Étape 2 : Création d'une AWS IoT règle pour envoyer des données à la table DynamoDB

Au cours de cette étape, vous allez utiliser l'instruction de requête de règle pour formater les données provenant des capteurs météorologiques imaginaires afin de les écrire dans la table de base de données.

Voici un exemple de charge utile de message reçu d'un capteur météorologique :

```
{  
    "temperature": 28,  
    "humidity": 80,  
    "barometer": 1013,  
    "wind": {  
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

Pour l'entrée de base de données, vous allez utiliser l'instruction de requête de règle pour aplatis 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 deux[Modèles de substitution \(p. 681\)](#). Les modèles de substitution sont des expressions qui vous permettent d'insérer des valeurs dynamiques à partir de fonctions et de données de messages.

Pour créer la AWS IoT règle permettant d'envoyer des données à la table DynamoDB

1. Ouvrez [le hub Rules de la AWS IoT console](#).
2. Pour commencer à créer votre nouvelle règle dans Règles, choisissez Crée une règle.
3. Dans les propriétés de la règle :

- a. Dans Nom de la règle, entrez `wx_data_dbb`.

N'oubliez pas que le nom d'une règle doit être unique au sein de votre régionCompte AWS et qu'il ne peut pas contenir d'espaces. Nous avons utilisé un trait de soulignement dans ce nom pour séparer les deux mots du nom de la règle.

- b. Dans Description de la règle, décrivez la règle.

Une description précise permet de se souvenir plus facilement du rôle de cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, donc soyez aussi détaillée que possible.

4. Choisissez Next (Suivant) pour continuer.
5. Dans une instruction SQL :
 - a. Dans la version SQL, sélectionnez **2016-03-23**.
 - b. Dans la zone d'édition de l'instruction SQL, 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 dont la rubrique correspond au filtre `dedevice/+/data` rubrique.
- Formate les éléments de l'`wind` attribut en tant qu'attributs individuels.
- Transmet les `barometer` attribut `temperature` `humidity`, et sans modification.

6. Choisissez Next (Suivant) pour continuer.
7. Dans les actions relatives aux règles :
 - a. Pour ouvrir la liste des actions de règle pour cette règle, dans Action 1, sélectionnez **DynamoDB**.
 - b. Dans Nom de la table, choisissez le nom de la table DynamoDB que vous avez créée lors d'une étape précédente : `wx_data`.

Les champs Type de clé de partition et Type de clé de tri sont remplis avec les valeurs de votre table DynamoDB.

 - c. Dans Clé de partition, tapez `sample_time`.
 - d. Dans Valeur de la clé de partition, tapez `#{timestamp()}`.

Il s'agit de la première[Modèles de substitution \(p. 681\)](#) que vous utiliserez dans cette règle. Au lieu d'utiliser une valeur provenant de la charge utile du message, il utilisera la valeur renvoyée par la fonction `timestamp`. Pour en savoir plus, consultez la section [horodatage \(p. 673\)](#) dans le Guide du AWS IoT Core développeur.

- e. Dans la zone Clé de tri, entrez **device_id**.
- f. Dans Valeur de la clé de tri, entrez `${cast(topic(2) AS DECIMAL)}`.

C'est la deuxième [Modèles de substitution \(p. 681\)](#) que vous utiliserez dans cette règle. Il insère la valeur du second élément dans le nom de la rubrique, qui est l'identifiant de l'appareil, après l'avoir converti en une valeur DÉCIMALE correspondant au format numérique de la clé. Pour en savoir plus sur les rubriques, consultez la [rubrique \(p. 673\)](#) du Guide du AWS IoT Core développeur. Ou pour en savoir plus sur le casting, consultez le [casting \(p. 638\)](#) dans le Guide du AWS IoT Core développeur.

- g. Dans Write message data to this column (Écrire des données de message dans cette colonne), entrez **device_data**.
Cela créera la **device_data** colonne dans la table DynamoDB.
- h. Laissez le champ Operation (Opération) vide.
- i. Dans Rôle IAM, choisissez Créer un nouveau rôle.
- j. Dans la boîte de dialogue Créer un rôle, dans le champ Nom du rôle, entrez **wx_ddb_role**. Ce nouveau rôle contiendra automatiquement une politique avec le préfixe «aws-iot-rule» qui permettra à **lawx_data_ddb** règle d'envoyer des données à la table **wx_data** DynamoDB que vous avez créée.
- k. Dans le rôle IAM, choisissez **wx_ddb_role**.
- l. Au bas de la page, sélectionnez Next.

8. Au bas de la page Réviser et créer, choisissez Crée pour créer la règle.

Étape 3 : Test de la AWS IoT règle et de la 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.

Ouvrez le [client MQTT dans la AWS IoT console](#) dans une nouvelle fenêtre. Cela vous permettra de modifier la règle sans perdre la configuration de votre client MQTT. Le client MQTT ne conserve aucun abonnement ni journal de messages si vous le laissez pour accéder à une autre page de la console. Vous souhaiterez également qu'une fenêtre de console distincte soit ouverte sur le [hub DynamoDB Tables de la AWS IoT console](#) pour afficher les nouvelles entrées envoyées par votre règle.

Pour utiliser le client MQTT pour tester votre règle

1. Dans le [client MQTT de la AWS IoT console](#), abonnez-vous à la rubrique d'entrée **device/+data**.
 - a. Dans le client MQTT, choisissez S'abonner à une rubrique.
 - b. Dans le champ Filtre de rubrique, entrez le sujet du filtre de rubrique d'entrée **device/+data**.
 - c. Choisissez Subscribe.
2. Maintenant, publiez un message dans la rubrique d'entrée avec un identifiant d'appareil spécifique **device/22/data**. Vous ne pouvez pas publier sur MQTT des rubriques contenant des caractères génériques.
 - a. Dans le client MQTT, choisissez Publier dans une rubrique.
 - b. Dans le champ Nom du sujet, entrez le nom du sujet d'entrée **device/22/data**.
 - c. Pour Message payload, entrez les exemples de données suivants.

```
{  
  "temperature": 28,  
  "humidity": 80,  
  "barometer": 1013,  
  "wind": {
```

```
        "velocity": 22,  
        "bearing": 255  
    }  
}
```

- d. Pour publier le message MQTT, choisissez Publier.
- e. À présent, dans le client MQTT, choisissez S'abonner à une rubrique. Dans la colonne S'abonner, choisissez l'**device/+data** abonnement. Vérifiez que les exemples de données de l'étape précédente y figurent.
3. Vérifiez la ligne de la table DynamoDB que votre règle a créée.
 - a. Dans le [hub de tables DynamoDB de la AWS IoT console](#), choisissez wx_data, puis choisissez l'onglet Éléments.

Si vous êtes déjà dans l'onglet Éléments, vous devrez peut-être actualiser l'affichage en choisissant l'icône en choisissant l'icône en forme d'actualisation dans le coin supérieur droit de l'en-tête du tableau.

- b. Notez que les valeurs sample_time de la table sont des liens et ouvrez-en un. Si vous venez d'envoyer votre premier message, ce sera le seul de la liste.
- Ce lien affiche toutes les données de cette ligne du tableau.
- c. Développez l'entrée device_data pour afficher les données résultant de l'instruction de requête de règle.
- d. Explorez les différentes représentations des données disponibles dans cet écran. Vous pouvez également modifier les données dans cet écran.
- e. Une fois que vous avez terminé de consulter cette ligne de données, pour enregistrer les modifications que vous avez apportées, cliquez sur Enregistrer, ou pour quitter sans enregistrer de modifications, cliquez sur Annuler.

Si vous ne constatez pas le comportement correct, consultez les conseils de dépannage.

Résolution des problèmes de votre règle DynamoDB

Voici quelques points à vérifier au cas où vous n'obtiendriez pas les résultats escomptés.

- Vous avez reçu une bannière d'erreur

Si une erreur est apparue lorsque vous avez publié le message d'entrée, corrigez-la d'abord. 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 la `device/22/data` rubrique, ce message doit apparaître dans le client MQTT si vous êtes abonné au filtre `device/+data` rubrique comme décrit dans la procédure.

Things to check

- Vérifiez le filtre de sujet 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 du sujet auquel vous êtes abonné et comparez-le au sujet dans lequel vous avez publié. Les noms des rubriques font la distinction majuscules/minuscules et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique dans laquelle vous avez publié le message.

- Vérifiez la fonction de publication des messages

Dans le client MQTT, sous Abonnements, choisissez device/+data, vérifiez le sujet du message de publication, puis choisissez Publier dans le sujet. Vous devriez voir apparaître la charge utile du message dans la zone d'édition située sous le sujet dans la liste des messages.

- Vous ne voyez pas vos données dans la table DynamoDB

La première chose à faire est d'actualiser l'affichage en choisissant l'icône en choisissant l'icône en forme d'actualisation dans le coin supérieur droit de l'en-tête du tableau. Si les données que vous recherchez ne s'affichent pas, vérifiez les points suivants.

Things to check

- Vérifiez le fonctionnement Région AWS de votre client MQTT et la règle que vous avez créée

La console sur laquelle vous exécutez le client MQTT doit se trouver dans la même AWS région que la règle que vous avez créée.

- Vérifiez le sujet du message d'entrée dans l'instruction de requête de règle

Pour que la règle fonctionne, elle doit recevoir un message dont le nom de rubrique correspond 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 font la distinction majuscules/minuscules et le sujet du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.

- Vérifiez le contenu de la charge utile du message

Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message déclarée dans l'instruction SELECT.

Vérifiez l'orthographe du champ dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ distinguent les majuscules et minuscules et le champ de l'instruction de requête de règle doit être identique à celui de la charge utile du message.

Assurez-vous que le document JSON contenu dans la charge utile du message est correctement formaté. Si le fichier JSON contient des erreurs, par exemple une virgule manquante, la règle ne pourra pas le lire.

- Vérifiez les noms de clé et de champ utilisés dans l'action de la règle

Les noms de champ utilisés dans la règle de rubrique doivent correspondre à ceux figurant dans la charge utile du message JSON du message publié.

Ouvrez la règle que vous avez créée dans la console et vérifiez que les noms des champs dans la configuration de l'action de la règle correspondent à ceux utilisés dans le client MQTT.

- Vérifiez le rôle utilisé par la règle

L'action de règle doit être autorisée à recevoir le sujet d'origine et à publier le nouveau sujet.

Les règles 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 le nom de rubrique ou de table DynamoDB utilisé par la règle, vous devez mettre à jour le rôle de l'action de la règle pour mettre à jour sa politique en conséquence.

Si vous pensez 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.

Étape 4 : Examiner les résultats et les étapes

Après avoir envoyé quelques messages à la table DynamoDB avec cette règle, essayez de l'utiliser pour voir comment la modification de certains aspects du didacticiel affecte les données écrites dans la table. Voici quelques idées pour vous aider à démarrer.

- Modifiez le *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 l'utiliser pour filtrer les données stockées dans le tableau.
- Ajoutez une action de règle de republication pour envoyer un message MQTT pour chaque ligne ajoutée au tableau. Vous pouvez l'utiliser pour le débogage.

Lorsque vous aurez terminé ce didacticiel, vous effectuerez des tâches [the section called “Formatage d'une notification à l'aide d'une AWS Lambda fonction” \(p. 235\)](#).

Tutoriel : Formatage d'une notification à l'aide d'une AWS Lambda fonction

Ce didacticiel explique comment envoyer des données de message MQTT à une AWS Lambda action afin de les formater et de les envoyer à un autre AWS service. Dans ce didacticiel, l'AWS Lambda action utilise le AWS SDK pour envoyer le message formaté à la rubrique Amazon SNS que vous avez créée dans le didacticiel expliquant comment procéder [the section called “Envoi d'une notification Amazon SNS” \(p. 221\)](#).

Dans le didacticiel expliquant comment procéder [the section called “Envoi d'une notification Amazon SNS” \(p. 221\)](#), le document JSON résultant de l'instruction de requête de la règle a été envoyé en tant que corps du message texte. Le résultat était un message texte qui ressemblait à cet exemple :

```
{"device_id": "32", "reported_temperature": 38, "max_temperature": 30}
```

Dans ce didacticiel, vous allez utiliser une action de AWS Lambda règle pour appeler une AWS Lambda fonction qui met en forme les données de l'instruction de requête de règle dans un format plus convivial, comme dans cet exemple :

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

La AWS Lambda fonction que vous allez créer dans ce didacticiel met en forme la chaîne du message en utilisant les données de l'instruction de requête de règle et appelle la fonction de [publication SNS](#) du AWS SDK pour créer la notification.

What you'll learn in this tutorial

- Comment créer et tester une AWS Lambda fonction
- Comment utiliser le AWS SDK dans une AWS Lambda fonction de publication d'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 une AWS IoT règle

Ce didacticiel vous prendra environ 45 minutes.

Dans ce didacticiel, vous effectuerez les tâches suivantes

- [Étape 1 : Créer une AWS Lambda fonction qui envoie un message \(p. 236\)](#)
- [Étape 2 : Création d'une AWS IoT règle avec une action de AWS Lambda règle \(p. 238\)](#)
- [Étape 3 : Test de la AWS IoT AWS Lambda règle et de son action \(p. 240\)](#)
- [Étape 4 : Examiner les résultats et les étapes \(p. 243\)](#)

Avant de commencer ce didacticiel, vérifiez que vous respectez les conditions

- [Configurez votre Compte AWS \(p. 19\)](#)

Vous aurez besoin de votre AWS IoT console Compte AWS et de votre console pour suivre ce didacticiel.

- Révisé [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#)

Assurez-vous de pouvoir utiliser le client MQTT pour vous abonner et publier sur un sujet. Vous utiliserez le client MQTT pour tester votre nouvelle règle dans le cadre de cette procédure.

- Vous avez terminé les autres didacticiels sur les règles de cette section.

Ce didacticiel nécessite la rubrique de notification SNS que vous avez créée dans le didacticiel pour savoir comment procéder [the section called "Envoi d'une notification Amazon SNS" \(p. 221\)](#). Cela suppose également que vous avez suivi les autres didacticiels relatifs aux règles de cette section.

- A revu la [AWS Lambda](#) vue d'ensemble

Si vous ne l'avez jamais utilisé AWS Lambda auparavant, consultez la [AWS Lambda](#) section [Commencer à utiliser Lambda](#) pour en apprendre les termes et les concepts.

Étape 1 : Créer une AWS Lambda fonction qui envoie un message

La AWS Lambda fonction de 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 sous forme de message dans une notification.

Contrairement au didacticiel expliquant comment [the section called "Envoi d'une notification Amazon SNS" \(p. 221\)](#), qui utilisait une action de AWS IoT règle pour envoyer la notification, ce didacticiel envoie la notification à partir de la fonction Lambda à l'aide d'une fonction du AWS SDK. La rubrique de notification Amazon SNS utilisée dans ce didacticiel est toutefois la même que celle que vous avez utilisée dans le didacticiel expliquant comment procéder [the section called "Envoi d'une notification Amazon SNS" \(p. 221\)](#).

Pour créer une AWS Lambda fonction qui envoie un message texte

1. Créez une AWS Lambda fonction.

- a. Dans la [console AWS Lambda](#), choisissez Créer une fonction.
- b. Dans la fonction Créer, sélectionnez Utiliser un plan.

Recherchez et sélectionnez le **hello-world-python** plan, puis choisissez Configurer.

- c. Dans Informations de base :

- i. Dans Nom de la fonction, entrez le nom de cette fonction, **format-high-temp-notification**.
- ii. Dans Rôle d'exécution, choisissez Créer un nouveau rôle à partir de modèles de AWS politique.
- iii. Dans Nom du rôle, entrez le nom du nouveau rôle **format-high-temp-notification-role**.
- iv. Dans Modèles de politique (facultatif), recherchez et sélectionnez la politique de publication Amazon SNS.

- v. Sélectionnez Create function (Créer une fonction).
2. Modifiez le code du plan pour le formater et envoyer une notification Amazon SNS.
- Une fois que vous avez créé votre fonction, vous devriez voir la page de format-high-temp-notificationdétails. Si ce n'est pas le cas, ouvrez-le depuis la page [Lambda Functions](#).
 - Sur la page de format-high-temp-notificationdétails, choisissez l'onglet Configuration et faites défiler l'écran jusqu'au panneau Code de fonction.
 - Dans la fenêtre Code de fonction, dans le volet Environnement, choisissez le fichier Python, lambda_function.py.
 - Dans la fenêtre Code de fonction, supprimez tout le code du programme d'origine du plan 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
```

- Choisissez Deploy (Déployer).
3. Dans une nouvelle fenêtre, recherchez le nom de ressource Amazon (ARN) de votre rubrique Amazon SNS dans le didacticiel [the section called “Envoi d'une notification Amazon SNS” \(p. 221\)](#).
- Dans une nouvelle fenêtre, ouvrez la [page Rubriques de la console Amazon SNS](#).
 - Sur la page Rubriques, recherchez la rubrique de notification high_temp_notice dans la liste des rubriques Amazon SNS.
 - Recherchez l'ARN de la rubrique de notification high_temp_notice à utiliser à l'étape suivante.

4. Créez un scénario de test pour votre fonction Lambda.
 - a. Sur la page [Fonctions Lambda](#) de la console, sur la page de format-high-temp-notificationdétails, choisissez Sélectionner un événement de test dans le coin supérieur droit de la page (même s'il semble désactivé), puis choisissez Configurer les événements de test.
 - b. Dans Configurer un événement de test, choisissez Créer un nouvel événement de test.
 - c. Dans Nom de l'événement, entrez **SampleRuleOutput**.
 - d. Dans l'éditeur JSON situé sous Nom de l'événement, collez cet exemple de document JSON. Voici un exemple de ce que votre AWS IoT 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 contenant l'ARN de la rubrique de notification `high_temp_notice` et copiez la valeur de l'ARN.
- f. Remplacez `notify_topic_arn` valeur dans l'éditeur JSON par l'ARN de votre sujet de notification.

Gardez cette fenêtre ouverte afin de pouvoir réutiliser cette valeur ARN lors de la création de la AWS IoT règle.

- g. Sélectionnez Create (Créer).
5. Testez la fonction à l'aide d'échantillons de données.

- a. Sur la page de format-high-temp-notificationdétails, dans le coin supérieur droit de la page, confirmez que cela SampleRuleOutputapparaît à côté du bouton Test. Si ce n'est pas le cas, choisissez-le dans la liste des événements de test disponibles.
- b. Pour envoyer le message de sortie de l'exemple de règle à votre fonction, choisissez Test.

Si la fonction et la notification fonctionnaient toutes deux, vous recevrez un SMS sur le téléphone qui s'est abonné à la notification.

Si vous n'avez pas reçu de SMS sur le téléphone, vérifiez le résultat de l'opération. Dans le panneau Code de fonction, dans l'onglet Résultat de l'exécution, passez en revue la réponse pour détecter toute erreur survenue. Ne passez à l'étape suivante tant que votre fonction n'a pas envoyé la notification à votre téléphone.

Étape 2 : Création d'une AWS IoT règle avec une action de AWS Lambda règle

Au cours de cette étape, vous allez utiliser l'instruction Rule Query pour formater les données provenant du capteur météorologique imaginaire afin de les envoyer à une fonction Lambda, qui formatera et enverra un message texte.

Voici un exemple de charge utile de message reçu des appareils météorologiques :

```
{  
    "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 la AWS IoT règle permettant d'appeler une fonction Lambda

1. Ouvrez le [hub Rules de la AWS IoT console](#).
2. Pour commencer à créer votre nouvelle règle dans Rules, choisissez Create.
3. Dans la partie supérieure de la section Créez une règle :

- a. Dans Nom, entrez le nom de la règle, **wx_friendly_text**.

N'oubliez pas que le nom d'une règle doit être unique au sein de votre région Compte AWS et qu'il ne peut pas contenir d'espaces. Nous avons utilisé un trait de soulignement dans ce nom pour séparer les deux mots du nom de la règle.

- b. Dans Description, décrivez la règle.

Une description précise permet de se souvenir plus facilement du rôle de cette règle et de la raison pour laquelle vous l'avez créée. La description peut être aussi longue que nécessaire, donc soyez aussi détaillée que possible.

- c. Dans l'instruction de requête Rule de Create a rule :

- a. Dans Utilisation de la version SQL, sélectionnez **2016-03-23**.
 - b. Dans la zone d'édition de l'instruction de requête de règles, saisissez 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 dont la rubrique correspond au filtre de device/+data rubrique et dont temperature la valeur est supérieure à 30.
 - Sélectionne le deuxième élément de la chaîne de rubrique, le convertit en nombre décimal, puis l'affecte à device_id champ.
 - Sélectionne la valeur du temperature champ à partir de la charge utile du message et l'affecte à reported_temperature champ.
 - Crée une valeur constante 30,, pour représenter la valeur limite et l'affecte à max_temperature champ.
 - Crée une valeur constante pour le notify_topic_arn champ.
- c. Reportez-vous à la fenêtre contenant l'ARN de la rubrique de notification high_temp_notice et copiez la valeur de l'ARN.

- d. Remplacez la valeur ARN (*arn:aws:sns:us-east-1:57 Example833:High_Temp_Notify*) dans l'éditeur d'instructions de requête de règles par l'ARN de votre sujet de notification.
5. Dans Définir une ou plusieurs actions :
 - a. Pour ouvrir la liste des actions de règle pour cette règle, choisissez Ajouter une action.
 - b. Dans Sélectionner une action, choisissez Envoyer un message à une fonction Lambda.
 - c. Pour ouvrir la page de configuration de l'action sélectionnée, en bas de la liste des actions, choisissez Configurer l'action.
6. Dans l'action Configurer :
 - a. Dans Nom de la fonction, choisissez Sélectionner.
 - b. Choisissez format-high-temp-notification.
 - c. Au bas de Configurer l'action, choisissez Ajouter une action.
 - d. Pour créer la règle, au bas de Créer une règle, choisissez Créer une règle.

Étape 3 : Test de laAWS IoT AWS Lambda règle et de son action

Pour tester votre nouvelle règle, vous allez utiliser le client MQTT pour publier et vous abonner aux messages MQTT utilisés par cette règle.

Ouvrez le [client MQTT dans laAWS IoT console](#) dans une nouvelle fenêtre. Vous pouvez désormais 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. Dans le [client MQTT de laAWS IoT console](#), abonnez-vous aux rubriques d'entrée, dans ce cas, `device/+data`.
 - a. Dans le client MQTT, sous Abonnements, choisissez S'abonner à une rubrique.
 - b. Dans Rubrique d'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).
2. Dans la colonne Abonnements, sous Publier dans une rubrique, `device/+data` apparaît.
 - 2.1. Publiez un message dans la rubrique d'entrée avec un identifiant d'appareil spécifique, `device/32/data`. Vous ne pouvez pas publier sur MQTT des rubriques contenant des caractères génériques.
 - a. Dans le client MQTT, sous Abonnements, choisissez Publier dans la rubrique.
 - b. Dans le champ Publier, entrez le nom de la rubrique saisie, `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, choisissez Publier dans le sujet.

3. Confirmez que le message texte a été envoyé.

- Dans le client MQTT, sous Abonnements, un point vert apparaît à côté du sujet auquel vous vous êtes abonné précédemment.

Le point vert indique qu'un ou plusieurs nouveaux messages ont été reçus depuis la dernière fois que vous les avez consultés.

- Sous Abonnements, choisissez device/+data pour 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  
    }  
}
```

- Vérifiez le téléphone que vous avez utilisé pour vous abonner à la rubrique SNS et assurez-vous que le contenu du message se présente comme suit :

```
Device 32 reports a temperature of 38, which exceeds the limit of 30.
```

Si vous modifiez l'élément d'identification du sujet dans le sujet du message, n'oubliez pas que la conversion de latopic(2) valeur en valeur numérique ne fonctionnera que si cet élément du sujet du message ne contient que des caractères numériques.

4. Essayez d'envoyer un message MQTT dans lequel la température ne dépasse pas la limite.

- Dans le client MQTT, sous Abonnements, choisissez Publier dans la rubrique.
- Dans le champ Publier, entrez le nom de la rubrique saisie, **device/33/data**.
- 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  
    }  
}
```

- Pour envoyer votre message MQTT, choisissez Publier dans le sujet.

Le message que vous avez envoyé dans l'**device/+data** abonnement devrait s'afficher. Toutefois, comme la valeur de température est inférieure à la température maximale indiquée dans l'instruction de requête de règle, vous ne devriez pas recevoir de message texte.

Si vous ne constatez pas le comportement correct, consultez les conseils de dépannage.

Résolution des problèmes liés à votre AWS Lambda règle et à votre notification

Voici quelques points à vérifier, au cas où vous n'obtiendriez pas les résultats escomptés.

- Vous avez reçu une bannière d'erreur

Si une erreur est apparue lorsque vous avez publié le message d'entrée, corrigez-la d'abord. 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 la device/32/data rubrique, ce message doit apparaître dans le client MQTT, si vous êtes abonné au filtre device/+data rubrique comme décrit dans la procédure.

Things to check

- Vérifiez le filtre de sujet 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 du sujet auquel vous êtes abonné et comparez-le au sujet dans lequel vous avez publié. Les noms des rubriques font la distinction majuscules/minuscules et la rubrique à laquelle vous vous êtes abonné doit être identique à la rubrique dans laquelle vous avez publié le message.

- Vérifiez la fonction de publication des messages

Dans le client MQTT, sous Abonnements, choisissez device/+data, vérifiez le sujet du message de publication, puis choisissez Publier dans le sujet. Vous devriez voir apparaître la charge utile du message dans la zone d'édition située sous le sujet dans la liste des messages.

- Vous ne recevez pas de SMS

Pour que votre règle fonctionne, elle doit disposer de la politique appropriée qui l'autorise à recevoir un message et à envoyer une notification SNS, et elle doit recevoir le message.

Things to check

- Vérifiez le fonctionnementRégion AWS de votre client MQTT et la règle que vous avez créée

La console sur laquelle vous exécutez le client MQTT doit se trouver dans la mêmeAWS région que la règle que vous avez créée.

- Vérifiez 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 la règle, la règle n'exécutera aucune de ses actions.

- Vérifiez le sujet du message d'entrée dans l'instruction de requête de règle

Pour que la règle fonctionne, elle doit recevoir un message dont le nom de rubrique correspond 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 font la distinction majuscules/minuscules et le sujet du message doit correspondre au filtre de rubrique de l'instruction de requête de règle.

- Vérifiez le contenu de la charge utile du message

Pour que la règle fonctionne, elle doit trouver le champ de données dans la charge utile du message déclarée dans l'instruction SELECT.

Vérifiez l'orthographe du temperature champ dans l'instruction de requête de règle avec celle de la charge utile du message dans le client MQTT. Les noms de champ distinguent les majuscules et minuscules et le temperature champ de l'instruction de requête de règle doit être identique à celui de la charge utile du message.temperature

Assurez-vous que le document JSON contenu dans la charge utile du message est correctement formaté. Si le fichier JSON contient des erreurs, par exemple une virgule manquante, la règle ne pourra pas le lire.

- Vérifiez la notification Amazon SNS

Dans [Étape 1 : Créer une rubrique Amazon SNS qui envoie un SMS \(p. 222\)](#), reportez-vous à l'étape 3 qui décrit comment tester la notification Amazon SNS et tester la notification pour vous assurer qu'elle fonctionne.

- Vérifiez la fonction Lambda

Dans [Étape 1 : Créer une AWS Lambda fonction qui envoie un message \(p. 236\)](#), reportez-vous à l'étape 5 qui décrit comment tester la fonction Lambda à l'aide de données de test et tester la fonction Lambda.

- Vérifiez le rôle utilisé par la règle

L'action de règle doit être autorisée à recevoir le sujet d'origine et à publier le nouveau sujet.

Les règles qui autorisent la règle à recevoir des données de message et à les republier sont spécifiques aux rubriques utilisées. Si vous modifiez le sujet utilisé pour republier les données du message, vous devez mettre à jour le rôle de l'action de règle afin de mettre à jour sa politique en fonction du sujet actuel.

Si vous pensez 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.

Étape 4 : Examiner les résultats et les étapes

Dans ce didacticiel :

- Vous avez créé une AWS IoT règle pour appeler une fonction Lambda qui a envoyé une notification Amazon SNS utilisant votre charge de message personnalisée.
- Vous avez utilisé une simple requête SQL et des fonctions dans une instruction de requête de règle pour créer une nouvelle charge de message pour votre fonction Lambda.
- Vous avez utilisé le client MQTT pour tester votre AWS IoT règle.

Étapes suivantes

Après avoir envoyé quelques SMS avec cette règle, essayez-la pour voir comment la modification de certains aspects du didacticiel affecte le message et le moment où il est envoyé. Voici quelques idées pour vous aider à démarrer.

- Modifiez le *device_id* dans le sujet 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 sur le contenu du message texte.
- Modifiez le test dans l'instruction de requête de règle pour tester une température minimale plutôt qu'une température maximale. Mettez à jour la fonction Lambda pour formater un nouveau message et n'oubliez pas de modifier le nom *demax_temperature*.
- Pour en savoir plus sur la détection des erreurs susceptibles de se produire lors du développement et de l'utilisation de AWS IoT règles, consultez [Surveillance des AWS IoT \(p. 466\)](#).

Conservation de l'état de l'appareil lorsque l'appareil est hors connexion avec Device Shadows

Ces didacticiels vous montrent comment utiliser l'AWS IoTService Device Shadow pour stocker et mettre à jour les informations d'état d'un appareil. 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 Shadow montre le changement de couleur d'une ampoule. Ces didacticiels montrent 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 didacticiels dans l'ordre dans lequel ils sont affichés ici, en commençant par le AWS IoTLes ressources que vous devez créer et la configuration matérielle nécessaire, ce qui vous aide également à apprendre les concepts de manière incrémentielle. Ces didacticiels montrent comment configurer et connecter un appareil Raspberry Pi à utiliser avec AWS IoT. Si vous ne disposez pas du matériel requis, vous pouvez suivre ces didacticiels en les adaptant à un appareil de votre choix ou en[création d'un appareil virtuel avec Amazon EC2 \(p. 44\)](#).

Présentation du scénario du didacticiel

Le scénario de ces didacticiels est une application ou un service local qui modifie la couleur d'une ampoule et publie ses données sur des sujets d'ombre réservés. Ces didacticiels sont similaires à la fonctionnalité Device Shadow décrite dans le[didacticiel de démarrage interactif \(p. 21\)](#)et sont implémentés sur un appareil Raspberry Pi. Les didacticiels de cette section se concentrent sur une seule ombre classique tout en montrant comment vous pouvez prendre en charge les ombres nommées ou plusieurs appareils.

Les didacticiels suivants vous apprendront à utiliser l'AWS IoTService Device Shadow.

- [Didacticiel : Préparation de votre Raspberry Pi pour exécuter l'application shadow \(p. 246\)](#)

Ce didacticiel montre comment configurer un appareil Raspberry Pi pour se connecter à AWS IoT. Vous allez également créer un AWS IoTdocument de stratégie et ressource objet, téléchargez les certificats, puis attachez la stratégie à cette ressource objet. Ce didacticiel vous prendra environ 30 minutes.

- [Didacticiel : Installation du kit SDK de périphériques et exécution de l'exemple d'application pour Device Shadows \(p. 250\)](#)

Ce didacticiel montre comment installer les outils, logiciels et logiciels requisAWS IoTSDK de périphérique pour Python, puis exécutez l'exemple d'application Shadow. Ce didacticiel s'appuie sur les concepts présentés dans[Connect un Raspberry Pi ou un autre appareil \(p. 61\)](#)et cela prend 20 minutes.

- [Didacticiel : Interaction avec Device Shadow à l'aide de l'exemple d'application et du client de test MQTT \(p. 257\)](#)

Ce didacticiel montre comment vous utilisez leshadow . pyExemple d'application et AWS IoTconsole pour observer l'interaction entre AWS IoTOmbres de l'appareil et changements d'état de l'ampoule. Le didacticiel montre également comment envoyer des messages MQTT aux rubriques réservées de Device Shadow. Ce didacticiel peut prendre 45 minutes.

AWS IoTPrésentation Device Shadow

Un Device Shadow est une représentation virtuelle persistante d'un appareil géré par un[ressource objet \(p. 287\)](#)que vous créez dans le AWS IoTregistre. Le document Shadow est un fichier JSON ou un JavaScriptDoc de notation utilisé pour stocker et extraire les informations sur l'état actuel d'un appareil. Vous pouvez utiliser l'shadow pour obtenir et définir l'état d'un appareil sur les rubriques MQTT ou les API REST HTTP, que l'appareil soit connecté ou non à Internet.

Un document Shadow contient unstatequi 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 l'`desired`objet.
- **reported:** les appareils rapportent leur état actuel dans le `reported`objet.
- **delta :**AWS IoT rapporte 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 appareil, vous pouvez utiliser le[Rubriques MQTT réservées \(p. 128\)](#), le[API REST Device Shadow \(p. 716\)](#)qui soutiennent l'GET,UPDATE, etDELETEopérations avec HTTP et le[AWS IoT CLI](#).

Dans l'exemple précédent, supposons que vous souhaitez modifier l'`desired`color to yellow. Pour ce faire, envoyez une demande au[UpdateThingShadow \(p. 718\)](#)API ou publiez un message dans l'[Mise à jour \(p. 723\)](#)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 correctement mis à jour le Device Shadow,AWS IoT publie la nouvelle `desired`état vers l'`delta` rubrique `$aws/things/THING_NAME/shadow/delta`. Dans ce cas, le document Shadow ressemble à ceci :

```
{  
  "state": {  
    "desired": {  
      "color": yellow  
    },  
    "reported": {  
      "color": green  
    },  
    "delta": {  
      "color": yellow  
    }  
  }  
}
```

Le nouvel état est ensuite signalé auAWS IoTDevice ShadowUpdatesujet\$aws/things/THING_NAME/shadow/updateavec le message JSON suivant :

```
{
```

```
"state": {  
    "reported": {  
        "color": yellow  
    }  
}
```

Si vous souhaitez obtenir les informations sur l'état actuel, envoyez une demande au[GetThingShadow \(p. 717\)](#)API ou publiez un message MQTT sur le[Faites \(p. 722\)](#)la rubrique :\$aws/things/THING_NAME/shadow/get.

Pour plus d'informations sur l'utilisation du service Device Shadow, consultez[Service AWS IoT Device Shadow \(p. 690\)](#).

Pour plus d'informations sur l'utilisation de Device Shadows dans les appareils, dans les applications et les services, consultez[Utilisation des shadows sur les appareils \(p. 694\)](#)et[Utilisation des shadows dans les applications et les services \(p. 697\)](#).

Pour plus d'informations sur l'interaction avecAWS IoTshadows, consultez[Interaction avec les shadows \(p. 709\)](#).

Pour en savoir plus sur les rubriques réservées MQTT et les API REST HTTP, consultez[Rubriques MQTT de Device Shadow \(p. 721\)](#)et[API REST Device Shadow \(p. 716\)](#).

Didacticiel : Préparation de votre Raspberry Pi pour exécuter l'application shadow

Ce didacticiel explique comment configurer et configurer un appareil Raspberry Pi et créer leAWS IoTressources dont un appareil a besoin pour connecter et échanger des messages MQTT.

Note

Si vous prévoyez de[the section called “Création d'un appareil virtuel avec Amazon EC2” \(p. 44\)](#), vous pouvez ignorer cette page et continuer à[the section called “Configurer votre appareil” \(p. 43\)](#). Vous allez créer ces ressources lorsque vous créez votre objet virtuel. Si vous souhaitez utiliser un autre appareil au lieu 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 :

- Configurez un appareil Raspberry Pi et configurez-le pour l'utiliser avecAWS IoT.
- Création d'unAWS IoTdocument de stratégie, qui autorise votre appareil à interagir avecAWS IoTServices .
- Crée une ressource objet dansAWS IoTles certificats du périphérique X.509, puis joignez le document de stratégie.

Le problème est la représentation virtuelle de votre appareil dans leAWS IoTregistre. Le certificat authentifie votre appareil auprès deAWS IoTCore, et le document de stratégie autorise votre appareil à interagir avecAWS IoT.

Comment exécuter ce tutoriel

Pour exécuter leshadow . pyexemple d'application pour Device Shadows, vous aurez besoin d'un appareil Raspberry Pi qui se connecte àAWS IoT. Nous vous recommandons de suivre ce tutoriel dans l'ordre dans lequel il est présenté ici, en commençant par la configuration du Raspberry Pi et de ses accessoires, puis la création d'une stratégie et l'attachement de la stratégie à une ressource objet que vous créez. Vous pouvez ensuite suivre ce didacticiel en utilisant l'interface utilisateur graphique (GUI) prise en charge par le 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 dans [Configurez votre Compte AWS \(p. 19\)](#) avant de continuer. Vous aurez besoin de votre Compte AWS pour terminer ce didacticiel.
- Le Raspberry Pi et ses accessoires nécessaires. Vous aurez besoin de :
 - UN [Raspberry Pi 3 Modèle B](#) ou un modèle plus récent. Ce tutoriel peut fonctionner sur des versions antérieures du Raspberry Pi, mais nous ne l'avons pas testé.
 - [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 pourraient 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.

Étape 1 : Configuration et configuration du périphérique Raspberry Pi

Dans cette section, nous allons configurer un appareil Raspberry Pi à utiliser avec AWS IoT.

Important

L'adaptation de ces instructions à d'autres appareils et systèmes d'exploitation peut s'avérer difficile. Vous devrez 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 exemple [Création d'un appareil virtuel avec Amazon EC2 \(p. 44\)](#) ou [Utilisez votre PC ou Mac Windows ou Linux comme AWS IoT appareil \(p. 52\)](#).

Vous devrez configurer votre Raspberry Pi de manière à ce qu'il puisse démarrer le système d'exploitation (OS), se connecter à Internet et vous permettre d'interagir avec lui via une interface de ligne de commande. Vous pouvez également utiliser l'interface utilisateur graphique (GUI) prise en charge par le Raspberry Pi pour ouvrir le AWS IoT et exécuter le reste de ce didacticiel.

Pour configurer le Raspberry Pi

1. Insérez la carte SD dans l'emplacement microSD du Raspberry Pi. Certaines cartes SD sont préchargées avec un gestionnaire d'installation qui vous invite à 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. Connectez un téléviseur ou un moniteur HDMI au câble HDMI qui se connecte au port HDMI du Raspberry Pi.
3. Connectez 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 difficultés dans l'installation du système d'exploitation, vous pouvez essayer les étapes suivantes. Pour plus d'informations sur la configuration du Raspberry Pi, consultez [Configuration de votre Raspberry Pi](#).

Si vous rencontrez des difficultés pour configurer le 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 peut ne pas apparaître.
- Assurez-vous que le téléviseur ou le moniteur est allumé et que l'entrée correcte est sélectionnée.
- Assurez-vous que vous utilisez un logiciel compatible avec Raspberry Pi.

Une fois que vous avez installé et configuré le système d'exploitation Raspberry Pi, ouvrez le navigateur Web du Raspberry Pi et accédez au AWS IoT Core pour poursuivre les autres étapes de ce didacticiel.

Si vous pouvez ouvrir l'AWS IoT Core, vous êtes Raspberry Pi est prêt et vous pouvez continuer à [the section called "Provisionnement de votre appareil dans AWS IoT" \(p. 248\)](#).

Si vous rencontrez des problèmes ou si vous avez besoin d'aide supplémentaire, consultez [Obtenez de l'aide pour votre Raspberry Pi](#).

Didacticiel : Provisionnement de votre appareil dans AWS IoT

Cette section crée les AWS IoT Core ressources que votre didacticiel utilisera.

Étapes à suivre pour provisionner votre appareil dans AWS IoT

- [Étape 1 : Création d'un AWS IoT stratégie Device Shadow \(p. 248\)](#)
 - [Étape 2 : Créez une ressource objet et attachez la stratégie à l'objet \(p. 249\)](#)
 - [Étape 3 : Passez en revue les résultats et les prochaines étapes \(p. 250\)](#)

Étape 1 : Création d'un AWS IoT stratégie Device Shadow

Les certificats X.509 authentifient votre appareil avec AWS IoT Core. AWS IoT Core utilise des stratégies attachées au certificat qui permettent au périphérique d'exécuter des opérations AWS IoT, telles que l'abonnement ou la publication de 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 l'AWS IoT opérations nécessaires pour exécuter l'exemple de programme. Nous vous recommandons de créer une stratégie qui accorde uniquement les autorisations requises pour exécuter la tâche. Vous créez le AWS IoT d'abord, puis attachez-la au certificat de l'appareil que vous allez créer ultérieurement.

Pour créer une stratégie AWS IoT

1. Dans le menu de gauche, choisissezSecure, puis choisissezStratégies. Si votre compte dispose de politiques existantes, choisissezCréer, sinon, sur leVous n'avez pas encore de politiqueChoisissez, choisissezCréation d'une stratégie.
 2. Sur la page Create a policy (Créer une stratégie) :
 - a. Saisissez un nom pour la stratégie dans leNomchamp (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 autorisent l'appareil à publier et à 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 l'objet que vous allez créer (par exemple, `My_light_bulb`), `region` avec la région AWS IoT dans laquelle vous utilisez les services, et `account` avec votre compte AWS nommé. Pour plus d'informations sur les AWS IoT Politiques, consultez [Stratégies AWS IoT Core \(p. 357\)](#).

```
{  
  "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-*"
        }
    ]
}
```

Étape 2 : Créez une ressource objet et attachez la stratégie à l'objet

Appareils connectés à AWS IoT peuvent être représentés par des ressources d'objets dans le registre AWS IoT. Une ressource objet représente un appareil spécifique ou une entité logique, telle que l'ampoule dans ce didacticiel.

Pour savoir comment créer un objet dans AWS IoT, suivez les étapes décrites dans [Création d'un objet \(p. 42\)](#). Voici quelques éléments clés à noter lorsque vous suivez les étapes de ce didacticiel :

1. Choisissez de créer un objet unique, et dans le nom, entrez un nom pour l'objet identique à l'objet thingname (par exemple, My_light_bulb) que vous avez spécifié lorsque vous avez créé 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 quethingname, créez une nouvelle chose avec le nom commethingnameet supprimez l'ancienne chose.

Note

N'utilisez pas d'informations personnelles identifiables dans votre nom d'objet. Le nom de l'objet peut apparaître dans des communications et des rapports non chiffrés.

2. Nous vous recommandons de télécharger chacun des fichiers de certificats sur leCertificat créé !dans un endroit où vous pouvez facilement les trouver. Vous devez installer ces fichiers pour exécuter l'exemple d'application.

Nous vous recommandons de télécharger les fichiers dans uncertssous-répertoire dans votrehomesur 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 d'attacher la stratégie que vous avez créée plus tôt (par exemple,**My_Device_Shadow_policy**) à la chose.

Une fois que vous avez créé un objet, vous pouvez voir la ressource de votre objet affichée dans la liste des éléments de laAWS IoTconsole

Étape 3 : Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous avez appris à :

- Configurez et configurez le périphérique Raspberry Pi.
- Création d'unAWS IoTdocument de stratégie qui autorise votre appareil à interagir avecAWS IoTServices .
- Créez une ressource objet et un certificat de périphérique X.509 associé, puis attachez-y le document de stratégie.

Étapes suivantes

Vous pouvez désormais installer leAWS IoTKit SDK des appareils pour Python, exécutez leshadow.pyexemple d'application et utilisez Device Shadows pour contrôler l'état. Pour plus d'informations sur l'exécution de ce didacticiel, consultez[Didacticiel : Installation du kit SDK de périphériques et exécution de l'exemple d'application pour Device Shadows \(p. 250\)](#).

Didacticiel : Installation du kit SDK de périphériques et exécution de l'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.pyexemple d'application pour modifier le document Shadow et contrôler l'état de l'ombre.

Dans ce didacticiel, vous allez apprendre à :

- Utilisez le logiciel installé et AWS IoTKit SDK pour appareil 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 souhaitée dans le AWS IoTconsole
- Vérifiez la rubrique shadow.py exemple d'application et comment il utilise le protocole MQTT pour mettre à jour l'état de l'ombre.

Avant de lancer ce didacticiel :

Vous avez dû configurer votreCompte AWS, a configuré votre appareil Raspberry Pi et créé unAWS IoTobjet et stratégie qui donnent à l'appareil les autorisations de publier et de s'abonner aux rubriques réservées MQTT du service Device Shadow. Pour plus d'informations, consultez [Didacticiel : Préparation de votre Raspberry Pi pour exécuter l'application shadow \(p. 246\)](#).

Vous devez également avoir installé Git, Python et le AWS IoTKit SDK des appareils pour Python. Ce didacticiel s'appuie sur les concepts présentés dans le didacticiel [Connect un Raspberry Pi ou un autre appareil \(p. 61\)](#). Si vous n'avez pas essayé ce didacticiel, nous vous recommandons de suivre les étapes décrites dans ce didacticiel pour installer les fichiers de certificats et le kit SDK de périphérique, puis de revenir à ce didacticiel pour exécuter le shadow.pyExemple d'application.

Dans ce didacticiel, vous allez :

- [Étape 1 : Exécutez l'exemple d'application shadow.py \(p. 251\)](#)
- [Étape 2 : Consultez l'exemple d'application shadow.py Device SDK \(p. 254\)](#)
- [Étape 3 : Résolution des problèmes liés à lashadow.pyExemple d'application \(p. 255\)](#)
- [Étape 4 : Passez en revue les résultats et les prochaines étapes \(p. 257\)](#)

Ce didacticiel vous prendra environ 20 minutes.

Étape 1 : Exécutez l'exemple d'application shadow.py

Avant de lancer le shadow.pyExemple d'application, vous aurez besoin des informations suivantes en plus des noms et de l'emplacement des fichiers de certificats que vous avez installés.

Valeurs des paramètres d'application

Paramètre	Où trouver la valeur
<i>your-iot-thing-Name</i>	Nom du AWS IoTchose que vous avez créée précédemment dans the section called “Étape 2 : Créez une ressource objet et attachez la stratégie à l'objet” (p. 249) .
<i>your-iot-endpoint</i>	Pour trouver cette valeur, dans le AWS IoTconsole , choisissez Gérer, puis choisissez Objets. La valeur a un format de : <i>endpoint_id</i> -ats.iot. <i>region</i> .amazonaws.com, par exemple, a3qj468EXAMPLE-ats.iot.us-west-2.amazonaws.com. Pour trouver cette valeur, procédez comme suit 1. Dans AWS IoTconsole , choisissez Gérer, puis choisissez Objets.

Paramètre	Où trouver la valeur
	2. Choisissez l'objet IoT que vous avez créé pour votre appareil, My_Light_Bulb, que vous avez utilisé précédemment, puis choisissez Interagir. Sur la page de détails de l'objet, votre point de terminaison s'affiche dans le HTTPSSection.

Installez et exécutez l'exemple d'application

- Accédez au répertoire d'exemples d'applications.

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

- Dans la fenêtre de ligne de commande, remplacez *your-iot-endpoint* et *your-iot-thing-Name* comme indiqué et exécutez cette commande.

```
python3 shadow.py --ca_file ~/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
```

- Notez que l'exemple d'application :

- Se connecte à AWS Service IoT pour votre compte.
- S'abonne à Delta événements et Update et Get Réponses.
- Vous invite à entrer la valeur souhaitée dans le terminal.
- Affiche une sortie 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 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, révisionnez la section ["Étape 3 : Résolution des problèmes liés à l'application"](#) ([p. 255](#)). Pour obtenir des informations supplémentaires susceptibles de vous aider à corriger le problème, ajoutez le `--verbose` debug de la ligne de commande afin que l'exemple d'application affiche des messages détaillés sur ce qu'il fait.

Entrez des valeurs et observez les mises à jour dans le document Shadow

Vous pouvez entrer des valeurs dans le terminal pour spécifier le paramètre `desired`, qui met également à jour la valeur `reportedValue`. Supposons que vous saisissez la couleur `yellow` dans le terminal. Le `reported` est également mise à jour vers la couleur `yellow`. Les messages affichés dans le terminal sont les suivants :

```
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 IoT crée une ombre classique par défaut pour la ressource objet. Vous pouvez observer la demande de mise à jour que vous avez publiée sur le `reported` et `desired` valeurs dans l'AWS IoT en regardant le document Shadow correspondant à la ressource objet que vous avez créée (par exemple, `My_light_bulb`). Pour voir la mise à jour dans le document Shadow :

1. Dans AWS IoT, choisissez `Gérer` puis `choisissez Objets`.
2. Dans la liste des objets affichés, sélectionnez l'objet que vous avez créé, choisissez `Shadows`, puis choisissez `Shadow Classique`.

Le document Shadow doit se présenter comme suit, montrant les `reported` et `desired` valeurs définies sur la couleur `yellow`. Vous voyez ces valeurs dans le `État` de l'ombre `selection` du document.

```
{  
  "desired": {  
    "welcome": "aws-iot",  
    "color": "yellow"  
  },  
  "reported": {  
    "welcome": "aws-iot",  
    "color": "yellow"  
  }  
}
```

Vous voyez également un `Métadonnées` qui 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 du document Shadow d'appareil. Si vous envoyez une autre demande de mise à jour, le numéro de version augmente 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": 1  
}
```

```
"version": 10
}
```

Pour en savoir plus sur le document Shadow et observer les modifications apportées aux informations d'état, passez au didacticiel suivant[Didacticiel : Interaction avec Device Shadow à l'aide de l'exemple d'application et du client de test MQTT \(p. 257\)](#)comme cela est décrit dans la[Étape 4 : Passez en revue les résultats et les prochaines étapes \(p. 257\)](#)section de ce didacticiel. Le cas échéant, vous pouvez également découvrir l'shadow.pyexemple de code et comment il utilise le protocole MQTT dans la section suivante.

Étape 2 : Consultez l'exemple d'application shadow.py Device SDK

Cette section passe en revue leshadow.pyExemple d'application de lAWS IoTKit SDK des appareils v2 pour Pythonutilisé dans ce didacticiel. Ici, nous allons examiner comment il se connecte àAWS IoT Coreen utilisant le protocole MQTT et MQTT sur WSS. Le[AWSRuntime commun \(AWS-CRT\)](#)fournit la prise en charge du protocole de communication de bas niveau et est incluse dans leAWS IoTKit SDK des appareils v2 pour Python.

Bien que ce tutoriel utilise MQTT et MQTT sur WSS,AWS IoTprend en charge les appareils qui publient des requêtes HTTPS. Pour obtenir un exemple de programme Python qui envoie un message HTTP depuis un appareil, consultez le[Exemple de code HTTPS \(p. 112\)](#)Utilisation de Pythonrequestsbibliothèque.

Pour plus d'informations sur la façon dont vous pouvez prendre une décision éclairée quant au protocole à utiliser pour les communications de votre appareil, consultez le[Choix d'un protocole pour la communication de votre appareil \(p. 90\)](#).

MQTT

Leshadow.pyExemples d'appelmtls_from_path(illustré ici) dans le[mqtt_connection_builder](#)pour établir une connexion avecAWS IoT Coreen utilisant le protocole MQTT.mtls_from_pathutilise les certificats X.509 et TLS v1.2 pour authentifier l'appareil. LeAWS-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.ca_file,
    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 IoTpoint de terminaison que vous avez transmis depuis la ligne de commande et`client_id`est l'ID qui identifie de manière unique cet appareil dans le champRégion AWS.
- `cert_filepath`,`pri_key_filepath`, et`ca_filepath`sont les chemins d'accès au certificat et aux fichiers de clé privée de l'appareil, ainsi que le fichier d'autorité de certification racine.
- `client_bootstrap`est l'objet d'exécution courant qui gère les activités de communication de socket et est instancié avant l'appel à`mqtt_connection_builder.mtls_from_path`.
- `on_connection_interrupted`et`on_connection_resumed`sont des fonctions de rappel à appeler lorsque la connexion de l'appareil est interrompue et reprend.
- `clean_session`indique s'il faut démarrer une nouvelle session persistante ou, s'il y en a une, se reconnecter à une session existante.`keep_alive_secs`est la valeur Keep Alive, en quelques secondes, à envoyer leCONNECTde 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.

Leshadow.py l'échantillon appelle également `websockets_with_default_aws_signing` dans `mqtt_connection_builder` pour établir une connexion avec AWS IoT Core en utilisant le protocole MQTT sur WSS. MQTT over WSS utilise également les mêmes paramètres que MQTT et prend les paramètres supplémentaires suivants :

- `region` est la région de signature utilisée par l'authentification Signature V4, `credentials_provider` est les informations d'identification fournies à utiliser pour l'authentification. La région est transmise à partir de la ligne de commande, et `credentials_provider` est instancié juste avant l'appel à `mqtt_connection_builder.websockets_with_default_aws_signing`.
- `websocket_proxy_options` sont les options proxy HTTP, si vous utilisez un hôte proxy. Dans `shadow.py` exemple d'application, cette valeur est instanciée juste avant l'appel à `mqtt_connection_builder.websockets_with_default_aws_signing`.

Abonnez-vous aux sujets et événements Shadow

`leshadow.py` exemple tente d'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, 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 appareil s'abonne à un message avec QoS niveau 1, le courtier de messages enregistre les messages auxquels l'appareil 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 [Vérifier le protocole MSON \(p. 197\)](#) et [MQTT \(p. 92\)](#).

Pour plus d'informations sur la façon dont MQTT, MQTT over WSS, les sessions persistantes et les niveaux de QoS utilisés dans ce didacticiel, voir [Consultez l'exemple d'application pubsub.py Device SDK \(p. 198\)](#).

Étape 3 : Résolution des problèmes liés à `leshadow.py` Exemple d'application

Lorsque vous exécutez `leshadow.py` exemple d'application, vous devriez voir certains messages s'afficher dans le terminal et une invite à entrer `undesiredValue`. 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 ce cas, voici quelques éléments que vous pouvez vérifier :

- Vérifiez l'adresse du point de terminaison dans la commande

Vérifiez la rubrique `endpoint` dans la commande que vous avez entrée pour exécuter l'exemple d'application (par exemple, `a3qEXAMPLEffp-ats.iot.us-west-2.amazonaws.com`) et vérifiez cette valeur dans le champ AWS IoT console.

Pour vérifier si vous avez utilisé la bonne valeur, procédez comme suit :

1. Dans AWS IoT console, choisissez Gérer puis choisissez Objets.
2. Choisissez l'objet que vous avez créé pour votre exemple d'application (par exemple, `My_Light_Bulb`) puis choisissez Interagir.

Sur la page de détails de l'objet, votre point de terminaison s'affiche dans la section HTTP. Vous devez également voir un message indiquant :`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, procédez comme suit :

1. Dans AWS IoT console, choisissez Gérer puis choisissez Objets.
2. Choisissez l'objet que vous avez créé pour votre exemple d'application (par exemple, My_Light_Bulb) puis choisissez Sécurité.
3. Sélectionnez le certificat, puis, dans la page des détails du certificat, choisissez Sélectionner le certificat, puis, dans la page de détails du certificat, choisissez Actions.

Si vous êtes dans la liste déroulante Activate, il n'est pas disponible et vous pouvez uniquement choisir Deactivate, votre certificat est actif. Sinon, choisissez Activate et réexécutez l'exemple de programme.

Si le programme ne s'exécute toujours pas, vérifiez les noms des fichiers de certificats dans le champ certs folder.

- Vérifiez la stratégie attachée à la ressource objet

Pendant que les certificats authentifient votre appareil, AWS IoT les stratégies permettent à l'appareil d'exécuter AWS IoT opérations, telles que l'abonnement ou la publication à des rubriques réservées MQTT.

Pour vérifier si la stratégie appropriée est attachée :

1. Recherchez le certificat comme décrit précédemment, puis choisissez Stratégies.
2. Choisissez la stratégie affichée et vérifiez si elle décrit la connect, subscribe, receive, et publish actions qui donnent à l'appareil l'autorisation de publier et de s'abonner aux rubriques réservées MQTT.

Pour obtenir un exemple de stratégie, consultez [Étape 1 : Création d'un AWS IoT stratégie Device Shadow \(p. 248\)](#).

Si des messages d'erreur indiquent un problème de connexion à AWS 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 offre un accès complet à AWS IoT Resources, puis réexécutez l'exemple de programme. Vous pouvez soit modifier la stratégie actuelle, soit choisir la stratégie actuelle, choisissez Detach, puis créez une autre stratégie qui fournit un accès complet et l'attache à votre ressource objet. 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 du SDK de votre appareil

Si le programme ne s'exécute toujours pas, vous pouvez réinstaller le kit SDK pour vous assurer que l'installation de votre SDK est terminée et correcte.

Étape 4 : Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous avez appris à :

- Installer le logiciel requis, les outils et l'AWS IoTKit SDK des appareils 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 le AWS 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 `shadow.py` exemple d'application et utilisez Device Shadows pour contrôler l'état. Vous pouvez observer les mises à jour du document Shadow dans le AWS 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 [Didacticiel : Interaction avec Device Shadow à l'aide de l'exemple d'application et du client de test MQTT \(p. 257\)](#).

Didacticiel : Interaction avec Device Shadow à l'aide de l'exemple d'application et du client de test MQTT

Pour interagir avec `shadow.py` exemple d'application, entrez une valeur dans le terminal pour l'application `desiredValeur`. Par exemple, vous pouvez spécifier des couleurs qui ressemblent aux feux de signalisation et AWS IoT répond à la demande et met à jour les valeurs signalées.

Dans ce didacticiel, vous allez apprendre à :

- Utilisation de `shadow.py` exemple d'application 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 comment le `shadow.py` exemple d'application y répond.
- Utilisez le client de test MQTT pour vous abonner à des rubriques instantanées et observer les mises à jour lorsque vous exécutez l'exemple de programme.

Avant de lancer ce didacticiel, vous devez disposer des éléments suivants :

Configurer votre compte AWS, a configuré votre appareil Raspberry Pi et créé un AWS IoT chose et politique. Vous devez également avoir installé le logiciel requis, le kit SDK de périphérique, les fichiers de certificats et exécuter l'exemple de programme dans le terminal. Pour plus d'informations, consultez les didacticiels précédents [Didacticiel : Préparation de votre Raspberry Pi pour exécuter l'application shadow \(p. 246\)](#) et [Étape 1 : Exécutez l'exemple d'application shadow.py \(p. 251\)](#). Si vous ne l'avez pas déjà fait, vous devez suivre ces didacticiels.

Dans ce didacticiel, vous allez :

- [Étape 1 : Mettre à jour les valeurs souhaitées et signalées à `shadow.py` Exemple d'application \(p. 258\)](#)
- [Étape 2 : Afficher les messages provenant de `shadow.py` Exemple d'application dans le client de test MQTT \(p. 259\)](#)
- [Étape 3 : Dépannage des erreurs liées aux interactions Device Shadow \(p. 263\)](#)
- [Étape 4 : Passez en revue les résultats et les prochaines étapes \(p. 264\)](#)

Ce didacticiel vous prendra environ 45 minutes.

Étape 1 : Mettre à jour les valeurs souhaitées et signalées à shadow.pyExemple d'application

Dans le didacticiel précédent [Étape 1 : Exécutez l'exemple d'application shadow.py \(p. 251\)](#), vous avez appris comment observer un message publié dans le document Shadow dans le AWS IoT lorsque vous entrez la valeur souhaitée, comme décrit dans la section [Didacticiel : Installation du kit SDK de périphériques et exécution de l'exemple d'application pour Device Shadows \(p. 250\)](#).

Dans l'exemple précédent, nous définissons la couleur souhaitée sur yellow. Une fois que vous avez entré chaque valeur, le terminal vous invite à saisir une autre valeur.desiredValeur . Si vous entrez à nouveau la même valeur (yellow), l'application reconnaît cela et vous invite à entrer un nouveau desiredValeur .

```
Enter desired value:  
yellow  
Local value is already 'yellow'.  
Enter desired value:
```

Maintenant, disons que vous entrez la couleur green. AWS IoT répond à la demande et met à jour le reportedValue green. C'est ainsi que la mise à jour se produit lorsque le desired est différent du reported état, provoquant un delta.

Procédure shadow.py exemple d'application simule les interactions Device Shadow :

1. Saisissez une desiredValeur (disons) yellow dans le terminal pour publier l'état souhaité.
2. Comme le desired est différent du reported état (dites la couleur) green, un delta se produit et l'application abonnée au delta reçoit ce message.
3. L'application répond au message et met à jour son état sur le desiredValeur, yellow.
4. L'application publie ensuite un message de mise à jour avec la nouvelle valeur signalée de l'état de l'appareil, yellow.

Vous trouverez ci-dessous les messages affichés dans le terminal qui indiquent 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'.
```

Dans AWS IoT, le document Shadow reflète la valeur mise à jour vers green pour les deux reported et desired et le numéro de version est incrémenté de 1. Par exemple, si le numéro de version précédente était affiché sur 10, le numéro de version actuel s'affiche sous la forme 11.

Note

La suppression d'une ombre ne réinitialise pas le numéro de version à 0. Vous verrez que la version de l'ombre est incrémentée de 1 lorsque vous publiez une demande de mise à jour ou que vous créez une autre ombre portant le même nom.

Modifier le document Shadow pour observer les événements Delta

L'application shadow.py Un exemple d'application est également abonné à delta et répond en cas de modification de la desiredValeur . Par exemple, vous pouvez modifier le desiredValeur de la couleur red. Pour ce faire, dans l'AWS IoT, modifiez le document Shadow en cliquant sur Modifier puis définissez le

paramètre `desired` valeur `red` dans le JSON, tout en conservant la valeur `green` dans le `reported`. 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"  
  }  
}
```

Une fois que vous avez enregistré la nouvelle valeur, le paramètre `shadow.py` répond à cette modification et affiche des messages dans le terminal indiquant le delta. Vous devriez alors voir les messages suivants apparaître sous l'invite de saisie `desiredValue`.

```
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'.
```

Étape 2 : Afficher les messages provenant du `shadow.py` Exemple d'application dans le client de test MQTT

Vous pouvez utiliser le plugin `Client de test MQTT` dans le `AWS IoT Console` pour surveiller les messages MQTT 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 consulter [Afficher les messages MQTT de l'appareil avec le client MQTT AWS IoT \(p. 71\)](#). Cela vous apprendra à utiliser l'Client de test MQTT dans le `AWS IoT Console` pour afficher les messages MQTT lorsqu'ils passent par le courtier de messages.

1. Ouvrez le client de test MQTT

Ouverture du [Test client MQTT dans le AWS IoT Console](#) dans 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 aucun abonnement ou journal 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 votre `AWS IoT` et le client de test MQTT s'ouvrent dans des fenêtres distinctes pour observer plus facilement l'interaction avec Device Shadows.

2. Abonnez-vous aux rubriques Shadow réservées 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 du `shadow.py` Exemple d'application. Pour vous abonner aux sujets suivants :

- a. Dans Client de test MQTT dans le `AWS IoT Console`, choisissez S'abonner à une rubrique.
- b. Dans Filtre de rubriques, saisissez `:$aws/things/thingName/shadow/update/ #`. Ici, `thingName` est le nom de la ressource objet que vous avez créée plus tôt (par exemple, `My_light_bulb`).

- c. Conservez les valeurs par défaut des paramètres de configuration supplémentaires, puis choisissez S'abonner.

En utilisant le `#` dans l'abonnement à la rubrique, vous pouvez vous abonner à plusieurs rubriques MQTT en même temps et observer tous les messages échangés entre l'appareil et son ombre dans une seule fenêtre. Pour plus d'informations sur les caractères génériques et leur utilisation, consultez [Rubriques MQTT \(p. 115\)](#).

3. Run (Exécuter Lambda) `shadow.py` exemple de programme et d'observation des 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 le Client de test MQTT dans la AWS IoT console.

- a. Exécutez la commande suivante pour redémarrer l'exemple de programme. Remplacez `your-iot-thing-Name` et `your-iot-endpoint` avec les noms du AWS IoT objet que vous avez créé précédemment (par exemple, `My_light_bulb`), et le point de terminaison pour interagir avec l'appareil.

```
cd ~/aws-iot-device-sdk-python-v2/samples
python3 shadow.py --ca_file ~/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
```

L'application shadow.py s'exécute ensuite et récupère l'état de l'ombre actuel. Si vous avez supprimé l'ombre ou effacé les états actuels, le programme définit la valeur actuelle `su` et `off` puis vous invite à entrer `undesiredValue`.

```
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:
```

Par contre, 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 `$aws/things/thingName/ombre/obtenir` et `$aws/things/thingName/shadow/get/accepted`.

Supposons que la dernière couleur signalée soit `green`. Voici le contenu de l'`$aws/things/thingName/shadow/get/accepted` 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 une `desired` valeur dans le terminal, telle que `yellow`. Le shadow.py exemple d'application répond et affiche les messages suivants dans le terminal qui indiquent la modification dans le `reported` valeur `yellow`.

```
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'.
```

Dans Client de test MQTT dans la AWS IoT console, sous Subscriptions, vous constatez que les rubriques suivantes ont reçu un message :

- `$aws/things/thingName/ombre/mise à jour`: montre que les deux `desired` et `updated` les valeurs changent en couleur `yellow`.
- `$aws/things/thingName/shadow/update/accepted`: affiche les valeurs actuelles du `desired` et `reported` états et leurs métadonnées et informations de version.
- `$aws/things/thingName/shadow/update/documents`: affiche les valeurs précédentes et actuelles du `desired` et `reported` états et leurs métadonnées et informations de version.

Comme document `$aws/things/thingName/shadow/update/documents` contient également des informations contenues dans les deux autres rubriques, nous pouvons les consulter pour voir les informations d'état. L'état précédent affiche la valeur signalée définie sur `green`, ses métadonnées et ses informations de version, ainsi que l'état actuel qui affiche la valeur signalée mise à jour vers `yellow`.

```
{
    "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 verrez d'autres modifications apportées à la `reported` les valeurs et les mises à jour des messages reçues par ces rubriques. Le numéro de version est également incrémenté de 1. Par exemple, si vous entrez la valeur `green`, l'état précédent indique la valeur `yellow` et l'état actuel indique 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 le AWS IoTconsole Par exemple, vous pouvez modifier ledesiredvaleur de la couleur red. Pour ce faire, dans l'AWS IoTchoisissez, choisissezModifierpuis définissez le paramètredesiredvaleur rouge dans le JSON, tout en conservant la valeurreportedDéfinit la valeurgreen. Avant d'enregistrer la modification, gardez le terminal ouvert car vous verrez le message Delta signalé dans le terminal.

```
{  
  "desired": {  
    "welcome": "aws-iot",  
    "color": "red"  
  },  
  "reported": {  
    "welcome": "aws-iot",  
    "color": "green"  
  }  
}
```

Le shadow .pyl'exemple d'application répond à cette modification et affiche des messages dans le terminal indiquant le delta. Dans le client de test MQTT, leupdateles rubriques auront reçu un message indiquant les modifications apportées a undesiredet reportedvaleurs.

Vous voyez aussi que le sujet\$aws/things/**thingName**/shadow/update/delta a reçu un message. Pour voir le message, choisissez cette rubrique, répertoriée sousSubscriptions.

```
{  
  "version": 13,  
  "timestamp": 1617318480,  
  "state": {  
    "color": "red"  
  },  
  "metadata": {  
    "color": {  
      "timestamp": 1617318480  
    }  
  }  
}
```

Étape 3 : Dépannage des erreurs liées aux interactions Device Shadow

Lorsque vous exécutez l'exemple d'application Shadow, vous risquez de rencontrer des problèmes lors de l'observation des interactions avec le service Device Shadow.

Si le programme s'exécute correctement et vous invite à entrer undesired, vous devriez pouvoir observer les interactions Device Shadow à l'aide du document Shadow et du client de test MQTT comme décrit précédemment. Toutefois, 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 le AWS 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 l'objet dans le AWS IoTconsole. Vous pouvez également vérifier si vous avez une ombre classique en choisissant la ressource de votre objet, puis en choisissantShadows. Ce tutoriel se concentre principalement sur les interactions avec l'ombre classique.

Vous pouvez également vérifier que l'appareil que vous avez utilisé est connecté à Internet. Dans AWS IoTconsole, choisissez l'objet que vous avez créé précédemment, puis choisissezInteragir. Sur la page

de détails de l'objet, vous devez voir ici un message indiquant :*This thing already appears to be connected.*

- Vérifiez les sujets réservés MQTT auxquels 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/**thingName**/shadow/et pourrait avoir update, get, ou delete suivant le suivant en fonction des actions que vous souhaitez effectuer sur l'ombre. Ce didacticiel utilise \$aws/things/**thingName**/ombre/#assurez-vous donc de l'avoir correctement saisi lorsque vous vous abonnez au sujet dans le Filtre de rubriques section du client de test.

Lorsque vous entrez le nom du sujet, assurez-vous que le **thingName** est le même que le nom du AWS IoT Ce que vous avez créé précédemment. 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 au sujet \$aws/things/**thingName**/shadow/update/accepted pour recevoir un message chaque fois qu'une demande de mise à jour échoue afin que vous puissiez déboguer les problèmes de connexion. Pour plus d'informations sur les rubriques réservées, consultez [the section called "Rubriques de shadow" \(p. 128\)](#) et [Rubriques MQTT de Device Shadow \(p. 721\)](#).

Étape 4 : Passez en revue les résultats et les prochaines étapes

Dans ce didacticiel, vous avez appris à :

- Utilisation de l'shadow .py exemple d'application 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 comment le shadow .py l'exemple d'application y répond.
- Utilisez le client de test MQTT pour vous abonner à des rubriques instantanées 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 au sujet \$aws/things/**thingName**/shadow/update/accepted, vous ne verrez que les informations d'état actuel lorsqu'une mise à jour est exécutée avec succès.

Vous pouvez également vous abonner à d'autres rubriques d'ombre pour déboguer les problèmes ou en savoir plus sur les interactions Device Shadow et également déboguer tout problème lié aux interactions Device Shadow. Pour plus d'informations, consultez [the section called "Rubriques de shadow" \(p. 128\)](#) et [Rubriques MQTT de Device Shadow \(p. 721\)](#).

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 d'état à l'aide des messages envoyés depuis le terminal.

Pour plus d'informations sur le service Device Shadow et l'utilisation du service dans les appareils, les applications et les services, reportez-vous à la section [Service AWS IoT Device Shadow \(p. 690\)](#), [Utilisation des shadows sur les appareils \(p. 694\)](#), et [Utilisation des shadows dans les applications et les services \(p. 697\)](#).

Didacticiel : Création d'un outil d'autorisation personnalisé pour AWS IoT Core

Ce didacticiel explique les étapes à suivre pour créer, valider et utiliser l'authentification personnalisée à l'aide du AWS CLI. En option, à l'aide de ce didacticiel, vous pouvez utiliser Postman pour envoyer des données à AWS IoT Core l'aide de l'API HTTP Publish.

Ce didacticiel explique comment créer un exemple de fonction Lambda qui implémente la logique d'autorisation et d'authentification, ainsi qu'un autorisateur personnalisé utilisant l'create-authorizer rappel avec la signature par jeton activée. L'autorisateur est ensuite validé à l'aide du test-invoker-authorizer, et vous pouvez enfin envoyer des données à une rubrique AWS IoT Core de test MQTT à l'aide de l'API HTTP Publish. Un exemple de demande spécifiera l'autorisateur à invoquer à l'aide de l'x-amz-customauthorizer-name en-tête et transmettra les en-têtes de demande token-key-name et x-amz-customauthorizer-signature in.

Ce que vous allez apprendre dans ce didacticiel :

- Comment créer une fonction Lambda en tant que gestionnaire d'autorisation personnalisé
- Comment créer un système d'autorisation personnalisé à l'aide de la fonction de signature AWS CLI par jeton activée
- Comment tester votre système d'autorisation personnalisé à l'aide de latest-invoker-authorizer commande
- Comment publier une rubrique MQTT à l'aide de [Postman](#) et valider la demande avec votre autorisation personnalisée

Ce didacticiel vous prendra environ 60 minutes.

Dans le présent didacticiel, vous effectuerez les tâches suivantes :

- [Étape 1 : Créez une fonction Lambda pour votre autorisation personnalisée \(p. 266\)](#)
- [Étape 2 : Créez une paire de clés publique et privée pour votre autorisation personnalisée \(p. 268\)](#)
- [Étape 3 : Créez une ressource d'autorisation client et de son autorisation \(p. 269\)](#)
- [Étape 4 : Testez l'autorisation en appelant test-invoker-authorizer \(p. 272\)](#)
- [Étape 5 : Test de publication d'un message MQTT avec Postman \(p. 273\)](#)
- [Étape 6 : Afficher les messages dans le client de test MQTT \(p. 275\)](#)
- [Étape 7 : Passez en revue les résultats et les étapes suivantes \(p. 276\)](#)
- [Étape 8 : Nettoyez \(p. 276\)](#)

Avant de commencer ce didacticiel, assurez-vous de disposer des éléments suivants :

- [Configurez votre Compte AWS \(p. 19\)](#)

Vous aurez besoin de votre AWS IoT console et de votre console pour suivre ce didacticiel.

Le compte que vous utilisez pour ce didacticiel fonctionne de manière optimale s'il inclut au moins les politiques AWS gérées suivantes :

- [IAMFullAccess](#)
- [AWSIoTFullAccess](#)
- [AWSLambda_FullAccess](#)

Important

Les politiques IAM utilisées dans ce didacticiel sont plus permissives que celles que vous devriez suivre dans une implémentation de production. Dans un environnement de production,

assurez-vous que vos politiques de compte et de ressources n'accordent que les autorisations nécessaires.

Lorsque vous créez des stratégies IAM pour la production, déterminez l'accès dont les utilisateurs et les rôles ont besoin, puis élaborez des stratégies leur permettant de réaliser uniquement ces tâches.

Pour de plus amples informations, veuillez consulter [Bonnes pratiques de sécurité dans IAM](#)

- A installé le AWS CLI

Pour plus d'informations sur l'installation de AWS CLI, consultez la section [Installation de l'AWS interface de ligne](#) de commande. Ce didacticiel nécessite une AWS CLI version aws-cli/2.1.3 Python/3.7.4 Darwin/18.7.0 exe/x86_64 ou une version ultérieure.

- Outils OpenSSL

Les exemples de ce didacticiel utilisent [LibreSSL 2.6.5](#). Vous pouvez également utiliser les outils [OpenSSL v1.1.1i](#) pour ce didacticiel.

- A revu la [AWS Lambda](#) vue d'ensemble

Si vous ne l'avez jamais utilisé AWS Lambda auparavant, consultez la [AWS Lambda](#) section [Commencer à utiliser Lambda](#) pour en apprendre les termes et les concepts.

- A examiné comment créer des demandes dans Postman

Pour de plus amples informations, veuillez consulter [Création de demandes](#).

- Autorisateurs personnalisés supprimés du didacticiel précédent

Vous ne pouvez configurer qu'un nombre limité d'autorisations personnalisées à la fois. Pour plus d'informations sur la suppression d'un autorisation personnalisée, consultez la section [the section called "Étape 8 : Nettoyer" \(p. 276\)](#).

Étape 1 : Créer une fonction Lambda pour votre autorisation personnalisée

L'authentification personnalisée AWS IoT Core utilise les [ressources d'autorisation](#) que 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 aux AWS IoT ressources AWS IoT Core et y accèdent.

La fonction Lambda effectue les opérations suivantes :

- Si une demande provient test-invoke-authorizer, elle renvoie une politique IAM avec uneDeny action.
- Si une demande provient de Passport via HTTP et que le actionToken paramètre a une valeur deallow, il renvoie une politique IAM avec uneAllow action. Dans le cas contraire, il renvoie une politique IAM avec uneDeny action.

Pour créer la fonction Lambda pour votre autorisation personnalisée

1. Dans la console [Lambda](#), ouvrez [Functions](#).
2. Sélectionnez Create function (Créer une fonction).
3. Confirmer que l'auteur à partir de zéro est sélectionné.
4. Sous Basic information :
 - a. Sous Nom de la fonction, entrez **custom-auth-function**.
 - b. Dans Runtime, confirmez Node.js 18.x
5. Sélectionnez Create function (Créer une fonction).

Lambda crée une fonction Node.js et un [rôle d'exécution](#) qui accorde à la fonction l'autorisation de charger 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 le AWS kit SDK et lire les données à partir des sources d'événements.

6. Pour voir le code et la configuration de la fonction dans l'[AWS Cloud9](#)éditeur, choisissez custom-auth-functiondans la fenêtre du concepteur, puis choisissez index.js dans 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 l'[AWS Cloud9](#)éditeur pour modifier votre fonction aussi longtemps que votre code source ne dépasse pas 3 Mo.

7. Remplacez le code index.js dans l'éditeur par 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 = async (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);

    //Get the dynamic region from function's ARN to be used
    // as full resource for IAM policy
    var REGION = context.invokedFunctionArn.split(":")[3];
    console.log("REGION---"+REGION);

    //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,REGION));

    } 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,REGION));
        }
    }
}
```

```
        }else{
            callback(null, generateAuthResponse(DENY_ACTION,ACCOUNT_ID,REGION));
        }
    };
}

// Helper function to generate the authorization IAM response.
var generateAuthResponse = function(effect,ACCOUNT_ID,REGION) {
    var full_resource = "arn:aws:iot:"+ REGION + ":" + ACCOUNT_ID + ":*";
    console.log("full_resource---"+full_resource);

    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. After Changes deployed apparaît au-dessus de l'éditeur :
 - a. Accédez à la section Présentation des fonctions située au-dessus de l'éditeur.
 - b. Copiez la fonction ARN et enregistrez-la pour l'utiliser plus tard dans ce didacticiel.
10. Testez votre fonction .
 - a. Choisissez l'onglet Test.
 - b. À l'aide des paramètres de test par défaut, choisissez Invoke.
 - c. Si le test a réussi, dans les résultats de l'exécution, ouvrez la vue Détails. Vous devriez voir le document de politique renvoyé par la fonction.

Si le test a échoué ou si aucun document de politique n'est affiché, consultez le code pour rechercher et corriger les erreurs.

Étape 2 : Créez une key pair publique et privée pour votre 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 publique et privée 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.

Étape 3 : Création d'une ressource d'autorisation client et de son autorisation

L'autorisateur AWS IoT personnalisé est la ressource qui relie tous les éléments créés lors des 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 la AWS IoT console AWS CLI, de l'AWS API.

Pour ce didacticiel, vous n'avez besoin de créer qu'un seul autorisateur personnalisé. Cette section explique comment créer à l'aide de la AWS IoT console et du AWS CLI, afin que vous puissiez utiliser la méthode qui vous convient le mieux. Il n'y a aucune différence entre les ressources d'autorisation personnalisées créées par l'une ou l'autre méthode.

Création d'une ressource d'autorisation client

Choisissez l'une de ces options pour créer votre ressource d'autorisation personnalisée

- [Création d'un système d'autorisation personnalisé à l'aide de la AWS IoT console \(p. 269\)](#)
- [Créez un système d'autorisation personnalisé à l'aide du AWS CLI \(p. 270\)](#)

Pour créer un système d'autorisation personnalisé (console)

1. Ouvrez la [page d'autorisation personnalisée de la AWS IoT console](#), puis choisissez Créez un mécanisme d'autorisation.
2. Dans Create Authorizer :
 - a. Dans Nom de l'autorisateur, entrez **my-new-authorizer**.
 - b. Dans État de l'autorisateur, cochez Actif.
 - c. Dans la fonction Authorizer, choisissez la fonction Lambda que vous avez créée précédemment.
 - d. Dans Validation des jetons, facultative :
 - i. Activez la validation des jetons.

- ii. Dans Nom de la clé de jeton, entrez **tokenKeyName**.
- iii. Sélectionnez Ajouter une clé.
- iv. Dans Nom de la clé, entrez **FirstKey**.
- v. Dans Clé publique, entrez le contenu du `public-key.pem` fichier. Veillez à inclure les lignes du fichier avec -----BEGIN PUBLIC KEY-----END PUBLIC KEY----- et, sans ajouter ni supprimer de fils de ligne, de retours de chariot ou d'autres caractères du contenu du fichier. La chaîne que vous entrez doit se présenter comme suit.

```
-----BEGIN PUBLIC KEY-----  
MIICIJANBqkqhkiG9w0BAQEEAAOCg8AMICCgKCAgEAvEBz0k4vhN+3Lgs1vEWt  
sLCqNmt5Damas3bmiTRvg2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z  
QPu/vQ0e5tyzz1MsKdmfFGxMqQ3qjEXAMPLE0mqyUKPP5mff58k6ePSfxAnzBH0q  
lg2HioefrpU50SANpuRAjYKofKjbc2Vrn6N2G7hV+IFTBvCE1f0csa1S/Rk4phD5  
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8  
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshT/F1LVCS5+v8AQ8UGGDfZmv  
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2S8Lgzpey9N50N1o1Cvldwvc  
KrJJtgwW6hVqRGuShownLpgG86M6neZ5sRmbVNz080zcobLngJ0Ibw9KkcUdk1W  
gvZ6HEJqBY2XE70iEXAMPLETPHzqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa  
/NN7l7zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJ1Uzn62Q+VeNV2tdA7MfpfMC  
8btGYladFAnitThaz6+F0VSBJPu7pZqoLnqyEp5zLmtF+kFl2y0BmGAP0RBivRd9  
JWBUCG0bqcLQPeQyjbXS0fUCAwEAAQ==  
-----END PUBLIC KEY-----
```

3. Choisissez Créer un autorisateur.
4. Si la ressource d'autorisation personnalisée a été créée, vous verrez la liste des autorisations personnalisées et votre nouvel autorisateur personnalisé devrait apparaître dans la liste. Vous pouvez passer à la section suivante pour le tester.

Si une erreur s'affiche, vérifiez-la, essayez de créer à nouveau votre autorisation personnalisée et revérifiez les entrées. Notez que chaque ressource d'autorisation personnalisée doit porter un nom unique.

Pour créer un système d'autorisation personnalisé (AWS CLI)

1. Remplacez vos valeurs para `authorizer-name`, `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:Region:57EXAMPLE833:function:custom-auth-  
function" \  
--token-signing-public-keys FirstKey="-----BEGIN PUBLIC KEY-----  
MIICIJANBqkqhkiG9w0BAQEEAAOCg8AMICCgKCAgEAvEBz0k4vhN+3Lgs1vEWt  
sLCqNmt5Damas3bmiTRvg2gjRJ6KXGTGQChqArAJwL1a9dkS9+maaXC3vc6xzx9z  
QPu/vQ0e5tyzz1MsKdmfFGxMqQ3qjEXAMPLE0mqyUKPP5mff58k6ePSfxAnzBH0q  
lg2HioefrpU50SANpuRAjYKofKjbc2Vrn6N2G7hV+IFTBvCE1f0csa1S/Rk4phD5  
oa4Y0GHISRnevypg5C8n9Rrz91PWGqP6M/q5DNJJXjMyleG92hQgu1N696bn5Dw8  
FhedszFa6b2x6xrItZFzewNQkPMLMFhNrQIIyvshT/F1LVCS5+v8AQ8UGGDfZmv  
QeqAMAF7WgagDMXcfgKSVU8yid2sIm56qsCLMvD2S8Lgzpey9N50N1o1Cvldwvc  
KrJJtgwW6hVqRGuShownLpgG86M6neZ5sRmbVNz080zcobLngJ0Ibw9KkcUdk1W  
gvZ6HEJqBY2XE70iEXAMPLETPHzqvK6Ei1HGxpHsXx6BNft582J1VpgYjXha8oa  
/NN7l7zbj/euAb41IVtmX8JrD9z613d1iM5L8HluJ1Uzn62Q+VeNV2tdA7MfpfMC  
8btGYladFAnitThaz6+F0VSBJPu7pZqoLnqyEp5zLmtF+kFl2y0BmGAP0RBivRd9  
JWBUCG0bqcLQPeQyjbXS0fUCAwEAAQ==  
-----END PUBLIC KEY-----"
```

Où :

- La `authorizerArn` valeur est l'Amazon Resource Name (ARN) de la fonction Lambda que vous avez créée pour votre autorisation personnalisée.
- La `token-signing-public-keys` valeur inclut le nom de la `FirstKey` clé et le contenu `public-key.pem` fichier. Veillez à inclure les lignes du fichier avec-----BEGIN PUBLIC KEY-----END PUBLIC KEY----- et, sans ajouter ni supprimer de fils de ligne, de retours de chariot ou d'autres caractères du contenu du fichier.

Remarque : soyez prudent lors de la saisie de la clé publique, car toute modification de la valeur de la clé publique la rend inutilisable.

2. Si l'autorisateur personnalisé est créé, la commande renvoie le nom et l'ARN de la nouvelle ressource, tels que les suivants.

```
{  
    "authorizerName": "my-new-authorizer",  
    "authorizerArn": "arn:aws:iot:Region:57EXAMPLE833:authorizer/my-new-authorizer"  
}
```

Enregistrez `authorizerArn` valeur pour l'utiliser lors de l'étape suivante.

N'oubliez pas que chaque ressource d'autorisation personnalisée doit porter un nom unique.

Autoriser la ressource d'autorisation personnalisée

Dans cette section, vous allez autoriser la ressource d'autorisation personnalisée que vous venez de créer à exécuter la fonction Lambda. Pour accorder l'autorisation, vous pouvez utiliser la commande CLI [add-permission](#).

Octroi de l'autorisation à votre fonction Lambda à l'aide AWS CLI

1. Après avoir inséré vos valeurs, entrez la commande suivante. Notez que la `statementId` valeur doit être unique. **Id-1234** Remplacez-le par une autre valeur si vous avez déjà exécuté ce didacticiel ou si vous obtenez une `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 aboutit, elle renvoie une déclaration d'autorisation, comme dans 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:Region:57EXAMPLE833:function:custom-auth-function\", \"Condition\": {\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:lambda:Region:57EXAMPLE833:function:custom-auth-function\"}}}"  
}
```

Si la commande échoue, elle renvoie une erreur, comme dans cet exemple. Vous devez 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:Region:57EXAMPLE833:function:custom-auth-function
```

Étape 4 : tester l'autorisation en appelant test-invoke-authorizer

Toutes les ressources étant définies, dans cette section, vous allez appeler test-invoke-authorizer depuis la ligne de commande pour tester la passe d'autorisation.

Notez que lorsque vous appelez l'autorisateur à partir de la ligne de commande, protocolData il n'est pas défini, de sorte que l'autorisateur renverra toujours un document DENY. Ce test confirme toutefois que votre système d'autorisation personnalisé et votre fonction Lambda sont correctement configurés, même s'il ne teste pas complètement la fonction Lambda.

Pour tester votre système d'autorisation personnalisé et sa fonction Lambda à l'aide du AWS CLI

1. Dans le répertoire où se trouve le private-key.pem fichier que vous avez créé lors de l'étape précédente, exécutez la commande suivante.

```
echo -n "tokenKeyValue" | openssl dgst -sha256 -sign private-key.pem | openssl base64 -A
```

La commande crée une chaîne de signature à utiliser lors de l'étape suivante. La chaîne de signature ressemble à ceci :

```
dBwykzl1b+fo+JmSGdwoGr8dyC2qB/IyLefJJr+rbCvmu9J14KHA9DG+V  
+MMWu09YSA86+64Y3Gt4t0ykZqn9mn  
VB1wyxp+0bDZh8hmquAUH3fwi3fpjBvCa4cwNuLQnqBZzbCvs1uv7i2IMjEg  
+CPY0zrWt1jr9BikgGPDXWkjaceeh  
bQHHTo357TegKs9pP30Uf4TrxyNmFswA5k7QIC01n4bIyRTm900yZ94R4bdJsHNig1JePgnu0BvMGCEFE09jGjj  
szEHfgAUAIWXiVGQj16BU1xKpTGSiTAWheLKUjIT0EXAMPECK3aHKYKY  
+d1vTvdthKtYHBq8MjhzJ0kggbt29V  
QJCb8Ri1N/P5+vcVniSXWPp1yB5jkYs9UvG08REoy64AtizfUhvSul/r/F3VV8ITtQp3aXiUtcspACi6ca  
+tsDuX  
f3LzCwQF/YSu02u5XkWh+sto6KCkpNlkD0wU8gl3+k0zxrthnQ8gEajd5Iylx230iqcXo3osjPha7JDyWM5o  
+K  
EWckTe911mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPELotyh7h  
+f1FeloZlAWQFH  
xR1XsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBI0gokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Copie de cette chaîne de signature pour l'utiliser lors de l'étape suivante. Veillez à ne pas inclure de caractères supplémentaires ou à ne pas en omettre.

2. Dans cette commande, remplacez la token-signature valeur par la chaîne de signature de l'étape précédente et exécutez cette commande pour tester votre autorisateur.

```
aws iot test-invoke-authorizer \  
--authorizer-name my-new-authorizer \  
--token tokenKeyValue \  
--token-signature dBwykzl1b+fo+JmSGdwoGr8dyC2qB/IyLefJJr+rbCvmu9J14KHA9DG+V...xR1XsPqiVKS1ZIUClaZWprh/orDJplpiWfBgBI0gokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

```
+KEWckTe91I1mokDr5sJ4JXixvnJTVSx1li49IalW4en1DAkc1a0s2U2UNm236EXAMPLElotyh7h  
+f1FeloZLAWQFHxRlxspqivKS1ZIUClazWprh/orDjplpiWfBgbIOgokJIDGP9gwhXIIk7zWrGmWpMK9o=
```

Si la commande aboutit, elle renvoie les informations générées par votre fonction d'autorisation client, comme dans cet exemple.

```
{  
    "isAuthenticated": true,  
    "principalId": "principalId",  
    "policyDocuments": [  
        {"\\"Version\\":\\"2012-10-17\\",\\"Statement\\": [{"\\\"Action\\":\\"iot:*\\",\\"Effect\\": \\"Deny\\",\\"Resource\\":\\"arn:aws:iot:Region:57EXAMPLE833:*\\"}]}]  
    ],  
    "refreshAfterInSeconds": 600,  
    "disconnectAfterInSeconds": 3600  
}
```

Si la commande renvoie une erreur, examinez l'erreur et revérifiez les commandes que vous avez utilisées dans cette section.

Étape 5 : Test de publication d'un message MQTT avec Postman

1. Pour obtenir le point de terminaison des données de votre appareil à partir de la ligne de commande, appelez [describe-endpoint](#) comme indiqué ici

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

Enregistrez cette adresse pour l'utiliser en tant que [*device_data_endpoint_address*](#) lors d'une étape ultérieure.

2. Ouvrez une nouvelle fenêtre Postman et créez une nouvelle requête HTTP POST.
 - a. Depuis votre ordinateur, ouvrez l'application Postman.
 - b. Dans Postman, dans le menu Fichier, choisissez Nouveau... .
 - c. Dans la boîte de dialogue Nouveau, choisissez Request.
 - d. Dans Enregistrer la demande,
 - i. Dans Nom de la demande, entrez **Custom authorizer test request**.
 - ii. Dans Sélectionnez une collection ou un dossier dans lequel enregistrer : choisissez ou créez une collection dans laquelle enregistrer cette demande.
 - iii. Choisissez Enregistrer dans **collection_name**.
3. Créez la requête POST pour tester votre système d'autorisation personnalisé.
 - a. Dans le sélecteur de méthode de demande situé à côté du champ URL, choisissez POST.
 - b. Dans le champ URL, créez l'URL de votre demande en utilisant l'URL suivante avec l'adresse [*device_data_endpoint_address*](#) de la commande [describe-endpoint utilisée](#) lors d'une étape précédente.

```
https://device\_data\_endpoint\_address:443/topics/test/cust-auth/topic?  
qos=0&actionToken=allow
```

Notez que cette URL inclut le paramètre `deactionToken=allow` requête qui indiquera à votre fonction Lambda de renvoyer un document de politique autorisant l'accès à AWS IoT. Une fois que vous avez saisi l'URL, les paramètres de la requête apparaissent également dans l'onglet Paramètres de Postman.

- c. Dans l'onglet Auth, dans le champ Type, choisissez No Auth.
- d. Dans l'onglet Headers :
 - i. Si une clé Host 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 la **Host** valeur par votre adresse **device_data_endpoint_address** et **lax-amz-customauthorizer-signature** valeur par la chaîne de signature que vous avez utilisée avec latest-invoke-authorize commande dans la section précédente.

Clé	Valeur
x-amz-customauthorizer-name	my-new-authorizer
Host	<i>adresse de terminaison des données de l'appareil</i>
tokenKeyName	tokenKeyValue
x-amz-customauthorizer-signature	<i>DBWlykZLB+F0+JMSGdwoGr 8dyC2QB/IyLef jjR+rbcVmU9JL4khAA9DG+V+MMWU09Ysa86+64Y3Gt4t0ykpzqn9MnVB1WypXP+0bdZh8HmQuAuH3FWi3fPjBvCa4CwnulQNQBZzbCvsluv 7i2iMjEg+Cpy0ZRWT1J9bIkKgGPDxWkjacehbQHhTo357TegKs 9pP30uF4TrxypNmFswA5k7qIC01n4biyrTM90oyZ94R4BDJShNiG1JePgnuObVMGCeFe09JGJSzeHFGauAqIWXivGQJ16Bu1xKPTGS1ExemplecK3ahKyKy+D1vTvdtKtyHBQ8MjHZJ0kGGbt29vqjcb8rirln/p5+p+VCVNISXWPPLyB5JKYS9UVG08RE0Y64AtizfUhvSul/r/F3VV8iTtQp 3aXiUtcspACi6ca+tsDuXf 3LzCw QQF/YSUY02u5XkWn +sT06kCkpNlkD0WU8gL3+KoZXrThNQ8GeAjD5iYLX230iQcx03+KEWckTe91i1mokdr5sj4jxixVNJtVsx1Li49iAlW4en1daKC1A0exempleLotoH7H+ AwQFvks1ZiucLaZWWWW50+kE 91i1m0kDr5Sj4JxH7H+flFeloZl AwQFHxRLXsPqiVks1ZiucLaZWWwA9A0S2u2UnM236PRH/ORDJplpiWfBgBioGokjIDgP9GWhXiIk7zWrGm wPmk9o=</i>

- e. Dans l'onglet Body :
 - i. Dans la zone d'option de format de données, choisissez Raw.
 - ii. Dans la liste des types de données, sélectionnez JavaScript.
 - iii. Dans le champ de texte, saisissez cette charge de message JSON pour votre message de test :

```
{
  "data_mode": "test",
  "vibration": 200,
  "temperature": 40
```

}

4. Sélectionnez Envoyer pour envoyer la demande.

Si la demande a abouti, elle renvoie :

```
{  
    "message": "OK",  
    "traceId": "ff35c33f-409a-ea90-b06f-fbEXAMPLE25c"  
}
```

La réponse positive indique que votre autorisateur personnalisé a autorisé la connexion à Broker inAWS IoT et que le message de test a été envoyé au courtier AWS IoT Core.

S'il renvoie une erreur, consultez le message d'erreur, le *device_data_endpoint_address*, la chaîne de signature et les autres valeurs d'en-tête.

Conservez cette demande dans Postman pour l'utiliser lors de la section suivante.

Étape 6 : Afficher les messages dans le client de test MQTT

À l'étape précédente, vous avez envoyé des messages d'appareil AWS IoT simulés à l'aide de Postman. La réponse positive indique que votre système d'autorisation personnalisé a autorisé la connexion à Broker inAWS IoT et que le message de test a été envoyé au courtier AWS IoT Core. Dans cette section, vous allez utiliser le client de test MQTT de la AWS IoT console pour voir le contenu du message, comme d'autres appareils et services peuvent le faire.

Pour voir les messages de test autorisés par votre système d'autorisation personnalisé

1. Dans la AWS IoT console, ouvrez le [client de test MQTT](#).
2. Dans l'onglet S'abonner à la rubrique, dans le filtre Rubrique **test/cust-auth/topic**, entrez, qui est le sujet du message utilisé 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, choisissez Envoyer.

Passez en revue la réponse pour vous assurer qu'elle a bien été prise en compte. Si ce n'est pas le cas, résolvez l'erreur comme décrit dans la section précédente.

5. Dans le client de test MQTT, vous devriez voir une nouvelle entrée indiquant le sujet du message et, s'il est développé, la charge utile du message provenant de la demande que vous avez envoyée depuis Postman.

Si vous ne voyez pas vos messages dans le client de test MQTT, voici quelques points à vérifier :

- Assurez-vous que votre demande Postman a été renvoyée avec succès. Si AWS IoT la connexion est rejetée et renvoie une erreur, le message contenu dans la demande n'est pas transmis au courtier de messages.
- Assurez-vous que le Compte AWS et Région AWS utilisé pour ouvrir la AWS IoT console sont les mêmes que ceux que vous utilisez dans l'URL de Postman.
- Assurez-vous d'avoir correctement saisi le sujet dans le client de test MQTT. Le filtre de rubrique est sensible à la casse. En cas de doute, vous pouvez également vous abonner à la # rubrique, qui contient tous les messages MQTT qui passent par le courtier de messages.
- Assurez-vous que le Compte AWS et Région AWS utilisés pour ouvrir la AWS IoT console sont utilisés pour ouvrir la AWS IoT console.

Étape 7 : Passez en revue les résultats et les étapes suivantes

Dans ce didacticiel :

- Vous avez créé une fonction Lambda en tant que gestionnaire d'autorisation personnalisé
- Vous avez créé un système d'autorisation personnalisé avec la signature par jeton activée
- Vous avez testé votre système d'autorisation personnalisé à l'aide de `latest-invoke-authorizer` commande
- Vous avez publié une rubrique MQTT à l'aide de [Postman](#) et vous avez validé la demande à l'aide de votre autorisateur personnalisé
- Vous avez utilisé le client de test MQTT pour afficher les messages envoyés depuis votre test Postman

Étapes suivantes

Après avoir envoyé des messages depuis Postman pour vérifier que l'autorisateur personnalisé fonctionne, essayez de faire des essais pour voir comment la modification des différents aspects de ce didacticiel affecte les résultats. Voici quelques exemples pour vous aider à démarrer.

- Modifiez la chaîne de signature afin qu'elle ne soit plus valide pour voir comment les tentatives de connexion non autorisées sont gérées. Vous devriez obtenir une réponse d'erreur, comme celle-ci, et le message ne devrait pas apparaître dans le client de test MQTT.

```
{  
    "message": "Forbidden",  
    "traceId": "15969756-a4a4-917c-b47a-5433e25b1356"  
}
```

- Pour en savoir plus sur la manière de détecter les erreurs susceptibles de se produire lors du développement et de l'utilisation de AWS IoT règles, consultez [Surveillance des AWS IoT \(p. 466\)](#).

Étape 8 : Nettoyer

Si vous souhaitez répéter ce didacticiel, vous devrez peut-être supprimer certains de vos autorisateurs personnalisés. Vous neCompte AWS pouvez configurer qu'un nombre limité d'autorisations personnalisées à la fois et vous pouvez en obtenir uneLimitExceededException lorsque vous essayez d'en ajouter une nouvelle sans supprimer une autorisation personnalisée existante.

Pour supprimer un système d'autorisation personnalisé (console)

1. Ouvrez la [page Autorisation personnalisée de la AWS IoT console](#) et, dans la liste des autorisations personnalisées, recherchez l'autorisation personnalisée à supprimer.
2. Ouvrez la page de détails de l'autorisateur personnalisé et, dans le menu Actions, choisissez Modifier.
3. Décochez l'option Activer l'autorisation, puis choisissez Mettre à jour.

Vous ne pouvez pas supprimer un système d'autorisation personnalisé lorsqu'il est actif.

4. Sur la page de détails de l'autorisateur personnalisé, ouvrez le menu Actions et choisissez Supprimer.

Pour supprimer un système d'autorisation personnalisé (AWS CLI)

1. Répertoriez les autorisations personnalisées que vous avez installées et recherchez le nom de l'autorisation personnalisée à supprimer.

```
aws iot list-authorizers
```

2. Définissez l'autorisateur personnalisé *surinactive* en exécutant cette commande après l'avoir *Custom_Auth_Name* remplacé par *authorizerName* celui de l'autorisateur personnalisé à supprimer.

```
aws iot update-authorizer --status INACTIVE --authorizer-name Custom_Auth_Name
```

3. Supprimez l'autorisateur personnalisé en exécutant cette commande après l'avoir *Custom_Auth_Name* remplacé par *authorizerName* l'autorisateur personnalisé à supprimer.

```
aws iot delete-authorizer --authorizer-name Custom_Auth_Name
```

Tutoriel : Surveillance de l'humidité du sol avec unAWS IoT 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 règleAWS IoT qui envoie un e-mail à une adresse abonnée à une rubrique Amazon SNS lorsque le niveau d'humidité tombe en dessous d'un seuil.

Note

Ce didacticiel n'est peut-être pas à jour. Certaines références ont peut-être été remplacées depuis la publication initiale de ce sujet.

Table des matières

- [Prérequis \(p. 277\)](#)
- [Configuration de AWS IoT \(p. 278\)](#)
 - [Étape 1 : Créer la stratégie AWS IoT \(p. 278\)](#)
 - [Étape 2 : Création de l'AWS IoT objet, du certificat et de la clé privée \(p. 279\)](#)
 - [Étape 3 : créer une rubrique Amazon SNS s'y abonner \(p. 280\)](#)
 - [Étape 4 : créer uneAWS IoT règle pour envoyer un e-mail \(p. 280\)](#)
- [Configuration de votre Raspberry Pi et du capteur d'humidité \(p. 281\)](#)

Prérequis

Pour suivre ce didacticiel, vous devez disposer des éléments suivants :

- Un Compte AWS.
- Utilisateur IAM disposant de droits d'administrateur.
- Un ordinateur de développement exécutant Windows, macOS, Linux ou Unix pour accéder à la [console AWS IoT](#).
- Un [Raspberry Pi 3B ou 4B](#) exécutant le dernier système d'[exploitation 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 de 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 politique 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 IoT avec lequel ils communiquent. Pour plus d'informations, veuillez consulter [Authentification du serveur \(p. 317\)](#).

- 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 Amazon SNS s'y abonner.

La règle écoute les données d'humidité de votre Raspberry Pi. Si la valeur est inférieure à un seuil, un message est envoyé à la rubrique Amazon SNS. Amazon SNS envoie ce message à toutes les adresses e-mail abonnées à la rubrique.

Étape 1 : Créer 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. Sinon, dans le volet de navigation, développez Sécurité, puis choisissez Politiques.
2. Si une boîte de dialogue Vous ne possédez pas encore de stratégie s'affiche, choisissez Créez 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 **la région** et le **compte** par votreCompte AWS numéroRégion AWS et.

```
{  
    "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 Create (Créer).

Étape 2 : Création de l'AWS IoTobjet, 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 Crédit d'objets AWS IoT, choisissez Crédit 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, choisissez Activer. Le certificat doit être actif pour qu'un appareil puisse s'y connecterAWS IoT.
8. Choisissez Attacher une stratégie.
9. Pour Ajouter une politique pour votre objet, choisissez MoistureSensorPolicy, puis choisissez Enregistrer l'objet.

Étape 3 : créer une rubrique Amazon SNS s'y abonner

Créez une rubrique Amazon SNS s'y abonner.

1. Dans la [consoleAWS SNS](#), dans le volet de navigation, choisissez Rubriques, puis choisissez Créer 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 (Rubriques) dans la console Amazon SNS.
4. Choisissez Create topic (Créer une rubrique).
5. Sur la page détaillée de la rubrique Amazon SNS, choisissez Créez un abonnement.
6. Pour Protocole, choisissez E-mail.
7. Saisissez votre adresse e-mail dans Endpoint (Point de terminaison).
8. Choisissez Create subscription (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 provenant 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.

Étape 4 : créer uneAWS IoT règle 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, veuillez consulter [Règles pour AWS IoT \(p. 524\)](#).

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 collecte les données et les envoie aux abonnés aux rubriques Amazon SNS si le taux d'humidité est inférieur à une valeur seuil.

créer 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 Créez une règle. Sinon, cliquez sur Create.
2. Sur la page Créez 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 le code exécuté 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 l'action, pour la cible SNS, choisissez Sélectionner, puis choisissez 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 Créez un rôle. Entrez un nom pour le rôle (par exemple, **LowMoistureTopicRole**), puis choisissez Créez un rôle.
11. Choisissez Add action.
12. Choisissez Create rule (Créez une règle).

Configuration de votre Raspberry Pi et du capteur d'humidité

Insérez votre carte microSD dans le Raspberry Pi, connectez votre moniteur, 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 :

- 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 Next (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. Sélectionnez OK.

Les bibliothèques du capteur d'humidité Adafruit STEMMA ont été conçues 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 de CircuitPython bibliothèques 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 :

point de terminaison

Votre point de terminaison AWS IoT personnalisé Pour plus d'informations, veuillez consulter [API REST Device Shadow \(p. 716\)](#).

rootCA

Le chemin complet vers votre certificat d'autorité de certificationAWS 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 IoT publie un message dans la rubrique Amazon SNS. Vous devez recevoir un e-mail contenant les données relatives à l'humidité et à la température.

Après avoir vérifié la réception des e-mails d'Amazon SNS, appuyez sur CTRL+C pour arrêter le programme Python. Il est peu probable que le programme Python envoie suffisamment de messages pour entraîner des frais, mais il est préférable d'arrêter le programme lorsque 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 d'une 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. Bien que nous n'appliquions pas de correspondance entre le nom de registre d'un objet et son utilisation des ID clients MQTT, des certificats ou de l'état fantôme, nous vous recommandons de choisir un nom d'objet et de l'utiliser comme identifiant client MQTT à la fois 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 un objet dans le registre pour connecter un appareil 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 AWS IoT console, AWS IoT API ou AWS CLI pour 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 les objets de votre objet :

- Vous ne devez pas utiliser d'informations personnelles identifiables dans le nom de votre objet. Le nom de l'objet peut apparaître dans les communications et les rapports non cryptés.

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": "12345678abcdefghijklmnopqrstuvwxyz"  
}
```

Note

Nous ne recommandons pas d'utiliser des informations d'identification personnelle dans les noms de vos objets.

Pour de plus amples informations, veuillez consulter [create-thing](#) à partir du Guide de référence des AWS CLI commandes de l'.

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 laListThings commande pour 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"
    },
{
    "thingType": "LightBulb",
    "attributes": {
        "model": "123",
        "wattage": "75"
    },
    "version": 1,
    "thingName": "MySecondLightBulb"
}
]
```

Vous pouvez utiliser laListThings commande pour rechercher tous les éléments dont l'attribut possède une valeur spécifique. Cette commande recherche jusqu'à trois attributs.

```
$ aws iot list-things --attribute-name "wattage" --attribute-value "75"
```

```
{
    "things": [
        {
            "thingType": "StopLight",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 3,
            "thingName": "MyLightBulb"
        },
        {
            "thingType": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyRGBLight"
        },
        {
            "thingType": "LightBulb",
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MySecondLightBulb"
        }
    ]
}
```

Pour de plus amples informations, veuillez consulter [list-things](#) dans la Référence des AWS CLI commandes.

Décrivez les choses

Vous pouvez utiliser laDescribeThing commande pour 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": "12345678abcdefghijklmnopqrstuvwxyz",  
    "defaultClientId": "MyLightBulb",  
    "thingTypeName": "StopLight",  
    "attributes": {  
        "model": "123",  
        "wattage": "75"  
    }  
}
```

Pour de plus amples informations, veuillez consulter la section [describe-thing](#) de la Référence des AWS CLI commandes de l'.

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"  
}
```

Pour de plus amples informations, veuillez consulter [update-thing](#) dans la Référence des AWS CLI commandes.

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.

Pour de plus amples informations, veuillez consulter [delete-thing](#) dans la Référence des AWS CLI commandes.

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.

Pour plus d'informations, consultez [attach-thing-principal](#) la référence des AWS CLI commandes.

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 DetachThingPrincipal commande ne produit aucune sortie.

Pour plus d'informations, consultez [detach-thing-principal](#) la référence des AWS CLI commandes.

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' LightBulb objet. Tous les éléments associés au type d' LightBulb objet partagent un ensemble d'attributs : numéro de série, fabricant et puissance en watts. Lorsque vous créez un objet de type LightBulb (ou que vous modifiez le type d'un objet existant LightBulb), vous pouvez spécifier des valeurs pour chacun des attributs définis dans le type d' LightBulb objet.

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 le nom d'un type d'objet une fois qu'il a été créé. 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 `ListThingTypes` commande renvoie une liste des types d'objets définis dans votre 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. Elles ne peuvent pas être modifiées après avoir été définies. 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": "12345678abcdefghijklmnopqrstuvwxyz",
"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 \(p. 304\)](#), en revanche, contiennent des objets qui correspondent à une requête spécifiée. Les groupes d'objets statiques peuvent également contenir d'autres groupes d'objets statiques. Vous pouvez 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.

Note

Les politiques relatives aux groupes d'objets n'autorisent pas l'accès aux opérations du plan deAWS IoT Greengrass données. Pour autoriser un objet à accéder à une opération de plan deAWS IoT Greengrass données, ajoutez l'autorisation à uneAWS IoT politique que vous associez au certificat de l'objet. Pour plus d'informations, consultez la section [Authentification et autorisation des appareils](#) dans le guide duAWS IoT Greengrass développeur.

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 descendants 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. Consultez [Configurer la journalisation AWS IoT \(p. 467\)](#).
- Créer des tâches qui sont envoyées vers et exécutées sur chaque objet d'un groupe et de ses groupes enfants. Consultez [Tâches \(p. 739\)](#).

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. Veuillez 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 plus d'un groupe de la même hiérarchie. (En d'autres termes, vous ne pouvez pas ajouter un objet à deux groupes de parents communs.)
- 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 d'objets peut apparaître dans les communications et les rapports non cryptés.

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": "abcdefghijklmnopqrstuvwxyz12345678qrstuvwxyz",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"  
}
```

Note

Nous vous déconseillons d'utiliser des informations personnellement identifiables dans les noms 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": "abcdefghijklmnopqrstuvwxyz",  
    "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": "12345678abcdefghijklmnopqrstuvwxyz",  
    "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. Vous ne pouvez pas ajouter un objet à plus d'un groupe de la même hiérarchie. (En d'autres termes, vous ne pouvez pas ajouter quoi que ce soit à deux groupes qui partagent un parent commun.)

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
```

LaListThingGroups commande renvoie la liste des groupes d'objets de votreCompte 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, laListThingGroups commande renvoie la liste des groupes enfants directs du groupe d'objets défini dans votreCompte 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 `ListThingGroupsForThing` commande pour répertorier les groupes directs auxquels appartient un objet :

```
$ aws iot list-thing-groups-for-thing --thing-name MyLightBulb
```

La `ListThingGroupsForThing` commande renvoie la liste des groupes d'objets directs auxquels cet objet appartient :

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

```
        "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 ne recommandons pas d'utiliser des informations d'identification personnelle dans les noms de vos politiques d'assurance.

Le `--target` paramètre 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 [Authentification \(p. 316\)](#).

Pour plus d'informations, consultez [AWS IoT Core les politiques](#).

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"
```

Le `--target` paramètre 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` paramètre et, éventuellement, ajoutez-le pour spécifier une identité Amazon Cognito.`--principal` 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 `TestAuthorization` commande pour tester si une action [MQTT](#) (`Publish`, `Subscribe`) 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 Amazon Cognito Identity, spécifiez une identité Cognito en tant que paramètre `--principal` ou utilisez le `--cognito-identity-pool-id` paramètre, 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 en savoir plus, consultez [Stratégies AWS IoT Core \(p. 357\)](#).

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. À l'aide de groupes d'objets dynamiques, vous pouvez modifier la façon dont vous interagissez avec les objets en fonction de leur connectivité, de leur registre, de leur ombre ou des données relatives aux violations de Device Defender. Les groupes d'objets dynamiques étant liés à votre index de flotte, vous devez activer l'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 [Indexation de la flotte \(p. 929\)](#) et [Syntaxe de requête \(p. 950\)](#).

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 souhaitez mettre à jour le microprogramme sur vos appareils, mais que, pour minimiser le risque d'interruption de cette mise à jour, vous souhaitez 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](#).

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 \(p. 951\)](#) qui définit l'appartenance au groupe.
- Les groupes d'objets dynamiques ne peuvent pas faire partie d'une hiérarchie.
- Aucune politique ne peut être appliquée aux groupes d'objets dynamiques.
- 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 d'objets peut apparaître dans les communications et les rapports non cryptés.

Pour plus d'informations sur les groupes d'objets statiques, consultez [Groupes d'objets statiques \(p. 294\)](#).

Par exemple, supposons que nous créons un groupe dynamique contenant toutes les salles d'un entrepôt dont la température est supérieure à 60°F. Lorsque la température d'une pièce est supérieure ou égale à 61 degrés, elle est ajoutée au groupe d'objets RoomTooWarm dynamiques. Les ventilateurs de refroidissement sont allumés dans toutes les pièces du groupe d'objets RoomTooWarm dynamiques. 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 personnellement identifiables dans les noms de vos groupes d'objets dynamiques.

La `CreateDynamicThingGroup` commande renvoie une réponse contenant le nom d'index, la chaîne de requête, la version de la requête, le nom du groupe d'objets, l'identifiant du groupe d'objets et le nom de ressource Amazon (ARN) de votre 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": "abcdefghijklmnopqrstuvwxyz"  
}
```

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 [DescribeThingGroup](#) commande.

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 [DescribeThingGroup](#) commande.

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 de groupes d'objets ne peuvent pas contenir de caractères internationaux tels que 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 \(p. 479\)](#).

Une [entrée de journal des CloudWatch erreurs](#) est créée pour chaque objet lorsqu'un objet éligible ne peut pas être ajouté à un groupe d'objets dynamique ou lorsqu'un objet est supprimé d'un groupe d'objets dynamique pour l'ajouter à un autre groupe. Lorsqu'un élément ne peut pas être ajouté à un groupe

dynamique, une [AddThingToDynamicThingGroupsFailed](#) 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.
 - OUI : L'objet est-il 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 une [métrique AddThingToDynamicThingGroupsFailed](#) est générée.
 - OUI : Le groupe d'objets dynamiques à rejoindre est-il plus ancien que les groupes 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 une [métrique AddThingToDynamicThingGroupsFailed](#) est générée.
 - OUI : 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](#) 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 génèrent pas d'entrées dans le 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 à un objet à l'aide des [UpdateThingGroupsForThing](#) commandes [AddThingToThingGroup](#), l'ajout du `--overrideDynamicGroups` paramètre 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.
 - OUI : 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.
 - OUI : Le paramètre `--overrideDynamicGroups` était-il activé ?
 - NON : L'objet ne peut pas être ajouté au groupe d'objets. La commande déclenche une exception.
 - OUI : L'objet est supprimé du groupe d'objets dynamiques le plus récent, une erreur est enregistrée et une [métrique AddThingToDynamicThingGroupsFailed](#) est générée 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 en raison 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 \(p. 307\)](#) et [Si l'attribut overrideDynamicGroups est activé, les groupes statiques sont prioritaires par rapport aux groupes dynamiques \(p. 308\)](#) 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 politiques 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 \(p. 313\)](#) 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 \(p. 311\)](#).

Pour faciliter l'utilisation, Tag Editor (Éditeur de balises) dans AWS Management Console permet de centraliser et d'unifier la création et la gestion de vos balises. Pour de plus amples informations, veuillez consulter [Utilisation de Tag Editor](#) dans [Utilisation de AWS Management Console](#).

Vous pouvez aussi gérer les balises à l'aide de la AWS CLI et l'AWS IoT API. 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 et des groupes de facturation lorsque vous les créez en utilisant le Tags dans les commandes suivantes :

- [CreateBillingGroup](#)
- [Créer une destination](#)
- [Créer un profil de périphérique](#)
- [CreateDynamicThingGroup](#)
- [CreateJob](#)
- [CreateOTAUpdate](#)
- [CreatePolicy](#)
- [CreateScheduledAudit](#)
- [CreateSecurityProfile](#)
- [Créer un profil de service](#)

- [CreateStream](#)
- [CreateThingGroup](#)
- [CreateThingType](#)
- [CreateTopicRule](#)
- [Créer une passerelle sans fil](#)
- [Créer un appareil sans fil](#)

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 les 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 AWS IoT les ressources identifiables et les clés de condition qu'elles prennent en charge, lire [Actions, ressources et clés de condition pour AWS IoT](#).

Pour de plus amples informations sur l'utilisation des balises, consultez [Contrôle de l'accès à l'aide de balises](#) dans le AWS Identity and Access Management Guide 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 pour le ThingGroup actions. Un utilisateur IAM restreint par cette stratégie :

- Ne peut pas créer un groupe d'objets la balise « env=prod » (dans l'exemple, voir la ligne)`"aws:RequestTag/env" : "prod"`).
- Ne peut pas modifier un groupe d'objets 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:CreateThingGroup",  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:RequestTag/env": "prod"  
                }  
            }  
        },  
        {  
            "Effect": "Deny",  
            "Action": [  
                "iot:CreateThingGroup",  
                "iot>DeleteThingGroup",  
                "iot:DescribeThingGroup",  
                "iot:UpdateThingGroup"  
            ],  
            "Resource": "*",  
            "Condition": {  
                "StringEquals": {  
                    "aws:ResourceTag/env": "prod"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:CreateThingGroup",  
            ]  
        }  
    ]  
}
```

```
        "iot:DeleteThingGroup",
        "iot:DescribeThingGroup",
        "iot:UpdateThingGroup"
    ],
    "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 périphériques sans fil et les passerelles, ne peuvent pas être ajoutés aux groupes de facturation. Cependant, ils peuvent être associés à AWS IoT Things, 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.

Le AWS IoT API sans fil fournit ces actions pour associer des périphériques et passerelles sans fil à AWS IoT Things.

- [Associer un appareil sans fil à Thing](#)

- [Associez la passerelle sans fil à Thing](#)

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 à des groupes de facturation (et donc aux éléments qu'ils incluent), AWS génère 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, consultez [Utiliser les balises de répartition des coûts](#) dans le [AWS Guide 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 \(p. 287\)](#).
- 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 des appareils” \(p. 89\)](#).
- 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 [Détection \(p. 1090\)](#) (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 \(p. 287\)](#).
- 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 \(p. 894\)](#).
- Rapports d'[Audit \(p. 1023\)](#).

Sécurité dans AWS IoT

Chez AWS, la sécurité dans le cloud est notre priorité numéro 1. 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 entre AWS et vous-même. 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 cloud AWS. AWS vous fournit également les 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 AWSservice 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.

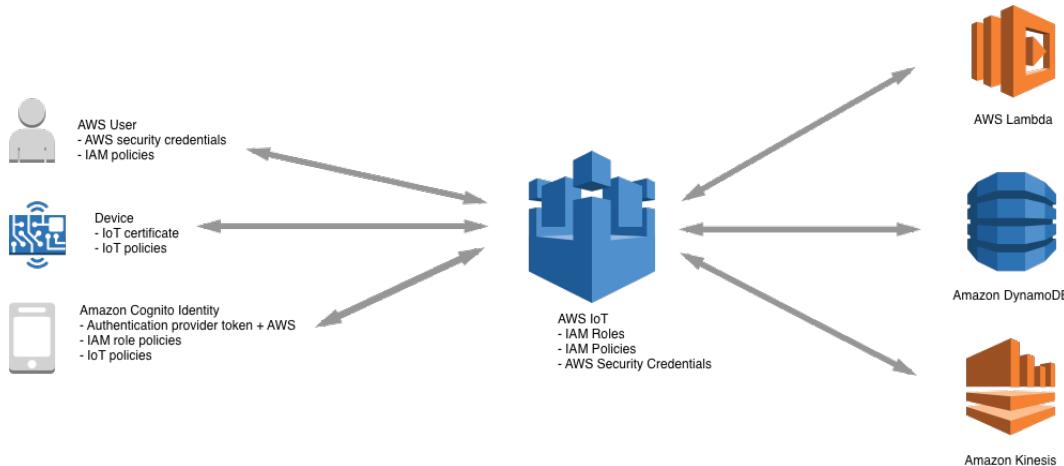
Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de AWS IoT. Les rubriques suivantes expliquent comment configurer AWS IoT pour répondre à vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres services AWS pour surveiller et sécuriser vos ressources AWS IoT.

Rubriques

- [Sécurité AWS IoT \(p. 315\)](#)
- [Authentification \(p. 316\)](#)
- [Autorisation \(p. 355\)](#)
- [Protection des données dans AWS IoT Core \(p. 408\)](#)
- [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#)
- [Journalisation et surveillance \(p. 453\)](#)
- [Validation de la conformité pour AWS IoT Core \(p. 454\)](#)
- [La résilience au cœur de AWS IoT \(p. 455\)](#)
- [Utilisation d'AWS IoT Core avec les points de terminaison d'un VPC d'interface \(p. 456\)](#)
- [Sécurité de l'infrastructure dans AWS IoT \(p. 458\)](#)
- [Surveillance de la sécurité des flottes de production ou des appareils avec Core AWS IoT \(p. 459\)](#)
- [Bonnes pratiques de sécurité dans AWS IoT Core \(p. 459\)](#)
- [Formation et certification AWS \(p. 465\)](#)

Sécurité AWS IoT

Chaque appareil ou client connecté doit disposer d'informations d'identification pour interagir avec AWS IoT. L'ensemble du trafic en provenance et à AWS IoT destination est envoyé en toute sécurité via le protocole TLS (Transport Layer Security). AWS les mécanismes de sécurité du cloud protègent les données lorsqu'elles transitent AWS IoT entre d'autres AWS services.



- Vous êtes responsable de la gestion des informations d'identification de l'appareil (certificats X.509, AWS informations d'identification, identités Amazon Cognito, identités fédérées ou jetons d'authentification personnalisés) et des politiques dans AWS IoT. Pour plus d'informations, veuillez consulter [Gestion des clés dans AWS IoT \(p. 413\)](#). 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 à AWS IoT l'aide de certificats X.509 ou d'identités Amazon Cognito via une connexion TLS sécurisée. Au cours de la recherche et du développement, et pour certaines applications qui effectuent des appels d'API ou utilisent des APIWebSockets, vous pouvez également vous authentifier à l'aide d'utilisateurs et de groupes IAM ou de jetons d'authentification personnalisés. Pour plus d'informations, veuillez consulter [Utilisateurs, groupes et rôles IAM \(p. 342\)](#).
- 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 autorisateur personnalisé est chargé d'authentifier vos appareils et d'accorder ou de refuser les autorisations d'accès que vous spécifiez pour vos appareils à l'aide des politiques IAMAWS IoT.
- Le moteur de AWS IoT règles transmet les données des appareils à d'autres appareils ou à d'autres AWS services conformément aux 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, veuillez consulter [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#).

Authentification

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-mêmes auprès d'AWS IoT.

Formation et certification AWS

Suivez le cours suivant pour en savoir plus sur l'authentification dans AWS IoT : [Deep Dive into AWS IoT Authentication and Authorization](#).

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 \(p. 320\)](#). 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 \(p. 324\)](#).

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 \(p. 409\)](#) 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 points de terminaison de données différents, `iot:Data` et `iot:Data-ATS`. `iot:Data` les points de terminaison présentent un certificat signé par le certificat d'autorité de [certification racine G5 public primaire de VeriSign classe 3](#). `iot:Data-ATS` les points de terminaison 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 `iot:Data-ATS` point de terminaison à l'aide de l'[DescribeEndpointAPI](#).
- 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 :

VeriSignPoints de terminaison (anciens)

- Clé RSA 2048 bits : certificat CA [VeriSignracine G5 public primaire de classe 3](#)

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 les Amazon Root CA certificats pris en charge.
- 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 les certificats [Amazon Root CA 3](#) et [Amazon Root CA 4](#) ECC. Si vos appareils ne mettent pas en œuvre la validation des certificats basée sur le RSA, vous pouvez omettre les certificats [Amazon Root CA 1](#) et [Amazon Root CA 2](#) 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.
 - [Signé croisé Amazon Root CA 1](#)
 - [Signé croisé Amazon Root CA 2](#) - Réservé pour une utilisation future.
 - [Signé croisé Amazon Root CA 3](#)
 - [Signé croisé Amazon Root CA 4](#) - Réservé pour une utilisation future.
- Si vous rencontrez des problèmes de validation de certificat de serveur, votre appareil devra peut-être explicitement approuver l'autorité de certification racine. Essayez de l'ajouter [Starfield Root CA Certificate](#) à votre Trust Store.
- Si vous rencontrez toujours des problèmes après avoir exécuté les étapes ci-dessus, veuillez contacter [l'Support AWS développeurs](#).

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 \(p. 1494\)](#).

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 \(p. 320\)](#)
- [Utilisateurs, groupes et rôles IAM \(p. 342\)](#)
- [Identités Amazon Cognito \(p. 342\)](#)

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. Généralement, AWS IoT les appareils utilisent des certificats X.509, tandis que les applications mobiles utilisent les identités Amazon Cognito. Les applications Web et de bureau utilisent des identités IAM ou fédérées. AWS CLI les commandes utilisent IAM. Pour de plus amples informations sur les identités IAM, veuillez consulter [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#).

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 en plusieurs fois Compte AWS Région AWS pour faciliter le déplacement d'appareils entre les vôtres Compte AWS situés dans la même région. Pour en savoir plus, consultez [Utilisation de certificats clients X.509 en plusieurs fois avec l'Compte AWS enregistrement de plusieurs comptes \(p. 321\)](#).

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 \(p. 894\)](#) 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 \(p. 322\)](#)
- [Création de vos propres certificats clients \(p. 323\)](#)
- [Enregistrement d'un certificat client \(p. 331\)](#)
- [Activation ou désactivation d'un certificat client \(p. 336\)](#)
- [Révocation d'un certificat client \(p. 338\)](#)

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 utilisé pour le chiffrement des données. Dans le cadre de l'authentification client TLS, AWS IoT demande un certificat client X.509 et valide l'état du certificat par Compte AWS rapport à 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 Core \(p. 409\)](#).

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 \(p. 323\)](#).

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 \(p. 322\)](#).

Utilisation de certificats clients X.509 en plusieurs fois avec l'Compte AWS enregistrement de plusieurs comptes

L'enregistrement de plusieurs comptes permet de déplacer des appareils entre vos Compte AWS appareils dans la même région ou dans des régions différentes. Vous pouvez enregistrer, tester et configurer un appareil dans un compte de pré-production, puis enregistrer et utiliser le même appareil et le même certificat d'appareil dans un compte de production. Vous pouvez également enregistrer le certificat client sur l'appareil ou les certificats de l'appareil sans une autorité de certification enregistrée auprès de AWS IoT. Pour plus d'informations, voir [Enregistrer un certificat client signé par une autorité de certification \(CLI non enregistrée \(p. 333\)\)](#).

Note

Les certificats utilisés pour l'enregistrement de plusieurs comptes sont pris en charge sur les types de `iot:CredentialProvider` terminaux `iot:Data-ATSIot:Jobs`, `iot:Data` (anciens) et. Pour de plus amples informations sur les points de terminaison de l'AWS IoT appareil, veuillez consulter [AWS IoT données de l'appareil et points de terminaison de service \(p. 85\)](#).

Les appareils qui utilisent l'enregistrement multi-comptes doivent envoyer l'[extension SNI \(Server Name Indication\)](#) au protocole TLS (Transport Layer Security) et fournir l'adresse complète du point de terminaison `host_name` sur le terrain, lorsqu'ils se connectent à AWS IoT. AWS IoT utilise l'adresse du point de terminaison `host_name` pour acheminer la connexion vers le AWS IoT compte approprié. 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 Core \(p. 409\)](#).

Pour utiliser l'enregistrement de plusieurs comptes

1. Vous pouvez enregistrer les certificats de l'appareil auprès d'une autorité de certification. Vous pouvez enregistrer l'autorité de certification signataire dans plusieurs comptes en SNI_ONLY mode et utiliser cette autorité de certification pour enregistrer le même certificat client sur plusieurs comptes. Pour plus d'informations, veuillez consulter [Enregistrer un certificat CA en mode SNI_ONLY \(CLI\) \(p. 327\)](#).
2. Vous pouvez enregistrer les certificats de l'appareil sans autorité de certification. Consultez [Enregistrer un certificat client signé par une autorité de certification \(CLI\) non enregistrée \(p. 333\)](#). L'enregistrement d'une autorité de certification est facultatif. Vous n'êtes pas obligé d'enregistrer l'autorité de certification avec laquelle vous avez signé les certificats de l'appareil AWS IoT.

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

Pour plus d'informations sur l'authentification et la sécurité de certificats, consultez [Qualité clé du certificat de l'appareil \(p. 1028\)](#).

Note

La demande de signature de certificat (CSR) doit inclure une clé publique qui est soit une clé RSA d'une longueur d'au moins 2048 bits, soit une clé ECC issue des courbes NIST P-256 ou NIST P-384. Pour plus d'informations, consultez l'[CreateCertificateFromCsr](#) API dans le Guide de référence des AWS IoT API.

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 par ce dernier AWS IoT contient des attributs d'émetteur et d'objet qui sont définis au moment de la création du certificat. Les attributs du certificat ne sont immuables qu'une fois le certificat créé.

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 console AWS de gestion et ouvrez la [AWS IoT console](#).
2. Dans le panneau de navigation de gauche, choisissez successivement Sécurité, Certificats et Créer.

3. Choisissez Création d'un certificat en un clic (recommandé) - Créez 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. Ces certificats générés par AWS IoT ne peuvent être utilisés qu'avec les AWS IoT services.

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.

Choisissez Activer pour activer le certificat client dès maintenant. Si vous ne souhaitez pas activer le certificat maintenant, [Activation d'un certificat client \(console\) \(p. 336\)](#) décrit comment l'activer ultérieurement.

6. Si vous souhaitez associer une politique au certificat, choisissez Joindre une politique.

Si vous ne souhaitez pas joindre de politique maintenant, choisissez OK pour 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 \(p. 318\)](#).

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-pemoutfile certificate_filename.pem \
--public-keyoutfile public_filename.key \
--private-keyoutfile private_filename.key
```

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 \(CLI\) \(p. 336\)](#) décrit comment activer le certificat ultérieurement.

```
aws iot create-keys-and-certificate \
--no-set-as-active \
--certificate-pemoutfile certificate_filename.pem \
--public-keyoutfile public_filename.key \
--private-keyoutfile private_filename.key
```

Installez les fichiers de certificat sur le client.

Création de vos propres certificats clients

AWS IoT prend en charge les certificats clients signés par n'importe quelle autorité de certification (CA) racine ou intermédiaire. AWS IoT utilise des certificats CA pour vérifier la propriété des certificats.

Pour utiliser des certificats d'appareil signés par une autorité de certification qui n'est pas l'autorité de certification d'Amazon, le certificat de l'autorité de certification doit être enregistré AWS IoT afin que nous puissions vérifier la propriété du certificat de l'appareil.

AWS IoT propose plusieurs méthodes pour apporter vos propres certificats (BYOC) :

- Tout d'abord, enregistrez l'autorité de certification utilisée pour signer les certificats clients, puis enregistrez les certificats clients individuels. Si vous souhaitez enregistrer l'appareil ou le client sur son certificat client lors de sa première connexion AWS IoT (également connu sous le nom de [provisionnement juste à temps](#)), vous devez enregistrer l'autorité de certification signataire auprès AWS IoT de l'autorité de certification et activer l'enregistrement automatique.
- Si vous ne parvenez pas à enregistrer l'autorité de certification signataire, vous pouvez choisir d'enregistrer les certificats clients sans autorité de certification. Pour les appareils enregistrés sans autorité de certification, vous devez présenter l'[indication du nom du serveur \(SNI\)](#) lorsque vous les connectez à AWS IoT.

Note

Pour enregistrer des certificats clients à l'aide d'une autorité de certification, vous devez enregistrer l'autorité de certification signataire auprès AWS IoT d'aucune autre autorité de certification de la hiérarchie.

Note

Un certificat CA ne peut être enregistré en DEFAULT mode que par un seul compte par région. Un certificat CA peut être enregistré en SNI_ONLY mode par plusieurs comptes 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 \(p. 894\)](#) 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 \(p. 324\)](#)
- [Création d'un certificat client à l'aide de votre certificat d'autorité de certification \(p. 330\)](#)

Gestion de vos certificats d'autorité de certification

Cette section décrit les tâches courantes de gestion de vos propres certificats 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 CA ne peut être enregistré en DEFAULT mode que par un seul compte par région. Un certificat CA peut être enregistré en SNI_ONLY mode par plusieurs comptes dans une région.

- [Création d'un certificat d'autorité de certification \(p. 325\)](#), si besoin.

Créez les fichiers de certificat et de clé dont vous avez besoin pour l'étape suivante.

- [Enregistrement de votre certificat d'autorité de certification \(p. 325\)](#)

Enregistrez votre certificat CA auprès de AWS IoT

- [Désactivation d'un certificat d'autorité de certification \(p. 329\)](#)

Création d'un certificat d'autorité de certification

Si vous ne possédez pas de certificat CA, vous pouvez utiliser les outils [OpenSSL v1.1.1i](#) pour en créer un.

Note

Vous ne pouvez pas effectuer cette procédure dans la console AWS IoT.

Pour créer un certificat CA à l'aide des outils [OpenSSL](#) v1.1.1i

1. Générez une paire de clés.

```
openssl genrsa -out root_CA_key_filename.key 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.key \
    -sha256 -days 1024 \
    -out root_CA_cert_filename.pem
```

Enregistrement de votre certificat d'autorité de certification

Ces procédures décrivent comment enregistrer un certificat auprès d'une autorité de certification autre que l'autorité de certification d'Amazon. AWS IoT Core utilise des certificats CA pour vérifier la propriété des certificats. Pour utiliser des certificats d'appareil signés par une autorité de certification qui n'est pas l'autorité de certification d'Amazon, vous devez enregistrer le certificat auprès de l'autorité de certification AWS IoT Core afin qu'elle puisse vérifier la propriété du certificat de l'appareil.

Enregistrement d'un certificat d'autorité de certification (console)

Note

Pour enregistrer un certificat CA dans la console, commencez dans la console en cliquant sur [Enregistrer le certificat CA](#). Vous pouvez enregistrer votre autorité de certification en mode multi-comptes, sans avoir à fournir de certificat de vérification ni à accéder à la clé privée. Une autorité de certification peut être enregistrée en mode multi-comptes par plusieurs utilisateurs Comptes AWS à la foisRégion AWS. Vous pouvez enregistrer votre autorité de certification en mode compte unique en fournissant un certificat de vérification et une preuve de propriété de la clé privée de l'autorité de certification. Une autorité de certification peut être enregistrée en mode multi-comptes par une seule Compte AWS personneRégion AWS.

Enregistrement d'un certificat d'autorité de certification (CLI)

Vous pouvez enregistrer un certificat CA en DEFAULT mode ou en SNI_ONLY mode. Une autorité de certification peut être enregistrée en DEFAULT mode une Compte AWS par uneRégion AWS. Une autorité de certification peut être enregistrée en SNI_ONLY mode par plusieurs Comptes AWS dans le même modeRégion AWS. Pour plus d'informations sur le mode certificat d'autorité de certification, consultez [CertificateMode](#).

Enregistrer un certificat CA en DEFAULT mode (CLI)

Prérequis

Vérifiez que les informations suivantes sont disponibles sur votre ordinateur avant de continuer :

- Le fichier de certificat de l'autorité de certification racine (référencé dans l'exemple suivant comme `root_CA_cert_filename.pem`)
- Le fichier de clé privée du certificat CA racine (référencé dans l'exemple suivant comme `root_CA_key_filename.key`)
- [OpenSSL v1.1.1i ou version ultérieure](#)

Pour enregistrer un certificat CA en **DEFAULT** mode à l'aide du AWS CLI

1. Pour obtenir un code d'enregistrement auprès de AWS IoT, utilisez `get-registration-code`. Enregistrez le résultat `registrationCode` à utiliser en tant que certificat Common Name de vérification de la clé privée. Pour plus d'informations, consultez la section [get-registration-code](#) dans la référence des commandes AWS CLI.

```
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.key 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.key \
    -out verification_cert_csr_filename.csr
```

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.csr \
    -CA root_CA_cert_filename.pem \
    -CAkey root_CA_key_filename.key \
    -CAcreateserial \
    -out verification_cert_filename.pem \
    -days 500 -sha256
```

5. Enregistrez le certificat CA auprès d'AWS IoT. Entrez le nom du fichier du certificat CA et le nom du fichier du certificat de vérification par clé privée à la register-ca-certificate commande, comme suit. Pour plus d'informations, consultez la section [register-ca-certificate](#) dans la référence des commandes AWS CLI.

```
aws iot register-ca-certificate \
--ca-certificate file://root_CA_cert_filename.pem \
--verification-cert file://verification_cert_filename.pem
```

Cette commande, si elle aboutit, renvoie l'*ID de certificat*.

6. À ce stade, le certificat CA a été enregistré auprès de AWS IoT mais n'est pas actif. Le certificat CA doit être actif pour que vous puissiez enregistrer les certificats clients qu'il a signés.

Cette étape active le certificat de l'autorité de certification.

Pour activer le certificat CA, utilisez la update-certificate commande suivante. Pour plus d'informations, voir [update-certificate](#) dans la AWS CLI référence des commandes.

```
aws iot update-ca-certificate \
--certificate-id certificateId \
--new-status ACTIVE
```

Pour voir l'état du certificat CA, utilisez la describe-ca-certificate commande. Pour plus d'informations, consultez la section [describe-ca-certificate](#) dans la référence des commandes AWS CLI.

Enregistrer un certificat CA en mode SNI_ONLY (CLI)

Prérequis

Vérifiez que les informations suivantes sont disponibles sur votre ordinateur avant de continuer :

- Le fichier de certificat de l'autorité de certification racine (référencé dans l'exemple suivant comme *root_CA_cert_filename.pem*)
- [OpenSSL v1.1.1i ou version ultérieure](#)

Pour enregistrer un certificat CA en **SNI_ONLY** mode à l'aide du AWS CLI

1. Enregistrez le certificat CA auprès d'AWS IoT. À l'aide de la register-ca-certificate commande, entrez le nom du fichier du certificat CA. Pour plus d'informations, consultez la section [register-ca-certificate](#) dans la référence des commandes AWS CLI.

```
aws iot register-ca-certificate \
--ca-certificate file://root_CA_cert_filename.pem \
--certificate-mode SNI_ONLY
```

Si elle aboutit, cette commande renvoie l'*certificateId*.

2. À ce stade, le certificat CA a été enregistré auprès de AWS IoT mais est inactif. Le certificat CA doit être actif pour que vous puissiez enregistrer les certificats clients qu'il a signés.

Cette étape active le certificat de l'autorité de certification.

Pour activer le certificat CA, utilisez la update-certificate commande suivante. Pour plus d'informations, voir [update-certificate](#) dans la AWS CLI référence des commandes.

```
aws iot update-ca-certificate \
```

```
--certificate-id certificateId \  
--new-status ACTIVE
```

Pour voir l'état du certificat CA, utilisez la describe-ca-certificate commande. Pour plus d'informations, consultez la section [describe-ca-certificate](#) dans la référence des commandes AWS CLI.

Créez un certificat de vérification CA pour enregistrer le certificat CA dans la console

Note

Cette procédure ne doit être utilisée que si vous enregistrez un certificat CA à partir de la AWS IoT console.

Si vous n'avez pas accédé à cette procédure depuis la AWS IoT console, lancez le processus d'enregistrement du certificat CA dans la console à l'adresse [Enregistrer le certificat CA](#).

Vérifiez que les éléments suivants sont disponibles sur le même ordinateur avant de continuer :

- Le fichier de certificat de l'autorité de certification racine (référencé dans l'exemple suivant comme *root_CA_cert_filename.pem*)
- Le fichier de clé privée du certificat CA racine (référencé dans l'exemple suivant comme *root_CA_key_filename.key*)
- [OpenSSL v1.1.1i ou version ultérieure](#)

Pour utiliser l'interface de ligne de commande pour créer un certificat de vérification CA afin d'enregistrer votre certificat CA dans la console

1. *verification_cert_key_filename.key* Remplacez-le par le nom du fichier clé du certificat de vérification que vous souhaitez créer (par exemple, **verification_cert.key**). Exécutez ensuite cette commande pour générer une key pair pour le certificat de vérification de clé privée :

```
openssl genrsa -out verification_cert_key_filename.key 2048
```

2. Remplacez *verification_cert_key_filename.key* par le nom du fichier clé que vous avez créé à l'étape 1.

Remplacez *verification_cert_csr_filename.csr* par le nom du fichier de demande de signature de certificat (CSR) que vous souhaitez créer. Par exemple, **verification_cert.csr**.

Exécutez cette commande pour créer le fichier CSR.

```
openssl req -new \  
-key verification_cert_key_filename.key \  
-out verification_cert_csr_filename.csr
```

La commande vous invite à fournir des informations supplémentaires qui seront expliquées ultérieurement.

3. Dans la AWS IoT console, dans le conteneur du certificat de vérification, copiez le code d'enregistrement.
4. Les informations que la openssl commande vous demande de saisir sont présentées dans l'exemple suivant. À l'exception du Common Name champ, vous pouvez saisir vos propres valeurs ou les laisser vides.

Dans le Common Name champ, collez le code d'enregistrement que vous avez copié à l'étape précédente.

```
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 []:
```

Une fois que vous avez terminé, la commande crée le fichier CSR.

5. *verification_cert_csr_filename.csr* Remplacez-le par celui *verification_cert_csr_filename.csr* que vous avez utilisé à l'étape précédente.
- root_CA_cert_filename.pem* Remplacez-le par le nom de fichier du certificat CA que vous souhaitez enregistrer.
- root_CA_key_filename.key* Remplacez-le par le nom du fichier de clé privée du certificat CA.

Remplacez *verification_cert_filename.pem* par le nom de fichier du certificat de vérification que vous souhaitez créer. Par exemple, **verification_cert.pem**.

```
openssl x509 -req \  
-in verification_cert_csr_filename.csr \  
-CA root_CA_cert_filename.pem \  
-CAkey root_CA_key_filename.key \  
-CAcreateserial \  
-out verification_cert_filename.pem \  
-days 500 -sha256
```

6. Une fois la commande OpenSSL terminée, vous devriez disposer de ces fichiers prêts à être utilisés lorsque vous retournez sur la console.
 - Votre fichier de certificat CA (*root_CA_cert_filename.pem* utilisé dans la commande précédente)
 - Le certificat de vérification que vous avez créé à l'étape précédente (*verification_cert_filename.pem* utilisé dans la commande précédente)

Désactivation d'un certificat d'autorité de certification

Lorsqu'un certificat d'autorité de certification (CA) est activé pour l'enregistrement automatique des certificats clients, AWS IoT vérifie le certificat de l'autorité de certification pour s'assurer que l'autorité de certification l'estACTIVE. 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 surINACTIVE, 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 AWS Management Console et ouvrez la [console AWS IoT](#).
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 choisissez l'icône représentant des points de suspension pour ouvrir le menu d'options.
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 \(CLI\) \(p. 330\)](#).

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 \(p. 331\)](#).

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.key 2048
```

2. Créez une demande de signature de certificat (CSR) pour le certificat client.

```
openssl req -new \  
  -key device_cert_key_filename.key \  
  -out device_cert_filename.csr
```

```
-out device_cert_csr_filename.csr
```

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 []:
```

- Créez un certificat client à partir de la CSR.

```
openssl x509 -req \
-in device_cert_csr_filename.csr \
-CA root_CA_cert_filename.pem \
-CAkey root_CA_key_filename.key \
-CAcreateserial \
-out device_cert_filename.pem \
-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 \(p. 331\)](#).

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 \(p. 325\)](#) 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 Comptes AWS et par 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, veuillez consulter [Sécurité du transport dans AWS IoT Core \(p. 409\)](#).

Rubriques

- [Enregistrement manuel d'un certificat client \(p. 332\)](#)
- [Enregistrer un certificat client lorsque le client se connecte à l'AWS IoT just-in-time enregistrement \(JITR\) \(p. 334\)](#)

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 varie selon que le certificat sera partagé Compte AWS ou non par les 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 des entités et des régions, mais uniquement si le certificat client est signé par une autorité de certification (CA) qui n'est PAS enregistrée auprès de AWS IoT.

Enregistrer 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 \(p. 325\)](#).

Pour enregistrer un certificat existant auprès d'AWS IoT à l'aide de la console

1. Connectez-vous à la console AWS de gestion et ouvrez la [AWS IoT console](#).
2. Dans le volet de navigation, dans la section Gérer, choisissez Sécurité, puis Certificats.
3. Sur la page Certificats de la boîte de dialogue Certificats, choisissez Ajouter un certificat, puis Enregistrer des certificats.
4. Sur la page Enregistrer le certificat de la boîte de dialogue Certificats à télécharger, procédez comme suit :
 - Choose CA est enregistré auprès de AWS IoT.
 - Dans Choisir un certificat CA, sélectionnez votre autorité de certification.
 - Choisissez Enregistrer une nouvelle autorité de certification pour enregistrer une nouvelle autorité de certification qui n'est pas enregistrée auprès de AWS IoT.
 - Laissez le champ Choisir un certificat CA vide si l'autorité de certification Amazon Root est votre autorité de certification.
 - Sélectionnez jusqu'à 10 certificats à télécharger et à enregistrer AWS IoT.
 - Utilisez les fichiers de certificat que vous avez créés dans [Création de certificats clients AWS IoT \(p. 322\)](#) et [Création d'un certificat client à l'aide de votre certificat d'autorité de certification \(p. 330\)](#).
 - Choisissez Activer ou Désactiver. Si vous choisissez Désactiver, [Activation ou désactivation d'un certificat client \(p. 336\)](#) explique comment activer votre certificat après son enregistrement.
 - Choisissez Register.

Sur la page Certificats de la boîte de dialogue Certificats, vos certificats enregistrés apparaîtront désormais.

Enregistrer 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 console AWS de gestion et ouvrez la [AWS IoTconsole](#).
2. Dans le volet de navigation de gauche, choisissez successivement Sécurité, Certificats et Créer.
3. Dans Créer un certificat, recherchez l'entrée Utiliser mon certificat, puis choisissez Commencer.
4. Dans Sélectionnez une autorité de certification, choisissez Suivant.
5. Dans Enregistrer les certificats d'appareils existants, choisissez Sélectionner les certificats, puis 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 \(console\) \(p. 336\)](#) décrit comment l'activer ultérieurement.

Si un certificat est révoqué lors de son enregistrement, il ne peut pas être activé ultérieurement.

Après avoir choisi les fichiers de certificats à enregistrer et sélectionné les actions à effectuer après l'enregistrement, sélectionnez Enregistrer les certificats.

Les certificats clients enregistrés apparaissent dans la liste des certificats.

Enregistrer 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 \(p. 325\)](#).

Utilisez la commande [register-certificate](#) pour enregistrer un certificat client sans l'activer.

```
aws iot register-certificate \
--certificate-pem file://device_cert_filename.pem \
--ca-certificate-pem file://ca_cert_filename.pem
```

Le certificat client est enregistré auprès d'AWS IoT, mais il n'est pas encore actif. Veuillez consulter [Activation d'un certificat client \(CLI\) \(p. 336\)](#) 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_filename.pem \
--ca-certificate-pem file://ca_cert_filename.pem
```

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 \(p. 336\)](#)

Enregistrer un certificat client signé par une autorité de certification (CLI) non enregistrée

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_filename.pem
```

Le certificat client est enregistré auprès d'AWS IoT, mais il n'est pas encore actif. Veuillez consulter [Activation d'un certificat client \(CLI\) \(p. 336\)](#) 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_filename.pem
```

Pour plus d'informations sur l'activation du certificat afin qu'il puisse être utilisé pour se connecter AWS IoT, consultez [Activation ou désactivation d'un certificat client \(p. 336\)](#).

Enregistrer un certificat client lorsque le client se connecte à l'AWS IoT just-in-time enregistrement (JITR)

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 console AWS de gestion et ouvrez la [AWS IoT console](#).
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 \  
--auto-registration-status ENABLE
```

```
--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_cert_filename.pem \  
--verification-cert file://verification_cert_filename.pem
```

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 AWS IoT pour la première fois, il doit présenter un fichier contenant votre certificat client signé par votre certificat CA dans le cadre de l'établissement de liaison TLS.

Lorsque le client se connecte à AWS IoT, utilisez le *client_certificate_filename* fichier comme fichier de certificat. AWS IoT reconnaît le certificat CA en tant que certificat CA 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.

Note

Vous pouvez approvisionner des appareils à l'aide de la fonction AWS IoT Core just-in-time d'enregistrement (JITR) sans avoir à envoyer l'intégralité de la chaîne de confiance lors de la première connexion des appareils à AWS IoT Core. La présentation du certificat CA est facultative mais l'appareil doit envoyer l'extension [SNI \(Server Name Indication\)](#) lors de la connexion.

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 certificats (CRL), active le certificat, crée et associe une politique au certificat. La stratégie détermine les ressources auxquelles le client peut accéder. Pour plus d'informations sur la façon de créer une règle Lambda qui écoute le \$aws/events/certificates/registered/*caCertificateID* sujet et exécute ces actions, consultez la section [just-in-timeEnregistrement des certificats clients](#) sur AWS IoT.

Si une erreur ou une exception se produit lors de l'enregistrement automatique des certificats clients, AWS IoT envoie des événements ou des messages à vos CloudWatch journaux dans Logs. Pour

de plus amples informations sur la configuration des journaux de votre compte, veuillez consulter la [CloudWatchdocumentation Amazon](#).

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 console AWS de gestion et ouvrez la [AWS 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 console AWS de gestion et ouvrez la [AWS 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 appareil AWS IoT afin qu'il puisse se connecter. L'attachement du certificat à une ressource objet établit la relation entre le périphérique (au moyen du certificat) et la ressource objet. Pour autoriser l'appareil à effectuer AWS IoT des actions, telles que l'autoriser à se connecter et à publier des messages, une politique appropriée doit être jointe 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 console AWS de gestion et ouvrez la [AWS 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'élément que vous souhaitez joindre au certificat, cochez la case correspondante, puis choisissez Joindre.

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 console AWS de gestion et ouvrez la [AWS 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 politique que vous souhaitez associer au certificat, cochez la case correspondante, puis choisissez Joindre.

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.

Note

Une fois qu'un certificat est révoqué, son statut ne peut pas être modifié. En d'autres termes, le statut du certificat ne peut pas être modifié Active ni aucun autre statut.

Révocation d'un certificat client (console)

Pour révoquer un certificat client à l'aide de la console AWS IoT

1. Connectez-vous à la console AWS de gestion et ouvrez la [AWS 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

Les certificats X.509 qui appartiennent à l'un Compte AWS peuvent être transférés à un autreCompte AWS.

Pour transférer un certificat X.509 de l'un Compte AWS à l'autre

1. [the section called “Commencer un transfert de certificat” \(p. 339\)](#)

Le certificat doit être désactivé et détaché de toutes les politiques et de tous les éléments avant de lancer le transfert.

2. [the section called “Accepter ou rejeter un transfert de certificat” \(p. 340\)](#)

Le compte destinataire doit accepter ou rejeter explicitement le certificat transféré. Une fois que le compte récepteur a accepté le certificat, celui-ci doit être activé avant d'être utilisé.

3. [the section called “Annulation d'un transfert de certificat” \(p. 341\)](#)

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 en Compte AWS utilisant la [AWS IoTconsole](#) ou leAWS CLI.

Commencer un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'ID du certificat que vous souhaitez transférer.

Effectuez cette procédure à partir du compte contenant le certificat à transférer.

Pour commencer à transférer un certificat vers un autre Compte AWS

1. Connectez-vous à la console AWS de gestion et ouvrez la [AWS IoTconsole](#).
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

Choisissez le certificat dont le statut est Actif ou Inactif que vous souhaitez transférer et ouvrez sa page de détails.

3. Sur la page Détails du certificat, dans le menu Actions, si l'option Désactiver est disponible, choisissez l'option Désactiver pour désactiver le certificat.
4. Sur la page Détails du certificat, dans le menu de gauche, choisissez Politiques.
5. Sur la page Politiques du certificat, si des politiques sont associées au certificat, détachez chacune d'elles en ouvrant le menu des options de la politique et en choisissant Détacher.

Aucune politique ne doit être associée au certificat avant que vous ne continuiez.

6. Sur la page Politiques du certificat, dans le menu de gauche, choisissez Objets.
7. Sur la page Objets du certificat, si des éléments sont attachés au certificat, détachez chacun d'entre eux en ouvrant le menu des options de l'objet et en choisissant Détacher.

Aucun élément ne doit être joint au certificat avant que vous ne continuiez.

8. Sur la page Objets du certificat, dans le menu de gauche, choisissez Détails.
9. Sur la page Détails du certificat, dans le menu Actions, choisissez Démarrer le transfert pour ouvrir la boîte de dialogue Démarrer le transfert.

10. Dans la boîte de dialogue Démarrer le transfert, entrez le Compte AWS numéro du compte auquel vous souhaitez recevoir le certificat et un court message facultatif.
11. Choisissez Démarrer le transfert pour transférer le certificat.

La console doit afficher un message indiquant la réussite ou l'échec du transfert. Si le transfert a été lancé, le statut du certificat passe à Transféré.

Commencer un transfert de certificat (CLI)

Pour effectuer cette procédure, vous aurez besoin du *certificateId* et du *CertificateArn* du certificat que vous souhaitez transférer.

Effectuez cette procédure à partir du compte contenant le certificat à transférer.

Pour commencer à transférer un certificat vers un autre AWS compte

1. Utilisez la [update-certificate](#) commande pour désactiver le certificat.

```
aws iot update-certificate --certificate-id certificateId --new-status INACTIVE
```

2. Détachez toutes les politiques.

1. Utilisez la [list-attached-policies](#) commande pour répertorier les politiques associées au certificat.

```
aws iot list-attached-policies --target certificateArn
```

2. Pour chaque politique attachée, utilisez la [detach-policy](#) commande pour détacher la stratégie.

```
aws iot detach-policy --target certificateArn --policy-name policy-name
```

3. Détachez tous les objets.

1. Utilisez la [list-principal-things](#) commande pour répertorier les éléments attachés au certificat.

```
aws iot list-principal-things --principal certificateArn
```

2. Pour chaque objet attaché, utilisez la [detach-thing-principal](#) commande pour le détacher.

```
aws iot detach-thing-principal --principal certificateArn --thing-name thing-name
```

4. Utilisez la [transfer-certificate](#) commande pour démarrer le transfert du 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 est transféré Compte AWS par une autre personne Compte AWS à l'aide de la [AWS IoT console](#) ou du AWS CLI.

Accepter ou rejeter un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'identifiant du certificat qui a été transféré sur votre compte.

Effectuez cette procédure à partir du compte qui a reçu le certificat transféré.

Pour accepter ou refuser un certificat transféré à votre Compte AWS

1. Connectez-vous à la console AWS de gestion et ouvrez la [AWS IoTconsole](#).
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

Choisissez le certificat dont le statut est En attente de transfert que vous souhaitez accepter ou rejeter et ouvrez sa page de détails.

3. Sur la page Détails du certificat, dans le menu Actions,
 - Pour accepter le certificat, choisissez Accepter le transfert.
 - Pour ne pas accepter le certificat, choisissez Refuser le transfert.

Accepter ou rejeter un transfert de certificat (CLI)

Pour terminer cette procédure, vous aurez besoin du *certificateId* de certificat du transfert de certificat que vous souhaitez accepter ou rejeter.

Effectuez cette procédure à partir du compte qui a reçu le certificat transféré.

Pour accepter ou refuser un certificat transféré à votre Compte AWS

1. Utilisez la [accept-certificate-transfer](#) commande pour accepter le certificat.

```
aws iot accept-certificate-transfer --certificate-id certificateId
```

2. Utilisez la [reject-certificate-transfer](#) commande pour 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 ne soit accepté en utilisant la [AWS IoTconsole](#) ou le AWS CLI.

Annuler un transfert de certificat (console)

Pour effectuer cette procédure, vous aurez besoin de l'identifiant du transfert de certificat que vous souhaitez annuler.

Effectuez cette procédure à partir du compte qui a initié le transfert du certificat.

Pour annuler un transfert de certificat

1. Connectez-vous à la console AWS de gestion et ouvrez la [AWS IoTconsole](#).
2. Dans le volet de navigation de gauche, choisissez Sécurité, puis Certificats.

Choisissez le certificat avec le statut Transféré dont vous souhaitez annuler le transfert et ouvrez son menu d'options.

3. Dans le menu des options du certificat, choisissez l'option Révoquer le transfert pour annuler le transfert du certificat.

Important

Veillez à ne pas confondre l'option Révoquer le transfert avec l'option Révoquer.

L'option Révoquer le transfert annule le transfert du certificat, tandis que l'option Révoquer rend le certificat irréversiblement inutilisable par AWS IoT

Annuler un transfert de certificat (CLI)

Pour terminer cette procédure, vous aurez besoin du *certificateId* de certificat du transfert de certificat que vous souhaitez annuler.

Effectuez cette procédure à partir du compte qui a initié le transfert du certificat.

Utilisez la [cancel-certificate-transfer](#) commande pour annuler le transfert du certificat.

```
aws iot cancel-certificate-transfer --certificate-id certificateId
```

Utilisateurs, groupes et rôles IAM

Les utilisateurs, les groupes et les rôles IAM constituent les mécanismes standard de gestion de l'identité et de l'authentification dans AWS. Vous pouvez les utiliser pour vous connecter à des interfaces AWS IoT HTTP à l'aide du AWS SDK et AWS CLI.

Les rôles IAM permettent également d'accéder AWS IoT à d'autres AWS ressources de votre compte en votre nom. Par exemple, si vous souhaitez qu'un appareil publie son état dans une table DynamoDB, les rôles IAM permettent AWS IoT d'interagir avec Amazon DynamoDB. Pour en savoir plus, consultez [Rôles IAM](#).

Pour les connexions à un courtier de messages via HTTP, AWS IoT authentifie les utilisateurs, les groupes et les rôles à l'aide du processus de signature de la version 4 de Signature. Pour plus d'informations, consultez [la section Signature des demandes d'AWS API](#).

Lors de l'utilisation de la version 4 de AWS Signature avec AWS IoT, les 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 \(p. 414\)](#).

identités Amazon Cognito

Amazon Cognito Identity vous permet de créer des informations d'AWS identification temporaires à privilégiés limités à utiliser dans des applications mobiles et Web. Lorsque vous utilisez Amazon Cognito, vous créez des groupes d'identités qui créent des identités uniques pour vos utilisateurs et vous authentifiez avec des fournisseurs d'identité tels que Login with Amazon, Facebook et Google. Vous pouvez également utiliser les identités Amazon Cognito avec vos propres identités authentifiées de développeur. Pour de plus amples informations, veuillez consulter [Amazon Cognito Identity](#).

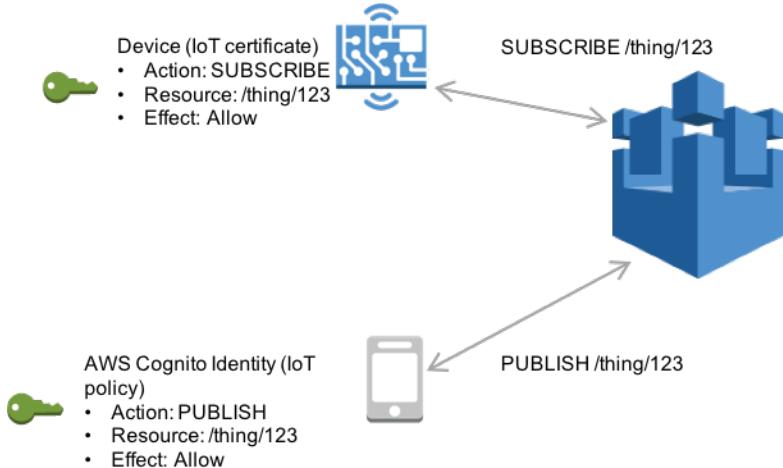
Pour utiliser Amazon Cognito Identity, vous devez définir un pool d'identités Amazon Cognito associé à un rôle IAM. Le rôle IAM est associé à une politique IAM qui autorise les identités issues de votre pool d'identités à accéder à des AWS ressources telles que les services d'appel AWS.

Amazon Cognito Identity crée des identités authentifiées et non 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 se voient accorder uniquement les autorisations spécifiées dans la politique IAM associée au pool d'identités.

Lorsque vous utilisez des identités authentifiées, en plus de la politique IAM associée au pool d'identités, vous devez associer une AWS IoT politique à une identité Amazon Cognito à l'aide de l'[AttachPolicy](#) API

et accorder des autorisations à un utilisateur individuel de votre application. AWS IoT Vous pouvez utiliser cette AWS IoT politique pour attribuer des autorisations précises à des clients spécifiques et à leurs appareils.

Les utilisateurs authentifiés et non authentifiés sont des types d'identité différents. Si vous n'associez pas de AWS IoT politique à Amazon Cognito Identity, un utilisateur authentifié ne reçoit pas d'autorisation AWS IoT et n'a pas accès aux AWS IoT ressources et aux actions. Pour plus d'informations sur la création de politiques pour les identités Amazon Cognito, consultez [Exemples de stratégie de publication/abonnement \(p. 376\)](#) et [Autorisation avec les identités Amazon Cognito \(p. 399\)](#).



Authentification et autorisation personnalisées

AWS IoT Core vous permet de définir des mécanismes d'autorisation personnalisée afin que vous puissiez gérer votre propre authentification et autorisation de client. Cela est utile lorsque vous devez utiliser des mécanismes d'authentification autres que ceux pris en charge de AWS IoT Core manière native. (Pour de plus amples informations sur les mécanismes pris en charge en mode natif, veuillez consulter[the section called "Authentification client" \(p. 320\)](#)).

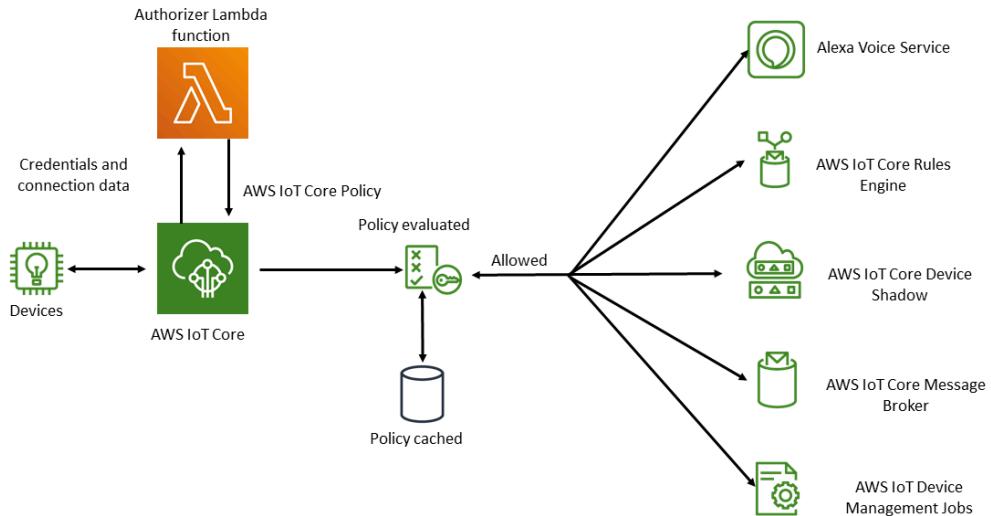
Par exemple, si vous migrez des appareils existants sur le terrain vers AWS IoT Core lesquels ces appareils utilisent un jeton porteur personnalisé ou un nom d'utilisateur et un mot de passe MQTT pour s'authentifier, vous pouvez les migrer vers AWS IoT Core sans avoir à leur attribuer de nouvelles identités. Vous pouvez utiliser l'authentification personnalisée avec tous les protocoles de communication pris en AWS IoT Core charge. Pour de plus amples informations sur les protocoles pris en AWS IoT Core charge, veuillez consulter[the section called "Protocoles de communication des appareils" \(p. 89\)](#).

Rubriques

- [Comprendre le flux de travail d'authentification personnalisé \(p. 343\)](#)
- [Création et gestion d'autoriseurs personnalisés \(p. 345\)](#)
- [Connexion à l'AWS IoT Core aide d'une authentification personnalisée \(p. 351\)](#)
- [Résolution des problèmes de vos autorisations \(p. 353\)](#)

Comprendre le flux de travail d'authentification personnalisé

L'authentification personnalisée vous permet de définir comment authentifier et autoriser les clients à l'aide des ressources d'[autorisation](#). Chaque autoriseur contient une référence à une fonction Lambda gérée par le client, une clé publique facultative pour valider les informations d'identification de l'appareil et des informations de configuration supplémentaires. Le schéma suivant illustre le flux de travail d'autorisation pour l'authentification personnalisée dans AWS IoT Core.



AWS IoT Coreflux de travail d'authentification et d'autorisation personnalisé

La liste suivante explique chaque étape du processus d'authentification et d'autorisation personnalisé.

1. Un appareil se connecte au point de terminaison de AWS IoT Core données d'un client à l'aide de l'un des appareils compatibles [the section called “Protocoles de communication des appareils” \(p. 89\)](#). L'appareil transmet des informations d'identification soit dans les champs d'en-tête de la demande, soit dans les paramètres de requête (pour les WebSockets protocoles HTTP Publish ou MQTT over), soit dans le champ nom d'utilisateur et mot de passe du message MQTT CONNECT (pour les protocoles MQTT et MQTT over). WebSockets
2. AWS IoT Core vérifie la présence de l'une des deux conditions suivantes :
 - La demande entrante spécifie un autorisateur.
 - Un dispositif d'autorisation par défaut est configuré pour le point de terminaison de AWS IoT Core données qui reçoit la demande.
3. (Facultatif) Si vous avez activé la signature par jeton, AWS IoT Core valide la signature de la demande 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 appeler la fonction Lambda.
4. La fonction Lambda reçoit les informations d'identification et les métadonnées de connexion contenues dans la demande et prend une décision d'authentification.
5. La fonction Lambda renvoie les résultats de la décision d'authentification ainsi qu'un document de AWS IoT Core politique qui spécifie les actions autorisées dans la connexion. La fonction Lambda renvoie également des informations qui indiquent la fréquence à laquelle les informations d'identification figurant dans la demande sont AWS IoT Core revalidées en appelant la fonction Lambda.
6. AWS IoT Core évalue l'activité de 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, elle est soumise à la tarification Lambda et à des limites de service, telles que le taux d'exécution simultanée. Pour plus d'informations sur la tarification Lambda, consultez Tarification [Lambda](#). Vous pouvez gérer la charge de votre fonction Lambda en ajustant les paramètres `disconnectAfterInSeconds` et `refreshAfterInSeconds` dans la réponse de votre fonction Lambda. Pour de plus amples informations sur le contenu de la réponse de votre fonction Lambda, veuillez consulter [the section called "Définition de votre fonction Lambda" \(p. 346\)](#)

Note

Si vous laissez la signature activée, vous pouvez empêcher le déclenchement excessif de votre Lambda par des clients non reconnus. Tenez compte de cela avant de désactiver la connexion dans votre autorisateur.

Note

La limite du délai d'expiration de la fonction Lambda pour l'autorisateur personnalisé est de 5 secondes.

Création et gestion d'autorisateurs personnalisés

AWS IoT Core implémente des schémas d'authentification et d'autorisation personnalisés à l'aide de [ressources d'autorisation](#). Chaque mécanisme d'autorisation est constitué des composants suivants :

- Nom : chaîne unique définie par l'utilisateur qui identifie l'autorisateur.
- ARN de la fonction Lambda : l'Amazon Resource Name (ARN) de la fonction Lambda qui met en œuvre la logique d'autorisation et d'authentification.
- Nom de clé du 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 la signature. Cette valeur est requise si la signature est activée dans votre autorisateur.
- Indicateur de désactivation de la signature (facultatif) : valeur booléenne qui indique si l'exigence de signature doit être désactivée sur les informations d'identification. Cela est utile pour les scénarios où la signature des informations d'identification n'a pas de sens, tels que les schémas d'authentification qui utilisent le nom d'utilisateur et le mot de passe MQTT. La valeur par défaut est `false`, donc la signature est activée par défaut.
- Clé publique de signature du jeton : clé publique AWS IoT Core utilisée pour valider la signature du jeton. Sa longueur minimale est de 2 048 bits. Cette valeur est requise si la signature est activée dans votre autorisateur.

Lambda vous facture le nombre de fois que votre fonction Lambda s'exécute et le temps nécessaire à l'exécution du code de votre fonction. Pour plus d'informations sur la tarification Lambda, consultez Tarification [Lambda](#). Pour de plus amples informations sur la création de fonctions Lambda, veuillez consulter le Guide du développeur [Lambda](#).

Note

Si vous laissez la signature activée, vous pouvez empêcher le déclenchement excessif de votre Lambda par des clients non reconnus. Tenez compte de cela avant de désactiver la connexion dans votre autorisateur.

Note

La limite du délai d'expiration de la fonction Lambda pour l'autorisateur personnalisé est de 5 secondes.

Définition de votre fonction Lambda

Lorsque vous AWS IoT Core invoquez votre autorisateur, il déclenche le Lambda associé à l'autorisateur avec un événement qui contient l'objet JSON suivant. L'exemple d'objet JSON contient tous les champs possibles. 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 contenant les valeurs suivantes.

- **isAuthenticated**: valeur booléenne indiquant si la demande est authentifiée.
- **principalId**: chaîne alphanumérique qui sert d'identifiant pour le jeton envoyé par la demande d'autorisation personnalisée. La valeur doit être une chaîne alphanumérique comportant au moins un caractère, et pas plus de 128, et correspondre au modèle d'expression régulière (regex) suivant : ([a-zA-Z0-9]{1,128}) Les caractères spéciaux qui ne sont pas alphanumériques ne sont pas autorisés à être utilisés avec principalId l'AWS IoT Coreentrée. Reportez-vous à la documentation relative aux autres AWS services si des caractères spéciaux non alphanumériques sont autorisés pour le principalId
- **policyDocuments**: liste de documents de AWS IoT Core politique au format JSON Pour plus d'informations sur la création de AWS IoT Core stratégies, consultez. [the section called “Stratégies AWS IoT Core” \(p. 357\)](#) Le nombre maximum de documents de politique est de 10 documents de politique. Chaque document de stratégie peut contenir un maximum de 2 048 caractères.
- **disconnectAfterInSeconds**: entier qui indique la durée maximale (en secondes) de la connexion à la AWS IoT Core passerelle. La valeur minimale est de 300 secondes et la valeur maximale de 86 400 secondes.
- **refreshAfterInSeconds**: entier qui indique l'intervalle entre les actualisations des politiques. Lorsque cet intervalle est dépassé, AWS IoT Core invoque la fonction Lambda pour permettre l'actualisation des politiques. 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` valeur doit contenir un document AWS IoT Core de politique valide. Pour plus d'informations sur AWS IoT Core les politiques, consultez[the section called “Stratégies AWS IoT Core” \(p. 357\)](#). Dans les connexions MQTT sur TLS et MQTT sur WebSockets connexions, met en AWS IoT Core cache cette politique pendant l'intervalle spécifié dans la valeur du champ `refreshAfterInSeconds`. Dans le cas des connexions HTTP, la fonction Lambda est appelée pour chaque demande d'autorisation, sauf si votre appareil utilise des connexions persistantes HTTP (également appelées maintien en activité HTTP ou réutilisation de la connexion HTTP). Vous pouvez choisir d'activer la mise en cache lors de la configuration de l'autorisateur. Pendant cet intervalle, AWS IoT Core autorise les actions dans une connexion établie par rapport à cette politique mise en cache sans déclencher à nouveau votre fonction Lambda. En cas d'échec lors de l'authentification personnalisée, AWS IoT Core met fin à la connexion. AWS IoT Core met également fin à la connexion si elle est restée ouverte plus longtemps que la valeur spécifiée dans le `disconnectAfterInSeconds` paramètre.

L'exemple suivant JavaScript contient un exemple de fonction Lambda Node.js qui recherche un mot de passe dans le message MQTT Connect avec une valeur de `test` et renvoie une politique accordant l'autorisation de se connecter AWS IoT Core avec un client nommé `myClientName` et de publier dans une rubrique contenant le même nom de client. S'il ne trouve pas le mot de passe attendu, il renvoie une politique 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;
}

```

La fonction Lambda précédente renvoie le code JSON suivant lorsqu'elle reçoit le mot de passe attendu de test dans le message MQTT Connect. Les valeurs des `principalId` propriétés `password` et seront celles du message MQTT Connect.

```

{
  "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:iot: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
    }
  ]
}

```

Création d'un mécanisme d'autorisation

Vous pouvez créer un autorisateur à l'aide de l'[CreateAuthorizerAPI](#). L'exemple suivant décrit la commande.

```
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 `signing-disabled` ce paramètre 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 y êtes obligé. La validation des signatures vous protège contre les appels excessifs de votre fonction Lambda à partir d'appareils inconnus. Vous ne pouvez pas mettre à jour le `signing-disabled` statut d'un autorisateur après l'avoir créé. Pour modifier ce comportement, vous devez créer un autre mécanisme d'autorisation personnalisé avec une valeur différente pour le `signing-disabled` paramètre.

Les valeurs des `tokenSigningPublicKeys` paramètres `tokenKeyName` et sont facultatives si vous avez désactivé la signature. Ce sont des valeurs obligatoires si la signature est activée.

Après avoir créé votre fonction Lambda et l'autorisateur personnalisé, vous devez explicitement autoriser le AWS IoT Core service à 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"
```

Tester vos autorisations

Vous pouvez utiliser l'[TestInvokeAuthorizerAPI](#) pour tester les valeurs d'invocation et de retour de votre autorisateur. Cette API vous permet de spécifier les métadonnées du protocole et de tester la validation de la signature dans votre autorisateur.

Les onglets suivants montrent comment utiliser l'AWS CLI pour tester votre autorisateur.

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 du `token-signature` paramètre est le jeton signé. Pour savoir comment obtenir cette valeur, veuillez consulter[the section called “Signer le jeton” \(p. 352\)](#).

Si votre autorisateur utilise un nom d'utilisateur et un mot de passe, vous pouvez transmettre ces informations à l'aide du `--mqtt-context` paramètre. Les onglets suivants indiquent comment utiliser l'`TestInvokeAuthorizerAPI` pour envoyer un objet JSON contenant un nom d'utilisateur, un mot de passe et un nom de client à votre autorisateur personnalisé.

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 autorisateurs à l'aide des API suivantes.

- [ListAuthorizers](#): Afficher tous les autorisateurs de votre compte.
- [DescribeAuthorizer](#): affiche les propriétés de l'autorisateur spécifié. Ces valeurs incluent la date de création, la date de dernière modification et d'autres attributs.
- [SetDefaultAuthorizer](#): Spécifie l'autorisateur par défaut pour vos points de terminaison AWS IoT Core de données. AWS IoT Core utilise cet autorisateur si un appareil ne transmet pas d'AWS IoT Core informations d'identification et ne spécifie pas d'autorisateur. Pour plus d'informations sur l'utilisation AWS IoT Core d'informations d'identification, consultez[the section called “Authentification client” \(p. 320\)](#).
- [UpdateAuthorizer](#): modifie le statut, le nom de la clé du jeton ou les clés publiques de l'autorisateur spécifié.
- [DeleteAuthorizer](#): Supprime l'autorisateur spécifié.

Note

Vous ne pouvez pas mettre à jour les exigences de signature d'un autorisateur. Cela signifie que vous ne pouvez pas désactiver la connexion à un système d'autorisation existant qui l'exige. Vous ne pouvez pas non plus exiger de vous connecter à un autorisateur existant qui n'en a pas besoin.

Connexion à l'AWS IoT Core aide d'une authentification personnalisée

Les appareils peuvent se connecter à AWS IoT Core l'aide d'une authentification personnalisée avec n'importe quel protocole prenant AWS IoT Core en charge la messagerie des appareils. Pour de plus amples informations sur les protocoles de communication pris en charge, veuillez consulter[the section called “Protocoles de communication des appareils” \(p. 89\)](#). Les données de connexion que vous transmettez à votre fonction d'autorisation Lambda 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” \(p. 346\)](#) Les sections suivantes expliquent comment se connecter pour s'authentifier à l'aide de chaque protocole pris en charge.

HTTPS

Les appareils qui envoient des données à AWS IoT Core l'aide de l'[API HTTP Publish](#) peuvent transmettre des informations d'identification via des en-têtes de requête ou des paramètres de requête dans leurs requêtes HTTP POST. Les appareils peuvent spécifier un mécanisme d'autorisation à invoquer à l'aide du paramètre d'x-amz-customauthorizer-nameen-tête ou de requête. Si la signature par jeton est activée dans votre système d'autorisation, vous devez transmettre le **token-key-name** et x-amz-customauthorizer-signature dans les en-têtes de requête ou dans les paramètres de requête. Notez que la **token-signature** valeur doit être codée en URL lorsque vous l'utilisez JavaScript depuis le navigateur.

Note

L'autorisateur client pour le protocole HTTPS prend uniquement en charge les opérations de publication. Pour de plus amples informations sur le protocole HTTPS, veuillez consulter[the section called “Protocoles de communication des appareils” \(p. 89\)](#).

Les exemples de requêtes suivants montrent comment vous transmettez ces paramètres à la fois dans les en-têtes de requête et dans 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: token-value
x-amz-customauthorizer-name: authorizer-name

//Passing credentials via query parameters
POST /topics/topic?qos=qos&x-amz-customauthorizer-signature=token-signature&token-key-name=token-value HTTP/1.1
```

MQTT

Les appareils qui se AWS IoT Core connectent via une connexion MQTT peuvent transmettre des informations d'identification via les password champs username et des messages MQTT. La username valeur peut également éventuellement contenir une chaîne de requête qui transmet des valeurs supplémentaires (y compris un jeton, une signature et le nom de l'autorisateur) à votre autorisateur. Vous pouvez utiliser cette chaîne de requête si vous souhaitez utiliser un schéma d'authentification basé sur des jetons au lieu de valeurs username et password.

Note

Les données du champ de mot de passe sont encodées en base64 par. AWS IoT Core Votre fonction Lambda doit le décoder.

L'exemple suivant contient une username chaîne qui contient des paramètres supplémentaires qui spécifient un jeton et une signature.

```
username?x-amz-customauthorizer-name=authorizer-name&x-amz-customauthorizer-signature=token-signature&token-key-name=token-value
```

Pour invoquer un autorisateur, les appareils qui se connectent à l'aide de MQTT et AWS IoT Core d'une authentification personnalisée doivent se connecter sur le port 443. Ils doivent également transmettre l'extension TLS ALPN (Application Layer Protocol Negotiation) avec une valeur de mqtt et l'extension SNI (Server Name Indication) avec le nom d'hôte de leur AWS IoT Core point de terminaison de données. Pour éviter d'éventuelles erreurs, la valeur de x-amz-customauthorizer-signature doit être codée en URL. Nous recommandons également vivement que les valeurs de x-amz-customauthorizer-name et token-key-name soient codées en URL. Pour de plus amples informations sur ces valeurs, veuillez consulter[the section called "Protocoles de communication des appareils" \(p. 89\)](#). La V2 [AWS IoT SDK pour appareils, kits SDK mobiles et client de AWS IoT l'appareil \(p. 1494\)](#) peut configurer ces deux extensions.

MQTT terminé WebSockets

Les appareils qui se connectent à AWS IoT Core à l'aide de MQTT over WebSockets peuvent 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 contenus dans la requête HTTP UPGRADE pour établir la WebSockets connexion.
- À travers les champs password username et du 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 de plus amples informations sur ces extensions, veuillez consulter[the section called "MQTT" \(p. 351\)](#). L'exemple suivant montre comment transmettre les informations d'identification via la demande de mise à niveau HTTP.

```
GET /mqtt HTTP/1.1
Host: your-endpoint
Upgrade: WebSocket
Connection: Upgrade
x-amz-customauthorizer-signature: token-signature
token-key-name: token-value
sec-WebSocket-Key: any random base64 value
sec-websocket-protocol: mqtt
sec-WebSocket-Version: websocket version
```

Signer le jeton

Vous devez signer le jeton avec la clé privée de la key pair publique-privée que vous avez utilisée lors de l'`create-authorizer` rappel. Les exemples suivants montrent comment créer la signature du jeton à l'aide d'une commande de type Unix et JavaScript. Ils utilisent l'algorithme de hachage SHA-256 pour coder 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 entraîner des problèmes dans les flux de travail d'authentification personnalisés et décrit les étapes permettant de les résoudre. Pour résoudre les problèmes le plus efficacement possible, activez les CloudWatch journaux AWS IoT Core et définissez le niveau de journalisation sur DEBUG. Vous pouvez activer CloudWatch les journaux dans la AWS IoT Core console (<https://console.aws.amazon.com/iot/>). Pour de plus amples informations sur l'activation et la configuration des journaux pour AWS IoT Core, veuillez consulter [the section called “Configurer la journalisation AWS IoT” \(p. 467\)](#).

Note

Si vous laissez le niveau de journalisation à DEBUG pendant de longues périodes, vous CloudWatch risquez de stocker de grandes quantités de données de journalisation. Cela peut augmenter vos CloudWatch frais. Envisagez d'utiliser la journalisation basée sur les ressources pour augmenter la précision uniquement pour les appareils d'un groupe d'objets particulier. Pour plus d'informations sur la journalisation basée sur les ressources, consultez [the section called “Configurer la journalisation AWS IoT” \(p. 467\)](#). De plus, lorsque vous avez terminé le dépannage, réduisez le niveau de journalisation à un niveau moins détaillé.

Avant de commencer à résoudre [the section called “Comprendre le flux de travail d'authentification personnalisé” \(p. 343\)](#) les problèmes, consultez une vue d'ensemble du processus d'authentification personnalisé. Cela vous permet de savoir où rechercher la source d'un problème.

Cette rubrique aborde les deux domaines suivants que vous devez étudier.

- Problèmes liés à la fonction Lambda de votre mécanisme d'autorisation.
- Problèmes liés à votre appareil.

Vérifiez si la fonction Lambda de votre autorisateur présente des problèmes

Procédez comme suit pour vous assurer que les tentatives de connexion de vos appareils appellent votre fonction Lambda.

1. Vérifiez quelle fonction Lambda est associée à votre système d'autorisation.

Vous pouvez le faire en appelant l'[DescribeAuthorizer](#) API ou en cliquant sur l'autorisateur souhaité dans la section Sécurisée de la AWS IoT Core console.

2. Vérifiez les mesures d'appel pour la fonction Lambda. Effectuez les étapes suivantes pour ce faire.

- a. Ouvrez la AWS Lambda console (<https://console.aws.amazon.com/lambda/>) et sélectionnez la fonction associée à votre autorisateur.
 - b. Choisissez l'onglet Surveiller et consultez les mesures correspondant à la période correspondant à votre problème.
3. Si aucune invocation ne s'affiche, vérifiez qu'il AWS IoT Core est autorisé à appeler votre fonction Lambda. Si des appels s'affichent, passez à l'étape suivante. Procédez comme suit pour vérifier que votre fonction Lambda dispose des autorisations requises.
 - a. Choisissez l'onglet Autorisations correspondant à votre fonction dans la AWS Lambda console.

- b. Trouvez la section Politique basée sur les ressources au bas de la page. Si votre fonction Lambda dispose des autorisations requises, la politique ressemble à l'exemple suivant.

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "Id123",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:lambda:us-east-1:111111111111:function:FunctionName",
      "Condition": {
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:iot:us-east-1:111111111111:authorizer/
AuthorizerName"
        },
        "StringEquals": {
          "AWS:SourceAccount": "111111111111"
        }
      }
    }
  ]
}
```

- c. Cette politique accorde l'InvokeFunction autorisation d'exercer votre fonction au AWS IoT Core directeur. Si vous ne le voyez pas, vous devrez l'ajouter à l'aide de l'[AddPermission API](#). 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'erreur. Une erreur peut indiquer que la fonction Lambda ne gère pas correctement l'événement de connexion qui AWS IoT Core lui est envoyé.

Pour plus d'informations sur la gestion de l'événement dans votre fonction Lambda, consultez. [the section called “Définition de votre fonction Lambda” \(p. 346\)](#) Vous pouvez utiliser la fonction de test de la AWS Lambda console (<https://console.aws.amazon.com/lambda/>) pour coder en dur les valeurs de test dans la fonction afin de vous assurer que la fonction gère correctement les événements.

5. Si vous voyez des appels sans erreur, mais que vos appareils ne parviennent pas à se connecter (ou à publier, à s'abonner et à recevoir des messages), le problème est peut-être dû au fait que la politique renvoyée par votre fonction Lambda n'autorise pas les actions que vos appareils essaient d'effectuer. Procédez comme suit pour déterminer si quelque chose ne va pas avec la politique renvoyée par la fonction.
 - a. Utilisez une requête Amazon CloudWatch Logs Insights pour analyser les journaux sur une courte période afin de détecter les défaillances. 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 qu'elle renvoie AWS IoT Core et l'événement qui déclenche la fonction. Vous pouvez utiliser ces journaux pour inspecter la politique créée par la fonction.

6. Si vous voyez des appels sans erreur, mais que vos appareils ne parviennent pas à se connecter (ou à publier, à s'abonner et à recevoir des messages), cela peut également être dû au fait que votre fonction Lambda dépasse le délai d'expiration. La limite du délai d'expiration de la fonction Lambda pour l'autorisateur personnalisé est de 5 secondes. Vous pouvez vérifier la durée de la fonction dans CloudWatch les journaux ou les statistiques.

Enquête sur les problèmes liés aux appareils

Si vous ne rencontrez aucun problème lors de l'invocation de votre fonction Lambda ou si vous ne rencontrez aucun problème avec la politique selon laquelle la fonction renvoie, recherchez les problèmes liés aux tentatives de connexion de vos appareils. Les demandes de connexion mal formées peuvent empêcher AWS IoT Core le déclenchement de votre autorisateur. Des problèmes de connexion peuvent survenir à la fois au niveau du protocole TLS et de la couche application.

Problèmes possibles liés à la couche TLS :

- Les clients doivent transmettre soit un en-tête de nom d'hôte (HTTP, MQTT overWebSockets), soit l'extension TLS d'indication du nom du serveur (HTTP, MQTT overWebSockets, MQTT) dans toutes les demandes d'authentification personnalisées. Dans les deux cas, la valeur transmise doit correspondre à l'un des points de terminaison de AWS IoT Core données de votre compte. 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 iot:Data`(pour les VeriSign terminaux existants)
- Les appareils qui utilisent une authentification personnalisée pour les connexions MQTT doivent également transmettre l'extension TLS ALPN (Application Layer Protocol Negotiation) avec une valeur de `.mqtt`
- L'authentification personnalisée est actuellement disponible uniquement sur le port 443.

Problèmes possibles liés à la couche d'application :

- Si la signature est activée (le `signingDisabled` champ est faux dans votre système d'autorisation), recherchez les problèmes de signature suivants.
 - Assurez-vous de transmettre la signature du jeton dans l'`x-amz-customauthorizer-signature` en-tête ou dans un paramètre de chaîne de requête.
 - Assurez-vous que le service ne signe pas une valeur autre que le jeton.
 - Assurez-vous de transmettre le jeton dans l'en-tête ou le paramètre de requête que vous avez spécifié dans le `token-key-name` champ de votre autorisateur.
- Assurez-vous que le nom de l'autorisateur que vous transmettez dans le paramètre `x-amz-customauthorizer-name` d'en-tête ou de chaîne de requête est valide ou qu'un autorisateur par défaut a été défini pour votre compte.

Autorisation

L'autorisation est le processus d'octroi d'autorisations à une identité authentifiée. Vous accordez des autorisations d'AWS IoT Core utilisation AWS IoT Core et des politiques IAM. Cette rubrique couvre les stratégies AWS IoT Core. Pour de plus amples informations sur les stratégies IAM, veuillez consulter [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#) et [Fonctionnement de AWS IoT avec IAM \(p. 419\)](#).

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 AWS IoT Core politiques et les politiques IAM sont utilisées AWS IoT Core pour contrôler les opérations qu'une identité (également appelée principal) 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 politique		
MQTT sur TLS/ TCP, authentification mutuelle TLS (port 8883 ou 443) †) (p. 89)	SDK pour les appareils AWS IoT	Certificats X.509	Stratégie AWS IoT Core		
MQTT via HTTPS/ WebSocket, authentification AWS SigV4 (port 443)	AWSKit SDK Mobile	Identity Amazon Cognito Authentity	IAM et politiques AWS IoT Core		
		Identity Amazon Cognito non authentity	Politique IAM		
		IAM, ou identité fédérée	Politique IAM		
HTTPS, authentification AWS Signature version 4 (port 443)	AWS CLI	Amazon Cognito, IAM ou identité fédérée	Politique IAM		
HTTPS, authentification mutuelle TLS (port 8443)	Pas de prise en charge SDK	Certificats X.509	Stratégie AWS IoT Core		

Protocole et mécanisme d'authentification	Kit SDK	Type d'identité	Type de politique		
HTTPS sur authentification personnalisée (Port 443)	SDK pour les appareils AWS IoT	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 politique		
Authentification de la version 4 de la AWS signature HTTPS (port 443)	AWS CLI	Amazon Cognito Identity Cognito	Politique IAM		
		IAM, ou identité fédérée	Politique IAM		

AWS IoT Core les politiques sont associées aux certificats X.509, aux identités Amazon Cognito ou aux groupes d'objets. Les stratégies IAM sont attachées à un utilisateur, un groupe ou un rôle IAM. Si vous utilisez la AWS IoT console ou l'AWS IoT Core interface de ligne de commande pour associer la politique (à un certificat, à Amazon Cognito Identity ou à un groupe d'objets), vous utilisez une AWS IoT Core politique. Dans le cas contraire, vous utilisez une politique IAM. AWS IoT Core les politiques associées à un groupe d'objets s'appliquent à tout élément de ce groupe d'objets. Pour que la AWS IoT Core politique soit prise en compte, le nom `clientId` et le nom de l'objet doivent correspondre.

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 politique peuvent prendre quelques minutes avant d'entrer en vigueur en raison de la façon dont les documents de politique sont mis en AWS IoT cache. En d'autres termes, l'accès à une ressource récemment autorisée peut prendre quelques minutes, et une ressource peut rester 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, suivez le cours [Approfondissement de AWS IoT Core l'authentification et de l'autorisation](#) sur le site Web de AWS formation et de certification.

Stratégies AWS IoT Core

Les stratégies AWS IoT Core sont des documents JSON. Ils suivent les mêmes conventions que les politiques IAM. AWS IoT Core prend en charge les politiques nommées, de sorte que de nombreuses identités peuvent faire référence au même document de politique. 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. Le plan de AWS IoT Core données comprend des opérations qui vous permettent de vous connecter au courtier de AWS IoT Core messages, d'envoyer et de recevoir des messages MQTT, et d'obtenir ou de mettre à jour Device Shadow d'un 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 politique peuvent prendre quelques minutes avant d'entrer en vigueur en raison de la façon dont les documents de politique sont mis en AWS IoT cache. En d'autres termes, l'accès à une ressource récemment autorisée peut prendre quelques minutes, et une ressource peut rester accessible pendant plusieurs minutes après la révocation de son accès.

AWS IoT Core les politiques peuvent être associées aux certificats X.509, aux identités Amazon Cognito et aux groupes d'objets. Les politiques associées à un groupe d'objets s'appliquent à tout élément de ce groupe. Pour que la politique entre en vigueur, le nom `clientId` et le nom de l'objet doivent correspondre. AWS IoT Core les politiques suivent la même logique d'évaluation que les politiques IAM. Par défaut, toutes les stratégies sont implicitement refusées. Une autorisation explicite dans une stratégie basée sur l'identité ou les ressources remplace le comportement par défaut. Un refus explicite dans n'importe quelle politique remplace toutes les autorisations. Pour plus d'informations, consultez la section [Logique d'évaluation des politiques](#) dans le Guide de AWS Identity and Access Management l'utilisateur.

Rubriques

- [Actions de stratégie AWS IoT Core \(p. 358\)](#)
- [Ressources d'action AWS IoT Core \(p. 360\)](#)
- [Variables de stratégie AWS IoT Core \(p. 362\)](#)
- [Prévention du député confus entre services \(p. 368\)](#)
- [Exemples de stratégies AWS IoT Core \(p. 369\)](#)
- [Autorisation avec les identités Amazon Cognito \(p. 399\)](#)

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 messagerie n'autorise pas deux clients ayant le même identifiant client à rester connectés en même temps. Une fois que le deuxième client se connecte, l'agent ferme la connexion existante. Utilisez cette `iot:Connect` autorisation pour vous assurer que seuls les clients autorisés utilisant un ID client spécifique peuvent se connecter.

`iot:GetRetainedMessage`

Représente l'autorisation d'obtenir des contenus d'un seul message conservé. Les messages conservés sont les messages qui ont été publiés avec l'indicateur RETAIN activé et stockés par AWS

IoT Core. Pour obtenir l'autorisation d'obtenir une liste de tous les messages conservés du compte, consultez [iot>ListRetainedMessages \(p. 359\)](#).

iot:ListRetainedMessages

Représente l'autorisation de récupérer des informations récapitulatives sur les messages conservés par le compte, mais pas sur le contenu des messages. Les messages conservés sont les messages qui ont été publiés avec l'indicateur RETAIN activé et stockés par AWS IoT Core. L'ARN de ressource spécifié pour cette action doit être*. Pour obtenir l'autorisation d'obtenir des contenus d'un seul message conservé, consultez [iot:GetRetainedMessage \(p. 358\)](#).

iot:Publish

Représente l'autorisation de publier une rubrique MQTT. Cette autorisation est vérifiée chaque fois qu'une demande PUBLISH est envoyée à l'agent. Vous pouvez l'utiliser pour autoriser les clients à publier selon des modèles de sujets 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'iot:Receive autorisation est confirmée chaque fois qu'un message est remis à un client. Comme cette autorisation est vérifiée à chaque diffusion, vous pouvez l'utiliser pour révoquer les autorisations accordées aux clients actuellement abonnés à une rubrique.

IoT : RetainPublish

Représente l'autorisation de publier un message MQTT avec l'indicateur RETAIN activé.

Note

Pour accorder l'autorisation iot:RetainPublish, vous devez également accorder l'autorisation iot:Publish.

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. Utilisez-le pour permettre aux clients de s'abonner à des sujets qui correspondent à des modèles de sujets spécifiques.

Note

Pour accorder l'autorisation iot:Subscribe, vous devez également accorder l'autorisation iot:Connect.

Actions relatives à la politique d'Device Shadow

iot>DeleteThingShadow

Représente l'autorisation de supprimer Device Shadow d'un objet.

L'iot>DeleteThingShadow autorisation est vérifiée chaque fois qu'une demande est faite pour supprimer le contenu de Device Shadow d'un objet.

iot:GetThingShadow

Représente l'autorisation de récupérer Device Shadow d'un objet.

L'iot:GetThingShadow autorisation est vérifiée chaque fois qu'une demande est faite pour récupérer le contenu de Device Shadow d'un objet.

`iot:ListNamedShadowsForThing`

Représente l'autorisation de répertorier un objet nommé Shadows.
L'`iot:ListNamedShadowsForThing` autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier un objet nommé Shadows.

`iot:UpdateThingShadow`

Représente l'autorisation de mettre à jour un shadow d'appareil.
L'`iot:UpdateThingShadow` autorisation est vérifiée chaque fois qu'une demande est faite pour mettre à jour le contenu de Device Shadow d'un 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.

Pour un exemple de politique d'exécution des tâches qui illustre cela, voir [the section called "Exemple de politique d'emploi de base" \(p. 398\)](#) qui fonctionne avec le protocole MQTT.

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.

AWS IoT CoreAction politique relative aux fournisseurs d'informations d'identification

`iot:AssumeRoleWithCertificate`

Représente l'autorisation d'appeler le fournisseur AWS IoT Core d'informations d'identification pour qu'il assume un rôle IAM avec une authentification basée sur des certificats.
L'`iot:AssumeRoleWithCertificate` autorisation est vérifiée chaque fois qu'une demande est faite au fournisseur d'AWS IoT Core d'informations d'identification pour qu'il assume un rôle.

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 du client	<code>arn:aws:iot:us-east-1:123456789012:client/myClientId</code>
iot:DeleteThingShadow	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
iot:DescribeJobExecution	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
iot:GetPendingJobExecutions	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
iot:GetRetainedMessage	topic	Un sujet de message conservé.	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
iot:GetThingShadow	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
iot>ListNamedShadowsForThing	thing	Tous les comptes	*
iot>ListRetainedMessagesForAllTopics	topic	Tous les comptes	*
iot:Publish	topic	Une chaîne de rubrique	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
iot:Receive	topic	Une chaîne de rubrique	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
iot:RetainPublish	topic	Rubrique à publier avec l'indicateur RETAIN activé.	<code>arn:aws:iot:us-east-1:123456789012:topic/myTopicName</code>
iot:StartNextPendingJobExecution	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>
iot:Subscribe	topicfilter	Une chaîne de filtre de rubrique	<code>arn:aws:iot:us-east-1:123456789012:topicfilter/myTopicFilter</code>
iot:UpdateJobExecution	thing	Le nom de l'objet	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code>

Action	Type de ressource	Nom de la ressource	Exemple d'ARN
<code>iot:UpdateThingShadow</code>	thing	Le nom de l'objet et le nom de l'ombre, le cas échéant	<code>arn:aws:iot:us-east-1:123456789012:thing/thingOne</code> <code>arn:aws:iot:us-east-1:123456789012:thing/thingOne/shadowOne</code>
<code>iot:AssumeRoleWithCertificate</code>	alias	Un alias de rôle qui pointe vers un ARN de rôle	<code>arn:aws:iot:us-east-1:123456789012:rolealiasCredentialProviderRole_alias</code>

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 ».

AWS IoT Core les politiques peuvent utiliser des caractères génériques et suivre une convention similaire à celle des politiques IAM. L'insertion d'un * astérisque dans la chaîne peut être traitée comme un caractère générique correspondant à n'importe quel caractère. Par exemple, vous pouvez l'utiliser * pour décrire plusieurs noms de rubriques MQTT dans l'Resourceattribut d'une politique. Les caractères + et # sont traités comme des chaînes littérales dans une politique. Pour obtenir un exemple de politique expliquant comment utiliser les caractères génériques, reportez-vous [Utilisation de caractères génériques dans MQTT et les politiques AWS IoT Core \(p. 376\)](#) à la section.

Vous pouvez également utiliser des variables de politique prédéfinies avec des valeurs fixes pour représenter des caractères qui, sinon, ont une signification spéciale. Ces caractères spéciaux incluent `$(*$($?))`, et `$($)`. Pour plus d'informations sur les variables de politique et les caractères spéciaux, voir [Éléments de stratégie IAM : variables et balises](#) et [Création d'une condition avec plusieurs clés ou valeurs](#).

Rubriques

- [Variables de stratégie AWS IoT Core de base \(p. 362\)](#)
- [Variables de stratégie d'objet \(p. 364\)](#)
- [Variables de stratégie AWS IoT Core de certificat X.509 \(p. 365\)](#)

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 AWS IoT Core politique suivante montre une stratégie qui utilise des variables de stratégie. `aws:SourceIp` peut être utilisé dans l'élément Condition de votre politique pour autoriser des mandataires à effectuer des demandes d'API uniquement dans une plage d'adresses spécifique. Pour obtenir des exemples, consultez [Autoriser les utilisateurs et les services cloud à utiliser Jobs AWS IoT \(p. 834\)](#).

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

```
{
    "Effect": "Allow",
    "Action": [
        "iot:Connect"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/clientid1"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "iot:Publish"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:ClientId}"
    ],
    "Condition": {
        "IpAddress": {
            "aws:SourceIp": "123.45.167.89"
        }
    }
}
]
```

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"
            ]
        }
    ]
}
```

Note

L'utilisation de la variable de politique \${iot:ClientId} avec n'Connect test pas recommandée. La valeur de n'étant pas vérifiéeClientId, un attacheur avec un identifiant client différent peut réussir la validation mais provoquer une déconnexion. Comme tout ClientId est autorisé, la définition d'un ID client aléatoire permet de contourner les politiques relatives aux groupes d'objets.

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.

- Utilisez l'[AttachThingPrincipal](#) API pour associer des certificats ou des principes (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 politique n'est disponible que lorsqu'un appareil se connecte via MQTT ou MQTT via le WebSocket protocole.

- `iot:Connection.Thing.ThingTypeName`

Cette variable est résolue en type d'objet associé à l'objet pour lequel la stratégie est évaluée. L'ID client de la WebSocket connexion MQTT/ doit être identique au nom de l'objet. Cette variable de politique n'est disponible que lors de la connexion via MQTT ou MQTT via le WebSocket protocole.

- `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. L'ID client de la WebSocket connexion MQTT/ doit être identique au nom de l'objet. Cette variable de politique n'est disponible que lors de la connexion via MQTT ou MQTT via le WebSocket protocole.

- `iot:Connection.Thing.IsAttached`

`iot:Connection.Thing.IsAttached: ["true"]` fait en sorte que seuls les appareils enregistrés AWS IoT et connectés au principal puissent accéder aux autorisations définies dans la politique. Vous pouvez utiliser cette variable pour empêcher un appareil de se connecter AWS IoT Core s'il présente un certificat qui n'est pas associé à un objet IoT figurant dans le AWS IoT Core registre. Cette variable contient des valeurs `true` ou `false` indique que l'objet de connexion est associé au certificat ou à l'identité Amazon Cognito dans le registre à l'aide de l'API. [AttachThingPrincipal](#) Le nom de l'objet est considéré comme identifiant 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'[RegisterCertificateAPI](#), le `certificateId` apparaît dans le corps de la réponse. Pour obtenir des informations sur votre certificat, vous pouvez utiliser `certificateId` l'entrée [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.

String length

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 0=Amazon.com Inc. L=Seattle ST=Washington C=US devient Amazon Web Service 0 \=Amazon.com Inc. L\=Seattle ST\=Washington C\=US.

Prévention du député confus entre services

Le problème de l'adjoint confus est un problème de sécurité dans lequel une entité qui n'a pas l'autorisation d'effectuer une action peut contraindre une entité plus privilégiée à effectuer cette action. Dans AWS, l'emprunt d'identité entre services peut entraîner le problème de député confus. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Pour limiter les autorisations à la ressource AWS IoT octroyées par à un autre service, nous vous recommandons d'utiliser les clés de condition `aws:SourceAccount` globale `aws:SourceArn` les clés de condition globale dans les politiques de ressources. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de politique.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de condition `aws:SourceArn` globale avec le nom de ressource Amazon Resource Name (ARN) complet de la ressource. Car AWS IoT, vous `aws:SourceArn` devez respecter le format `:arn:aws:iot:region:account-id:*`. Assurez-vous que la *région correspond à votre*

AWS IoT région et que l'identifiant du *compte correspond à l'identifiant* de votre compte client.

L'exemple suivant montre comment éviter le problème de député confus en utilisant les clés de condition aws:SourceAccount globale aws:SourceArn et les clés de condition globale dans la stratégie de confiance des AWS IoT rôles.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service": "iot.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole",  
            "Condition": {  
                "StringEquals": {  
                    "aws:SourceAccount": "123456789012"  
                },  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:iot:us-east-1:123456789012:/*"  
                }  
            }  
        }  
    ]  
}
```

Exemples de stratégies AWS IoT Core

Les exemples de politiques présentés dans cette section illustrent les documents de politique utilisés pour effectuer des tâches courantes dans AWS IoT Core. Vous pouvez les utiliser comme exemples pour commencer à créer les politiques de vos solutions.

Les exemples de cette section utilisent les éléments de stratégie suivants :

- [the section called “Actions de stratégie AWS IoT Core” \(p. 358\)](#)
- [the section called “Ressources d'action AWS IoT Core” \(p. 360\)](#)
- [the section called “Exemples de politiques basées sur l'identité” \(p. 438\)](#)
- [the section called “Variables de stratégie AWS IoT Core de base” \(p. 362\)](#)
- [the section called “Variables de stratégie AWS IoT Core de certificat X.509” \(p. 365\)](#)

Exemples de stratégie dans cette section :

- [Exemples de stratégies de connexion \(p. 369\)](#)
- [Exemples de stratégie de publication/abonnement \(p. 376\)](#)
- [Exemples de stratégies de connexion et de publication \(p. 390\)](#)
- [Exemples de politiques de conservation des messages \(p. 391\)](#)
- [Exemples de stratégies de certificat \(p. 393\)](#)
- [Exemples de stratégies d'objet \(p. 397\)](#)
- [Exemple de politique d'emploi de base \(p. 398\)](#)

Exemples de stratégies de connexion

La politique suivante refuse l'accès aux identifiants clients client1 et client2 à la connexion AWS IoT Core, tout en autorisant les appareils à se connecter à l'aide d'un identifiant client. L'ID client correspond au nom d'un objet enregistré dans le AWS IoT Core registre et associé au principal utilisé pour la connexion :

Note

Pour les appareils enregistrés, nous vous recommandons d'utiliser des [variables de politique \(p. 364\)](#) d'objet pour les Connect actions et d'associer l'objet au principal utilisé pour la connexion.

```
{  
    "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}"  
            ],  
            "Condition": {  
                "Bool": {  
                    "iot:Connection.Thing.IsAttached": "true"  
                }  
            }  
        }  
    ]  
}
```

La politique suivante accorde l'autorisation de se connecter à l'AWS IoT Core à l'ID client `client1`. Cet exemple de politique concerne les appareils non enregistrés.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/client1"  
            ]  
        }  
    ]  
}
```

Exemples de politique de sessions persistantes MQTT

`connectAttributes` vous permettent de spécifier les attributs que vous souhaitez utiliser dans votre message de connexion dans vos politiques IAM, tels que `PersistentConnect` et `LastWill`. Pour plus d'informations, veuillez consulter [Utilisation de ConnectAttributes \(p. 102\)](#).

La politique suivante permet de se connecter à la `PersistentConnect` fonctionnalité :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Connect"  
      ],  
      "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",  
      "Condition": {  
        "ForAllValues:StringEquals": {  
          "iot:ConnectAttributes": [  
            "PersistentConnect"  
          ]  
        }  
      }  
    ]  
  ]  
}
```

La politique suivante l'interdit PersistentConnect, mais d'autres fonctionnalités sont autorisées :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Connect"  
      ],  
      "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",  
      "Condition": {  
        "ForAllValues:StringNotEquals": {  
          "iot:ConnectAttributes": [  
            "PersistentConnect"  
          ]  
        }  
      }  
    ]  
  ]  
}
```

La politique ci-dessus peut également être exprimée en utilisant StringEquals, toute autre fonctionnalité, y compris une nouvelle fonctionnalité, est autorisée :

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Connect"  
      ],  
      "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",  
    },  
    {  
      "Effect": "Deny",  
      "Action": [  
        "iot:Connect"  
      ],  
      "Resource": "*",  
      "Condition": {  
        "ForAnyValue:StringEquals": {  
          "iot:ConnectAttributes": [  
            "PersistentConnect"  
          ]  
        }  
      }  
    }  
  ]  
}
```

```
    "iot:ConnectAttributes": [
        "PersistentConnect"
    ]
}
]
}
}
```

La politique suivante autorise la connexion par les deux LastWill moyens PersistentConnect et aucune autre nouvelle fonctionnalité n'est autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        ]
    }
}
```

La politique suivante autorise les clients à se connecter correctement avec ou sans LastWill qu'aucune autre fonctionnalité ne soit autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            }
        ]
    }
}
```

La politique suivante autorise uniquement la connexion à l'aide des fonctionnalités par défaut :

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

```

```
"Effect": "Allow",
"Action": [
    "iot:Connect"
],
"Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
"Condition": {
    "ForAllValues:StringEquals": {
        "iot:ConnectAttributes": []
    }
}
]
```

La politique suivante autorise la connexion uniquement avec `PersistentConnect`. Toute nouvelle fonctionnalité est autorisée tant que la connexion l'utilise `PersistentConnect` :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect"
                    ]
                }
            }
        ]
    ]
}
```

La politique suivante indique que la connexion doit avoir à la fois une `LastWill` utilisant `PersistentConnect` et qu'aucune nouvelle fonctionnalité n'est autorisée :

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "PersistentConnect",
                        "LastWill"
                    ]
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "*"
        }
    ]
}
```

```

    "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 été PersistentConnect appliquéeLastWill, mais peut l'être. 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": "arn:aws:iot:us-east-1:123456789012:client/client1",
        }
    ]
}

```

```

        "Condition": {
            "ForAllValues:StringEquals": [
                "iot:ConnectAttributes": [
                    "LastWill"
                ]
            ]
        }
    }
}

```

La politique suivante autorise uniquement les clients qui ont une rubrique "my/lastwill/topicName" correspondante LastWill à se connecter. Toutes les fonctionnalités sont autorisées tant qu'elles utilisent la LastWill rubrique :

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "Condition": {
                "ArnEquals": {
                    "iot>LastWillTopic": "arn:aws:iot:region:account-id:topic/my/lastwill/
topicName"
                }
            }
        }
    ]
}

```

La politique suivante autorise uniquement la connexion propre à l'aide d'une fonctionnalité spécifiqueLastWillTopic. N'importe quelle fonctionnalité est autorisée tant qu'elle utilise LastWillTopic :

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:client/client1",
            "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"
    ]
}
}
```

Exemples de stratégie de publication/abonnement

La politique que vous utilisez dépend de la façon dont vous vous connectez à AWS IoT Core. Vous pouvez vous connecter à AWS IoT Core l'aide d'un client MQTT, HTTP ou WebSocket. Lorsque vous vous connectez à un client MQTT, vous vous authentifiez à l'aide d'un certificat X.509. Lorsque vous vous connectez via HTTP ou le WebSocket protocole, vous vous authentifiez avec Signature Version 4 et Amazon Cognito.

Note

Pour les appareils enregistrés, nous vous recommandons d'utiliser des [variables de politique \(p. 364\)](#) d'objet pour les Connect actions et d'associer l'objet au principal utilisé pour la connexion.

Dans cette section :

- [Utilisation de caractères génériques dans MQTT et les politiques AWS IoT Core \(p. 376\)](#)
- [Règles relatives à la publication, à l'abonnement et à la réception de messages vers/provenant de sujets spécifiques \(p. 377\)](#)
- [Règles relatives à la publication, à l'abonnement et à la réception de messages vers/depuis des sujets comportant un préfixe spécifique \(p. 379\)](#)
- [Règles relatives à la publication, à l'abonnement et à la réception de messages vers/provenant de sujets spécifiques à chaque appareil \(p. 380\)](#)
- [Règles relatives à la publication, à l'abonnement et à la réception de messages vers/depuis des sujets dont le nom du sujet contient un attribut d'objet \(p. 382\)](#)
- [Politiques visant à refuser la publication de messages sur les sous-rubriques d'un nom de rubrique \(p. 384\)](#)
- [Politiques permettant de refuser de recevoir des messages provenant des sous-rubriques d'un nom de rubrique \(p. 385\)](#)
- [Règles relatives à l'abonnement à des sujets utilisant des caractères génériques MQTT \(p. 387\)](#)
- [Politiques pour le protocole HTTP et les WebSocket clients \(p. 388\)](#)

Utilisation de caractères génériques dans MQTT et les politiques AWS IoT Core

Le MQTT et AWS IoT Core les politiques ont des caractères génériques différents et vous devez les choisir après mûre réflexion. Dans MQTT, les caractères génériques + et # sont utilisés dans les [filtres de sujets MQTT](#) pour s'abonner à plusieurs noms de sujets. AWS IoT Core les politiques utilisent * et ? comme caractères génériques et respectent les conventions des politiques [IAM](#). Dans un document de politique, le * représente n'importe quelle combinaison de caractères et un point d'interrogation ? représente un seul caractère quelconque. Dans les documents de politique, les caractères génériques MQTT + et # sont considérés comme des caractères sans signification particulière. Pour décrire plusieurs noms de rubriques et filtres de rubriques dans l'`resource` attribut d'une politique, utilisez les caractères ? génériques * et à la place des caractères génériques MQTT.

Lorsque vous choisissez des caractères génériques à utiliser dans un document de politique, tenez compte du fait que le * caractère n'est pas limité à un seul niveau de sujet car il se trouve dans un filtre de sujet MQTT. + Pour limiter une spécification générique à un seul niveau de filtre de rubrique MQTT, pensez à

utiliser plusieurs caractères. ? Pour plus d'informations sur l'utilisation de caractères génériques dans une ressource de politique et d'autres exemples de ce à quoi ils correspondent, consultez la section [Utilisation de caractères génériques dans les ARN des ressources](#).

Le tableau ci-dessous présente les différents caractères génériques utilisés dans MQTT et les AWS IoT Core politiques applicables aux clients MQTT.

Caractère générique	Est-ce un caractère générique MQTT	Exemple en MQTT	La AWS IoT Core politique est-elle un caractère générique	Exemple de AWS IoT Core politiques pour les clients MQTT
#	Oui	some/#	Non	N/A
+	Oui	some/+/topic	Non	N/A
*	Non	N/A	Oui	topicfilter/some/*/topic topicfilter/some/sensor*/topic
?	Non	N/A	Oui	topic/some/?????/topic topicfilter/some/sensor??/?topic

Règles relatives à la publication, à l'abonnement et à la réception de messages vers/provenant de sujets spécifiques

Vous trouverez ci-dessous des exemples d'appareils enregistrés et non enregistrés permettant de publier, de s'abonner et de recevoir des messages vers/depuis le sujet intitulé « some_specific_topic ». Les exemples le soulignent également Publish et Receive utilisent « topic » comme ressource, et Subscribe « topicfilter » comme ressource.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Il fournit également Publish Subscribe des Receive autorisations pour le sujet nommé « some_specific_topic ».

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": {
          "iot:Connection.Thing.IsAttached": "true"
        }
      }
    }
  ]
}
```

```

},
{
  "Effect": "Allow",
  "Action": [
    "iot:Publish"
  ],
  "Resource": [
    "arn:aws:iot:us-east-1:123456789012:topic/some_specific_topic"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iot:Subscribe"
  ],
  "Resource": [
    "arn:aws:iot:us-east-1:123456789012:topicfilter/some_specific_topic"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "iot:Receive"
  ],
  "Resource": [
    "arn:aws:iot:us-east-1:123456789012:topic/some_specific_topic"
  ]
}
]
}

```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Il fournit également Publish Subscribe des Receive autorisations pour le sujet nommé « some_specific_topic ».

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/clientId1",
        "arn:aws:iot:us-east-1:123456789012:client/clientId2",
        "arn:aws:iot:us-east-1:123456789012:client/clientId3"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/some_specific_topic"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ]
    }
  ]
}

```

```
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topicfilter/some_specific_topic"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Receive"
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topic/some_specific_topic"
        ]
    }
]
```

Règles relatives à la publication, à l'abonnement et à la réception de messages vers/depuis des sujets comportant un préfixe spécifique

Vous trouverez ci-dessous des exemples d'appareils enregistrés et non enregistrés permettant de publier, de s'abonner et de recevoir des messages vers/depuis des sujets préfixés par « topic_prefix ».

Note

Notez l'utilisation du caractère générique * dans cet exemple. Bien que le caractère générique* soit utile pour fournir des autorisations pour plusieurs noms de rubriques dans une seule instruction, il peut avoir des conséquences imprévues en accordant aux appareils plus de priviléges que ce qui est requis. Nous vous recommandons donc de n'utiliser le caractère générique* qu'après mûre réflexion.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Il fournit également des Publish Receive autorisations pour Subscribe les sujets préfixés par « topic_prefix ».

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
            ],
            "Condition": {
                "Bool": {
                    "iot:Connection.Thing.IsAttached": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
                "iot:Receive"
            ],
            "Resource": [

```

```
    "arn:aws:iot:us-east-1:123456789012:topic/topic_prefix*"
]
},
{
  "Effect": "Allow",
  "Action": [
    "iot:Subscribe"
  ],
  "Resource": [
    "arn:aws:iot:us-east-1:123456789012:topicfilter/topic_prefix*"
  ]
}
]
```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Il fournit également des Publish Receive autorisations pour Subscribe les sujets préfixés par « topic_prefix ».

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/clientId1",
        "arn:aws:iot:us-east-1:123456789012:client/clientId2",
        "arn:aws:iot:us-east-1:123456789012:client/clientId3"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish",
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/topic_prefix*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topicfilter/topic_prefix*"
      ]
    }
  ]
}
```

Règles relatives à la publication, à l'abonnement et à la réception de messages vers/provenant de sujets spécifiques à chaque appareil

Vous trouverez ci-dessous des exemples d'appareils enregistrés et non enregistrés permettant de publier, de s'abonner et de recevoir des messages vers/depuis des sujets spécifiques à l'appareil en question.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Il permet de publier sur le sujet spécifique à l'objet (sensor/device/\${iot:Connection.Thing.ThingName}) ainsi que de s'y abonner et d'en recevoir des informations. command/device/\${iot:Connection.Thing.ThingName} Si le nom de l'objet dans le registre est « objet 1 », l'appareil sera autorisé à publier sur la rubrique « capteur/appareil/objet 1 », ainsi qu'à s'abonner à la rubrique « commande/appareil/objet 1 » et à recevoir des informations provenant de ce sujet.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"  
            ],  
            "Condition": {  
                "Bool": {  
                    "iot:Connection.Thing.IsAttached": "true"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/sensor/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topicfilter/command/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/command/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        }  
    ]  
}
```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Il permet de publier sur la rubrique spécifique au client (`sensor/device/${iot:ClientId}`), ainsi que de s'abonner à la rubrique spécifique au client () et d'en recevoir des informations. `command/device/${iot:ClientId}` Si l'appareil se connecte à clientId en tant que ClientID1, il sera autorisé à publier sur la rubrique « Capteur/Device/ClientID1 », à s'abonner à la rubrique et à en recevoir des informations. `device/clientId1/command`

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/clientId1",  
                "arn:aws:iot:us-east-1:123456789012:client/clientId2",  
                "arn:aws:iot:us-east-1:123456789012:client/clientId3"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/sensor/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topicfilter/command/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/command/device/  
${iot:Connection.Thing.ThingName}"  
            ]  
        }  
    ]  
}
```

Règles relatives à la publication, à l'abonnement et à la réception de messages vers/depuis des sujets dont le nom du sujet contient un attribut d'objet

L'exemple suivant montre comment les appareils enregistrés peuvent publier, s'abonner et recevoir des messages vers/depuis des sujets dont les noms incluent des attributs d'objet.

Note

Les attributs d'objet n'existent que pour les appareils enregistrés dans AWS IoT Core le registre. Il n'existe aucun exemple correspondant pour les appareils non enregistrés.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Il permet de publier dans le sujet (sensor/\${iot:Connection.Thing.Attributes[version]}), de s'y abonner et de recevoir des informations sur le sujet (command/\${iot:Connection.Thing.Attributes[location]}) lorsque le nom du sujet inclut des attributs d'objet. Si le nom de l'objet dans le registre contient version=v1 et location=Seattle, l'appareil sera autorisé à publier sur la rubrique « sensor/v1 », à s'abonner à la rubrique « Command/Seattle » et à recevoir des informations depuis cette rubrique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": {
          "iot:Connection.Thing.IsAttached": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/sensor/
${iot:Connection.Thing.Attributes[version]}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Subscribe"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topicfilter/command/
${iot:Connection.Thing.Attributes[location]}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/command/
${iot:Connection.Thing.Attributes[location]}"
      ]
    }
  ]
}
```

```
    ]  
}
```

Unregistered devices

Comme les attributs d'objet n'existent que pour les appareils enregistrés dans AWS IoT Core le registre, il n'existe aucun exemple correspondant pour les objets non enregistrés.

Politiques visant à refuser la publication de messages sur les sous-rubriques d'un nom de rubrique

Vous trouverez ci-dessous des exemples d'appareils enregistrés et non enregistrés permettant de publier des messages sur tous les sujets à l'exception de certains sous-sujets.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Il autorise la publication dans tous les sujets préfixés par « département/ », mais pas dans le sous-sujet « département/administrateurs ».

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Connect"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"  
      ],  
      "Condition": {  
        "Bool": {  
          "iot:Connection.Thing.IsAttached": "true"  
        }  
      }  
    },  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Publish"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-east-1:123456789012:topic/department/*"  
      ]  
    },  
    {  
      "Effect": "Deny",  
      "Action": [  
        "iot:Publish"  
      ],  
      "Resource": [  
        "arn:aws:iot:us-east-1:123456789012:topic/department/admins"  
      ]  
    }  
  ]  
}
```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Il autorise

la publication dans tous les sujets préfixés par « département/ », mais pas dans le sous-sujet « département/administrateurs ».

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/clientId1",
                "arn:aws:iot:us-east-1:123456789012:client/clientId2",
                "arn:aws:iot:us-east-1:123456789012:client/clientId3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/department/*"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/department/admins"
            ]
        }
    ]
}
```

Politiques permettant de refuser de recevoir des messages provenant des sous-rubriques d'un nom de rubrique

Vous trouverez ci-dessous des exemples d'appareils enregistrés et non enregistrés auxquels vous pouvez vous abonner et recevoir des messages provenant de sujets comportant des préfixes spécifiques, à l'exception de certains sous-sujets.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. Cette politique permet aux appareils de s'abonner à n'importe quel sujet préfixé par « topic_prefix ». NotResourceEn utilisant dans la déclaration pouriot:Receive, nous autorisons l'appareil à recevoir des messages concernant tous les sujets auxquels il est abonné, à l'exception des sujets préfixés par « topic_prefix/restricted ». Par exemple, avec cette politique, les appareils peuvent s'abonner à « topic_prefix/topic1 » et même à « topic_prefix/restricted ». Toutefois, ils ne recevront que des messages provenant du sujet « topic_prefix/topic1 » et aucun message provenant du sujet « topic_prefix/restricted ».

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/clientId1"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/topic1"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/topic/restricted"
            ]
        }
    ]
}
```

```

    "Effect": "Allow",
    "Action": [
        "iot:Connect"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
    ],
    "Condition": {
        "Bool": {
            "iot:Connection.Thing.IsAttached": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": "iot:Subscribe",
    "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/topic_prefix/*"
},
{
    "Effect": "Allow",
    "Action": "iot:Receive",
    "NotResource": "arn:aws:iot:us-east-1:123456789012:topic/topic_prefix/restricted/*"
}
]
}

```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Cette politique permet aux appareils de s'abonner à n'importe quel sujet préfixé par « topic_prefix ». NotResourceEn utilisant dans la déclaration pouriot:Receive, nous autorisons l'appareil à recevoir des messages concernant tous les sujets auxquels il est abonné, à l'exception des sujets préfixés par « topic_prefix/restricted ». Par exemple, avec cette politique, les appareils peuvent s'abonner à « topic_prefix/topic1 » et même à « topic_prefix/restricted ». Toutefois, ils ne recevront que des messages provenant du sujet « topic_prefix/topic1 » et aucun message provenant du sujet « topic_prefix/restricted ».

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/clientId1",
                "arn:aws:iot:us-east-1:123456789012:client/clientId2",
                "arn:aws:iot:us-east-1:123456789012:client/clientId3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": "iot:Subscribe",
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/topic_prefix/*"
        },
        {
            "Effect": "Allow",
            "Action": "iot:Receive",
            "NotResource": "arn:aws:iot:us-east-1:123456789012:topic/topic_prefix/restricted/*"
        }
    ]
}

```

}

Règles relatives à l'abonnement à des sujets utilisant des caractères génériques MQTT

Les caractères génériques MQTT + et # sont traités comme des chaînes littérales, mais ils ne sont pas traités comme des caractères génériques lorsqu'ils sont utilisés dans des politiques. AWS IoT Core Dans MQTT, + et # sont traités comme des caractères génériques uniquement lors de l'abonnement à un filtre de sujet, mais comme une chaîne littérale dans tous les autres contextes. Nous vous recommandons de n'utiliser ces caractères génériques MQTT dans le cadre des AWS IoT Core politiques qu'après mûre réflexion.

Vous trouverez ci-dessous des exemples d'objets enregistrés et non enregistrés utilisant des caractères génériques MQTT dans les politiques. Ces caractères génériques sont traités comme des chaînes littérales.

Registered devices

Pour les appareils enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter avec clientId qui correspond au nom d'un objet figurant dans AWS IoT Core le registre. La politique autorise les appareils à s'abonner aux rubriques « département/+/employés » et « emplacement/# ». Notez que les sigles + et # étant considérés comme des chaînes littérales dans AWS IoT Core les politiques, les appareils peuvent s'abonner à la rubrique « département/+/ employés » mais pas à la rubrique « département/ingénierie/employés ». De même, les appareils peuvent s'abonner à la rubrique « localisation/# » mais pas à la rubrique « Localisation/Seattle ». Toutefois, une fois que l'appareil est abonné à la rubrique « département/+/ employés », la politique lui permettra de recevoir des messages concernant la rubrique « département/ingénierie/employés ». De même, une fois que l'appareil s'est abonné au sujet « localisation/# », il recevra également des messages concernant le sujet « Localisation/Seattle ».

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
      ],
      "Condition": {
        "Bool": {
          "iot:Connection.Thing.IsAttached": "true"
        }
      }
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/department/+/employees"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/location/#"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:us-east-1:123456789012:topic/*"
    }
  ]
}
```

```
    ]  
}
```

Unregistered devices

Pour les appareils qui ne sont pas enregistrés dans AWS IoT Core le registre, la politique suivante permet aux appareils de se connecter à l'aide de ClientID1, ClientID2 ou ClientID3. Cette politique permet aux appareils de s'abonner aux rubriques « département/+/employés » et « emplacement/numéro ». Notez que les sigles + et # étant considérés comme des chaînes littérales dans AWS IoT Core les politiques, les appareils peuvent s'abonner à la rubrique « département/+/employés » mais pas à la rubrique « département/ingénierie/employés ». De même, les appareils peuvent s'abonner à la rubrique « localisation/# » mais pas à la rubrique « Localisation/Seattle ». Toutefois, une fois que l'appareil est abonné à la rubrique « département/+ employés », la politique lui permettra de recevoir des messages concernant la rubrique « département/ingénierie/employés ». De même, une fois que l'appareil s'est abonné au sujet « localisation/# », il recevra également des messages concernant le sujet « Localisation/Seattle ».

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/clientId1",  
                "arn:aws:iot:us-east-1:123456789012:client/clientId2",  
                "arn:aws:iot:us-east-1:123456789012:client/clientId3"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/department/+/  
employees"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:us-east-1:123456789012:topicfilter/location/#"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iot:Receive",  
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/*"  
        }  
    ]  
}
```

Politiques pour le protocole HTTP et les WebSocket clients

Lorsque vous vous connectez via HTTP ou le WebSocket protocole, vous vous authentifiez avec Signature Version 4 et Amazon Cognito. 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 à des utilisateurs invités qui ne s'authentifient pas avec un fournisseur d'identité. Amazon Cognito fournit un identifiant et des informations d'AWSidentification uniques pour prendre en charge les identités non authentifiées.

Pour plus d'informations, veuillez consulter [the section called “Autorisation avec les identités Amazon Cognito” \(p. 399\)](#).

Pour les opérations suivantes, AWS IoT Core utilise AWS IoT Core les politiques associées aux identités Amazon Cognito (via l'`AttachPolicyAPI`) afin de limiter les autorisations associées au pool d'identités Amazon Cognito avec des identités authentifiées.

- `iot:Connect`
- `iot:Publish`
- `iot:Subscribe`
- `iot:Receive`
- `iot:GetThingShadow`
- `iot:UpdateThingShadow`
- `iot>DeleteThingShadow`

Cela signifie qu'une identité Amazon Cognito doit être autorisée par la politique de rôle IAM associée au pool et par la AWS IoT Core politique associée à l'identité Amazon Cognito via l'API AWS IoT Core `AttachPolicy`.

Les utilisateurs authentifiés et non authentifiés sont des types d'identité différents. Si vous n'associez pas de AWS IoT politique à Amazon Cognito Identity, un utilisateur authentifié ne reçoit pas d'autorisation AWS IoT et n'a pas accès aux AWS IoT ressources et aux actions.

Note

Pour les autres AWS IoT Core opérations ou pour les identités non authentifiées, AWS IoT Core ne limite pas les autorisations associées au rôle du pool d'identités Amazon Cognito. Pour les identités authentifiées et non authentifiées, il s'agit de la politique la plus permissive que nous vous recommandons d'associer au rôle du pool Amazon Cognito.

HTTP

Pour autoriser les identités Amazon Cognito non authentifiées à publier des messages via HTTP sur un sujet spécifique à l'identité Amazon Cognito, associez la politique IAM suivante au rôle du pool Amazon Cognito Identity :

```
{  
    "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 les utilisateurs authentifiés, associez la politique précédente au rôle du pool Amazon Cognito Identity et à Amazon Cognito Identity à l'aide de l'API AWS IoT Core [AttachPolicy](#).

Note

Lorsque vous autorisez les identités Amazon Cognito, prenez en AWS IoT Core compte les deux politiques et accordez le moins de priviléges spécifiés. Une action n'est autorisée que si les deux stratégies autorisent l'action demandée. Si l'une des stratégies empêche une action, cette action n'est pas autorisée.

MQTT

Pour autoriser les identités Amazon Cognito non authentifiées à publier des messages MQTT WebSocket sur un sujet spécifique à l'identité Amazon Cognito de votre compte, associez la politique IAM suivante au rôle du pool Amazon Cognito Identity :

```
{  
    "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 les utilisateurs authentifiés, associez la politique précédente au rôle du pool Amazon Cognito Identity et à Amazon Cognito Identity à l'aide de l'API AWS IoT Core [AttachPolicy](#)

Note

Lorsque vous autorisez des identités Amazon Cognito, prenez en AWS IoT Core compte les deux et accordez le moins de priviléges spécifiés. Une action n'est autorisée que si les deux stratégies autorisent l'action demandée. Si l'une des stratégies empêche une action, cette action n'est pas autorisée.

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 politiques de conservation des messages

L'utilisation [des messages conservés \(p. 97\)](#) nécessite des politiques spécifiques. Les messages conservés sont des messages MQTT publiés avec l'indicateur RETAIN activé et stockés par AWS IoT Core. Cette section présente des exemples de politiques qui autorisent les utilisations courantes des messages conservés.

Dans cette section :

- [Politique de connexion et de publication des messages conservés \(p. 391\)](#)
- [Politique de connexion et de publication des messages Will conservés \(p. 392\)](#)
- [Politique visant à répertorier et à récupérer les messages conservés \(p. 392\)](#)

Politique de connexion et de publication des messages conservés

Pour qu'un appareil puisse publier des messages conservés, il doit être capable de se connecter, de publier (n'importe quel message MQTT) et de publier des messages MQTT conservés. La politique suivante accorde ces autorisations pour le sujet : `device/sample/configuration` au client `device1`. Pour un autre exemple d'autorisation de connexion, reportez-vous à la section [the section called “Exemples de stratégies de connexion et de publication” \(p. 390\)](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/device1"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/device1/*"  
            ]  
        }  
    ]  
}
```

```
    "iot:Publish",
    "iot:RetainPublish"
],
"Resource": [
    "arn:aws:iot:us-east-1:123456789012:topic/device/sample/configuration"
]
}
}
```

Politique de connexion et de publication des messages Will conservés

Les clients peuvent configurer un message qui AWS IoT Core sera publié en cas de déconnexion inattendue du client. MQTT appelle un tel message un [message Will](#). Un client doit ajouter une condition supplémentaire à son autorisation de connexion pour les inclure.

Le document de politique suivant autorise tous les clients à se connecter et à publier un message Will, identifié par son sujet `will`, qui AWS IoT Core sera également conservé.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/device1"
            ],
            "Condition": {
                "ForAllValues:StringEquals": {
                    "iot:ConnectAttributes": [
                        "LastWill"
                    ]
                }
            },
            {
                "Effect": "Allow",
                "Action": [
                    "iot:Publish",
                    "iot:RetainPublish"
                ],
                "Resource": [
                    "arn:aws:iot:us-east-1:123456789012:topic/will"
                ]
            }
        ]
    }
}
```

Politique visant à répertorier et à récupérer les messages conservés

Les services et les applications peuvent accéder aux messages conservés sans avoir à prendre en charge un client MQTT en appelant [ListRetainedMessages](#) et [GetRetainedMessage](#). Les services et applications qui appellent ces actions doivent être autorisés à l'aide d'une politique telle que celle présentée dans l'exemple suivant.

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

```

```

    "Effect": "Allow",
    "Action": [
        "iot>ListRetainedMessages"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:client/device1"
    ],
},
{
    "Effect": "Allow",
    "Action": [
        "iot:GetRetainedMessage"
    ],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topic/foo"
    ]
}
]
}

```

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"]
                }
            }
        }
    ]
}
```

La politique suivante permet à un appareil de publier si le certificat est associé à un objet avec un type d'objet particulier et si l'objet possède un attribut `attributeName` avec valeur `attributeValue`. Pour de plus amples informations sur les variables de politique d'objet, veuillez consulter [Variables de stratégie d'objet \(p. 364\)](#).

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

```
{
    "Effect": "Allow",
    "Action": [
        "iot:Publish"
    ],
    "Resource": "arn:aws:iot:us-east-1:123456789012:topic/device/stats",
    "Condition": {
        "StringEquals": {
            "iot:Connection.Thing.Attributes[attributeName)": "attributeValue",
            "iot:Connection.Thing.ThingTypeName": "Thing_Type_Name"
        },
        "Bool": {
            "iot:Connection.Thing.IsAttached": "true"
        }
    }
}
```

La politique suivante permet à un appareil de publier dans une rubrique qui commence par un attribut de l'objet. Si le certificat de l'appareil n'est pas associé à l'objet, cette variable ne sera pas résolue et provoquera une erreur d'accès refusé. Pour de plus amples informations sur les variables de politique d'objet, veuillez consulter [Variables de stratégie d'objet \(p. 364\)](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": "arn:aws:iot:us-east-1:123456789012:topic/
${iot:Connection.Thing.Attributes[attributeName]}/"
        }
    ]
}
```

Exemple de politique d'emploi de base

Cet exemple montre les déclarations de politique requises pour une cible de tâche qui est un appareil unique auquel il est possible de recevoir une demande de tâche et de communiquer l'état d'exécution de la tâche. AWS IoT

Remplacez ***us-west- 2:57 EXAMPLE833*** par votre Région AWS, deux points (:)) et votre Compte AWS numéro à 12 chiffres, puis remplacez-le par ***uniqueThingName*** le nom de la ressource objet dans laquelle le périphérique est représenté. AWS IoT

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:57EXAMPLE833:client/uniqueThingName"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-west-2:57EXAMPLE833:topic/+/+/+"
            ]
        }
    ]
}
```

```
        "Action": [
            "iot:Publish"
        ],
        "Resource": [
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/pubtopic",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/job/*",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/events/jobExecution/*",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Subscribe"
        ],
        "Resource": [
            "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/test/dc/subtopic",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/$aws/events/jobExecution/*",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topicfilter/$aws/things/uniqueThingName/jobs/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Receive"
        ],
        "Resource": [
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/test/dc/subtopic",
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName/jobs/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:DescribeJobExecution",
            "iot:GetPendingJobExecutions",
            "iot:StartNextPendingJobExecution",
            "iot:UpdateJobExecution"
        ],
        "Resource": [
            "arn:aws:iot:us-west-2:57EXAMPLE833:topic/$aws/things/uniqueThingName"
        ]
    }
}
```

Autorisation avec les identités Amazon Cognito

Il existe deux types d'identités Amazon Cognito : les identités non authentifiées et les identités authentifiées. Si votre application prend en charge les identités Amazon Cognito non authentifiées, aucune authentification n'est effectuée. Vous ne savez donc pas qui est l'utilisateur.

Identités non authentifiées : pour les identités Amazon Cognito non authentifiées, vous accordez des autorisations en attachant un rôle IAM à un pool d'identités non authentifié. Nous vous recommandons de n'accorder l'accès aux ressources que vous souhaitez mettre à la disposition des utilisateurs inconnus.

Important

Pour les utilisateurs non authentifiés d'Amazon Cognito qui se connectent àAWS IoT Core, nous vous recommandons de donner accès à des ressources très limitées dans les politiques IAM.

Identités authentifiées : pour les identités Amazon Cognito authentifiées, vous devez spécifier les autorisations à deux endroits :

- Joignez une politique IAM au pool d'identités Amazon Cognito authentifié et
- Joignez une AWS IoT Core politique à l'identité Amazon Cognito (utilisateur authentifié).

Exemples de politiques pour les utilisateurs non authentifiés et authentifiés d'Amazon Cognito qui se connectent à AWS IoT Core

L'exemple suivant montre les autorisations dans la politique IAM et dans la politique IoT d'une identité Amazon Cognito. L'utilisateur authentifié souhaite publier sur une rubrique spécifique à l'appareil (par exemple, device/DEVICE_ID/status).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/Client_ID"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/device/Device_ID/status"  
            ]  
        }  
    ]  
}
```

L'exemple suivant montre les autorisations d'un rôle non authentifié Amazon Cognito dans une politique IAM. L'utilisateur non authentifié souhaite publier sur des sujets non spécifiques à l'appareil qui ne nécessitent pas d'authentification.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/non_device_specific_topic"  
            ]  
        }  
    ]  
}
```

```
    ]  
}
```

GitHubexemples

Les exemples d'applications Web suivants GitHub montrent comment intégrer l'attachement à une politique pour les utilisateurs authentifiés dans le processus d'inscription et d'authentification des utilisateurs.

- [Application Web MQTT pour publier/souscrire à React à l'aide du AWS AmplifyKit SDK des appareils AWS IoT pour JavaScript](#)
- [Application Web React de publication/d'abonnement MQTT utilisant AWS AmplifyKit SDK des appareils AWS IoT pour JavaScript, la et une fonction Lambda](#)

Amplify est un ensemble d'outils et de services qui vous aide à créer des applications Web et mobiles qui s'intègrent aux AWS services. Pour plus d'informations sur Amplify, consultez [Documentation du framework Amplify](#).

Les deux exemples exécutent les étapes suivantes.

1. Lorsqu'un utilisateur crée un compte, l'application crée un groupe d'utilisateurs et une identité Amazon Cognito.
2. Lorsqu'un utilisateur s'authentifie, l'application crée et associe une politique à l'identité. Cela donne à l'utilisateur des autorisations de publication et d'abonnement.
3. L'utilisateur peut utiliser l'application pour publier des rubriques MQTT et s'y abonner.

Le premier exemple utilise l'opération `AttachPolicy` d'API directement dans l'opération d'authentification. L'exemple suivant montre comment implémenter cet appel d'API dans une application Web React qui utilise Amplify et leKit SDK des appareils AWS IoT pour JavaScript.

```
function attachPolicy(id, policyName) {  
    var Iot = new AWS.Iot({region: AWSConfiguration.region, apiVersion:  
        AWSConfiguration.apiVersion, endpoint: AWSConfiguration.endpoint});  
    var params = {policyName: policyName, target: id};  
  
    console.log("Attach IoT Policy: " + policyName + " with cognito identity id: " + id);  
    Iot.attachPolicy(params, function(err, data) {  
        if (err) {  
            if (err.code !== 'ResourceAlreadyExistsException') {  
                console.log(err);  
            }  
        }  
        else {  
            console.log("Successfully attached policy with the identity", data);  
        }  
    });  
}
```

Ce code apparaît dans le fichier [AuthDisplay.js](#).

Le deuxième exemple implémente l'opération `AttachPolicy` d'API dans une fonction Lambda. L'exemple suivant montre comment Lambda utilise cet appel d'API.

```
iot.attachPolicy(params, function(err, data) {  
    if (err) {
```

```
        if (err.code !== 'ResourceAlreadyExistsException') {
            console.log(err);
            res.json({error: err, url: req.url, body: req.body});
        }
    } else {
        console.log(data);
        res.json({success: 'Create and attach policy call succeed!', url: req.url, body: req.body});
    }
});
```

Ce code apparaît à l'intérieur de la `iot.GetPolicy` fonction dans le fichier [app.js](#).

Note

Lorsque vous appelez la fonction avec des AWS informations d'identification que vous obtenez via les pools d'identités Amazon Cognito, l'objet de contexte de votre fonction Lambda contient une valeur pour `context.cognito_identity_id`. Pour plus d'informations, consultez les rubriques suivantes.

- [AWS LambdaObjet de contexte dans Node.js](#)
- [AWS LambdaObjet de contexte en Python](#)
- [AWS LambdaObjet de contexte dans Ruby](#)
- [AWS LambdaObjet de contexte en Java](#)
- [AWS LambdaObjet de contexte dans Go](#)
- [AWS LambdaObjet de contexte en C#](#)
- [AWS Lambdaobjet de contexte dans PowerShell](#)

Autorisation d'appels directs vers des AWS services à l'aide du fournisseur AWS IoT Core d'informations d'identification

Les appareils peuvent utiliser des certificats X.509 pour se connecter à AWS IoT Core en utilisant les protocoles d'authentification mutuelle TLS. Les autres services ne prennent pas en charge l'authentification basée sur des certificats, mais ils peuvent être appelés à l'aide AWS d'informations d'identification au format [AWSSignature Version 4](#). L'[algorithme Signature Version 4](#) exige normalement que l'appelant dispose 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 le [certificat X.509](#) intégré comme identité unique de l'appareil pour authentifier les demandes AWS. Ainsi, vous n'avez plus besoin de stocker un ID de clé d'accès et une clé d'accès secrète sur votre appareil.

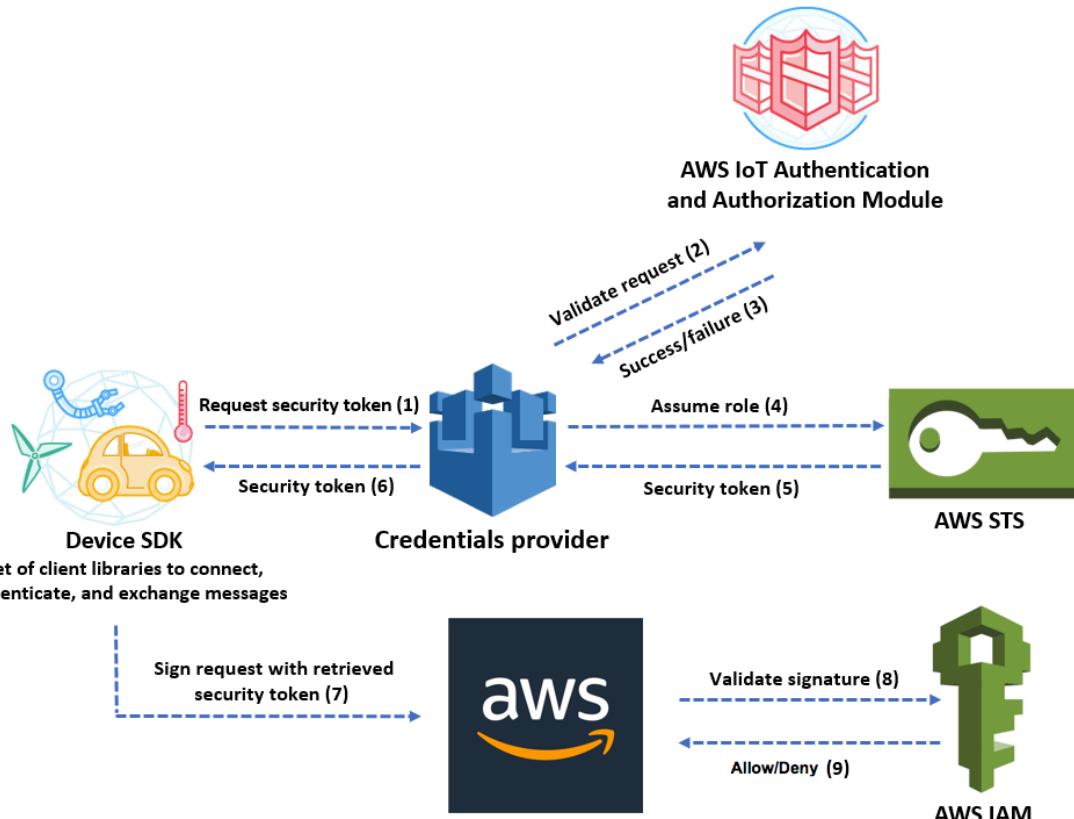
Le fournisseur d'informations d'identification authentifie un mandataire en utilisant un certificat X.509 et émet un jeton de sécurité temporaire à priviléges limités. Le jeton peut être utilisé pour signer et authentifier n'importe quelle AWS demande. Pour authentifier vos AWS demandes, vous devez créer et configurer un rôle [AWS Identity and Access Management \(IAM\)](#) et y associer des politiques IAM appropriées afin que le fournisseur d'informations d'identification puisse assumer le rôle en votre nom. Pour plus d'informations sur AWS IoT Core et IAM, consultez [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#).

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 :

- Ils endpointAddress sont revenus paraws `iot describe-endpoint --endpoint-type iot:CredentialProvider`.

Les connexions tentées par les appareils sans la host_name valeur correcte échoueront.

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. L'appareil utilise le jeton de sécurité pour signer une AWS demande avec AWS Signature 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 est réussie. 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 chaque AWS service que vous souhaitez appeler, attachez une politique 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 de l'objet dans sa demande à un AWS service, le fournisseur d'informations d'identification ajoute credentials-iot:ThingName et credentials-iot:ThingTypeName en tant que variables contextuelles 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 accorde à la fois iam:GetRole des iam:PassRole autorisations et des autorisations à un AWS utilisateur. L'autorisation iam:GetRole autorise l'utilisateur à obtenir des informations sur le rôle que vous venez de créer. L'iam:PassRole autorisation permet à l'utilisateur de transmettre le rôle à un autre AWS service.

```
{  
    "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 aux AWS services doit savoir quel rôle ARN utiliser lors de la connexion 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.

credentialDurationSeconds

Facultatif. Durée de validité (en secondes) des informations d'identification. La valeur minimale est de 900 secondes (15 minutes). La valeur maximale est de 43 200 secondes (12 heures). La valeur par défaut est de 3 600 secondes (1 heure).

Note

Le fournisseur AWS IoT Core d'identifiants peut émettre un identifiant dont la durée de vie maximale est de 43 200 secondes (12 heures). Le fait que les informations d'identification soient valides jusqu'à 12 heures peut contribuer à réduire le nombre d'appels au fournisseur d'informations d'identification en les mettant en cache plus longtemps.

Cette `credentialDurationSeconds` valeur doit être inférieure ou égale à la durée de session maximale du rôle IAM auquel l'alias de rôle fait référence.

Pour plus d'informations sur cette API, consultez [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_rolealias"  
        }  
    ]  
}
```

5. Adressez une demande HTTPS au fournisseur d'informations d'identification pour obtenir un jeton de sécurité. Fournissez les informations suivantes :

- Certificate : s'agissant d'une demande HTTP avec 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 créé pour le fournisseur d'informations d'identification.
- ThingName: le nom de l'objet que vous avez créé lors de l'enregistrement de votre AWS IoT Core 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

Le nom ThingName que vous fournissez x-amzn-iot-thingname doit correspondre au nom de la AWS IoT ressource Objet affectée à un certificat. S'il ne correspond pas, une erreur 403 est renvoyée.

Exécutez la commande suivante dans le AWS CLI pour obtenir le point de terminaison du fournisseur d'informations d'identification pour votreCompte AWS. Pour plus d'informations 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 les sessionToken valeurs accessKeyIdsecretAccessKey, et pour signer les demandes adressées aux AWS services. Pour une démonstration complète, consultez [Comment éliminer le besoin d'informations d'AWS identification codées en dur sur les appareils en utilisant le billet de blog du fournisseur AWS IoT d'informations d'identification](#) sur le blog de sécurité AWS.

Accès multicompte avec IAM

AWS IoT Core vous permet de permettre à un principal de publier ou de s'abonner à un sujet qui est défini dans un sujet qui Compte AWS ne lui appartient pas. Vous configurez l'accès entre comptes en créant une politique IAM et un rôle IAM, puis en attachant la politique au rôle.

Commencez par créer une politique IAM gérée par le client comme décrit dans la section [Création de politiques IAM](#), comme vous le feriez pour les autres utilisateurs et les certificats de votre Compte AWS.

Pour les appareils enregistrés dans le AWS IoT Core registre, la politique suivante autorise les appareils auxquels ils se connectent à AWS IoT Core l'aide d'un ID client correspondant au nom de l'objet de l'appareil et à publier my/topic/*thing-name* là où le nom de l'objet est le nom de l'objet 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 de la section [Création d'un rôle pour déléguer des autorisations à un utilisateur IAM](#). Entrez l'ID de compte 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'ID de AWS compte auquel vous accordez l'accès, vous pouvez utiliser le format de politique de confiance suivant pour ce faire :

```
{  
    "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

Le [modèle de responsabilité partagée](#) AWS s'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 AWS Cloud s'exécute. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Ce contenu comprend les tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour en savoir plus sur la confidentialité des données, consultez [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée [AWSet RGPD \(Règlement général sur la protection des données\)](#) sur le AWSBlog de sécurité.

À des fins de protection des données, nous vous recommandons de protéger les informations d'identification Compte AWS et de configurer les comptes utilisateurs individuels avec AWS IAM Identity Center (successor to AWS Single Sign-On) ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- Utilisez les certificats SSL/TLS pour communiquer avec les ressources AWS. Nous exigeons TLS 1.2 et nous recommandons TLS 1.3.
- Configurez une API (Interface de programmation) et le journal de l'activité des utilisateurs 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 qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés FIPS (Federal Information Processing Standard) 140-2 lorsque vous accédez à AWS via une CLI (Interface de ligne de commande) ou une API (Interface de programmation), utilisez un point de terminaison FIPS (Federal Information Processing Standard). Pour en savoir plus sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela est également valable lorsque vous utilisez AWS IoT ou d'autres Services AWS à l'aide de la console, de l'API, d'AWS CLI ou des kits SDK AWS. Toutes les données que vous saisissez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

Pour en savoir plus sur la protection des données, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD](#) sur le Blog sur la sécurité d'AWS.

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. Vous êtes responsable de la protection de ces données au repos. Lorsque votre appareil envoie des données à AWS IoT, il le fait via une connexion TLS, comme indiqué plus loin dans cette section. AWS IoT les appareils peuvent envoyer des données à n'importe quel AWS service. Pour de plus amples informations sur la sécurité des données de chaque service, veuillez consulter la documentation de ce service. AWS IoT peut être configuré pour écrire des journaux dans CloudWatch Logs et consigner les appels d'AWS IoT API dans AWS CloudTrail. Pour plus d'informations sur la sécurité des données de ces services, consultez [Authentification et contrôle d'accès pour Amazon CloudWatch](#) et [Chiffrement des fichiers CloudTrail journaux à l'aide de clés gérées par AWS KMS](#).

Chiffrement des données dans AWS IoT

Par défaut, toutes les AWS IoT données en transit et au repos sont chiffrées. [Les données en transit sont cryptées à l'aide du protocole TLS \(p. 409\)](#), tandis que les données au repos sont cryptées à l'aide de clés AWS détenues. AWS IoT ne prend pas actuellement en charge les clés gérées par le client AWS KMS keys (clés KMS) à partir de AWS Key Management Service (AWS KMS) ; toutefois, Device Advisor et AWS IoT Wireless utilisent uniquement un et une clé détenue par AWS pour crypter les données des clients.

Sécurité du transport dans AWS IoT Core

Le TLS (Transport Layer Security) est un protocole cryptographique conçu pour une communication sécurisée sur un réseau informatique. La passerelle AWS IoT Core Device Gateway oblige les clients à crypter toutes les communications en transit en utilisant le protocole TLS pour les connexions entre les appareils et la passerelle. Le protocole TLS est utilisé pour garantir la confidentialité des protocoles d'application (MQTT, HTTP et WebSocket) pris en charge par AWS IoT Core. Le support TLS est disponible dans un certain nombre de langages de programmation et de systèmes d'exploitation. Les données qu'AWS IoT contient sont cryptées par le service AWS spécifique. Pour plus d'informations sur le chiffrement des données dans d'autres services AWS, consultez la documentation relative à la sécurité du service concerné.

Table des matières

- [Protocoles TLS \(p. 409\)](#)
- [Stratégies de sécurité \(p. 410\)](#)
- [Remarques importantes concernant la sécurité des transports dans AWS IoT Core \(p. 412\)](#)
- [Sécurité du transport pour les appareils sans fil LoRa WAN \(p. 413\)](#)

Protocoles TLS

AWS IoT Core prend en charge les versions suivantes du protocole TLS :

- TLS 1.3
- TLS 1.2
- TLS 1.1 (non recommandé)
- TLS 1.0 (non recommandé)

Avec AWS IoT Core, vous pouvez configurer les paramètres TLS (pour [TLS 1.2 et TLS 1.3](#)) dans les configurations de domaine. Pour plus d'informations, veuillez consulter [???](#) (p. 137).

Stratégies de sécurité

Une politique de sécurité est une combinaison de protocoles TLS et de leurs chiffrements qui détermine quels protocoles et quels chiffrements sont pris en charge lors des négociations TLS entre un client et un serveur. Vous pouvez configurer vos appareils pour qu'ils utilisent des politiques de sécurité prédéfinies en fonction de vos besoins. Notez que cela AWS IoT Core ne prend pas en charge les politiques de sécurité personnalisées.

Vous pouvez choisir l'une des politiques de sécurité prédéfinies pour vos appareils lorsque vous les connectez à AWS IoT Core. Les noms des politiques de sécurité prédéfinies les plus récentes AWS IoT Core incluent des informations de version basées sur l'année et le mois de leur publication. La politique de sécurité prédéfinie par défaut est `IoTSecurityPolicy_TLS13_1_2_2022_10`. Pour définir une politique de sécurité, vous pouvez utiliser la AWS IoT console ou le AWS CLI. Pour plus d'informations, veuillez consulter [???](#) (p. 137).

Le tableau suivant décrit les politiques de sécurité prédéfinies les plus récentes prises AWS IoT Core en charge. Le `IotSecurityPolicy_` a été supprimé des noms de stratégie dans la ligne d'en-tête afin qu'ils correspondent.

Politique de sécurité	TLS13_1_3	TLS13_1_2	TLS12_1_2	TLS12_1_0_2016_01*	TLS12_1_0_2015_01*
Port TCP	443/8443/883443/8443/883443/8443/883443			843/883	443
Protocoles TLS					
TLS 1.0				✓	✓
TLS 1.1				✓	✓
TLS 1.2		✓	✓	✓	✓
TLS 1.3	✓	✓			
Chiffrements TLS					
TLS_AES_128_GCM_SHA256	✓				
TLS_AES_256_GCM_SHA384	✓				
TLS_CHACHA20_POLY1305_SHA256					
ECDHE-RSA-AES128-GCM-SHA256		✓	✓	✓	✓
ECDHE-RSA-		✓	✓	✓	✓

Politique de sécurité	TLS13_1_3	TLS13_1_2	TLS12_1_2	TLS12_1_0_2016_01*		TLS12_1_0_2015_01*	
AES128-SHA256							
ECDHE-RSA-AES128-SHA		✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-GCM-SHA384		✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-SHA384		✓	✓	✓	✓	✓	✓
ECDHE-RSA-AES256-SHA		✓	✓	✓	✓	✓	✓
AES128-GCM-SHA256		✓	✓	✓	✓	✓	✓
AES128-SHA256		✓	✓	✓		✓	✓
AES128-SHA		✓	✓	✓	✓	✓	✓
AES256-GCM-SHA384		✓	✓	✓	✓	✓	✓
AES256-SHA256		✓	✓	✓	✓	✓	✓
AES256-SHA		✓	✓	✓	✓	✓	✓
DHE-RSA-AES256-SHA						✓	✓
ECDHE-ECDSA-AES128-GCM-SHA256		✓	✓	✓	✓	✓	✓

Politique de sécurité	TLS13_1_3	TLS13_1_2	TLS12_1_2	TLS12_1_0_2016_01*		TLS12_1_0_2015_01*	
ECDHE-ECDSA-AES128-SHA256		✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES128-SHA		✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES256-GCM-SHA384		✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES256-SHA384		✓	✓	✓	✓	✓	✓
ECDHE-ECDSA-AES256-SHA		✓	✓	✓	✓	✓	✓

Note

TLS12_1_0_2016_01 est uniquement disponible dans les éléments suivants Régions AWS : ap-east-1, ap-northeast-1, ap-southeast-2, ca-central-1, cn-south-1, cn-northeast-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, sa-south-1, sa-east-1, us-east-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, sa-east-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, sa-east-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-west-1, eu-w-us-gov-west-us-gov-west

TLS12_1_0_2015_01 est uniquement disponible dans les pays suivants Régions AWS : ap-northeast-1, ap-southeast-1, us-east-1, us-west-1, us-west-1, us-west-2, us-west-2.

Remarques importantes concernant la sécurité des transports dans AWS IoT Core

Pour les appareils qui se connectent AWS IoT Core via [MQTT](#), le protocole TLS chiffre la connexion entre les appareils et le broker et AWS IoT Core utilise l'authentification client TLS pour identifier les appareils. Pour de plus amples informations, veuillez consulter [Authentification client](#). Pour les appareils qui se connectent AWS IoT Core via [HTTP](#), le protocole TLS chiffre la connexion entre les appareils et le broker, et l'authentification est déléguée à AWS Signature Version 4. Pour de plus amples informations, veuillez consulter [Signature des demandes avec Signature Version 4](#) dans la Référence AWS générale.

AWS IoT Core oblige les appareils à envoyer l'[extension SNI \(Server Name Indication\)](#) au protocole TLS et à fournir l'adresse complète du point de terminaison sur le host_name terrain. Le host_name champ doit contenir le point de terminaison que vous appelez. Ce point de terminaison doit être l'un des suivants :

- L'adresse endpointAddress renvoyée par aws iot [describe-endpoint](#) --endpoint-type iot:Data-ATS

- L'adresse domainName renvoyée par aws iot [describe-domain-configuration](#) --domain-configuration-name "*domain_configuration_name*"

Les connexions tentées par les appareils sans la host_name valeur correcte échoueront. AWS IoT Core enregistrera les échecs CloudWatch pour le type d'[authentification personnalisé](#).

AWS IoT Core ne prend pas en charge l'[extension SessionTicket TLS](#).

Sécurité du transport pour les appareils sans fil LoRa WAN

Les appareils WAN suivent les pratiques de sécurité décrites dans [LoRaWAN™ SECURITY : A White Paper for the LoRa Alliance™](#) par Gemalto, Actility et Semtech.

Pour plus d'informations sur la sécurité du transport avec les périphériques LoRa WAN, consultez [Sécurité des données avec AWS IoT for LoRa WAN \(p. 1398\)](#).

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 AWS IoT sont envoyées via une connexion TLS à l'aide de MQTT, HTTPS et de WebSocket protocoles, ce qui les rend sécurisées par défaut pendant le transit. AWS IoT les appareils collectent des données puis les envoient à d'autres AWS services pour un 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 fait abstraction du stockage des clés, de l'accès aux objets cryptographiques et de la gestion des sessions. Il vous incombe d'utiliser cette bibliothèque pour chiffrer les données au repos sur vos appareils. Pour plus d'informations, consultez la [bibliothèque 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 cryptées à l'aide de la version 4 de Signature. Pour plus d'informations sur la façon dont les demandes AWS d'API sont signées, consultez la section [Signature des demandes AWS d'API](#). Toutes les données envoyées depuis vos appareils de test vers votre point de terminaison de test Device Advisor sont envoyées via une connexion TLS. Elles sont donc sécurisées par défaut en 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 commande [create-keys-and-certificate](#) CLI, le certificat et les clés sont renvoyés par la commande CLI. Vous êtes responsable de copier le certificat et la clé privée sur votre appareil et de les conserver en lieu sûr.

AWS IoT ne prend pas actuellement en charge les clés gérées par le client AWS KMS keys (clés KMS) à partir de AWS Key Management Service (AWS KMS) ; toutefois, Device Advisor et AWS IoT Wireless utilisent uniquement une clé détenue par AWS pour crypter les données des clients.

Device Advisor

Toutes les données envoyées à Device Advisor lors de l'utilisation AWS des API sont cryptées au repos. Device Advisor chiffre toutes vos données au repos à l'aide de clés KMS stockées et gérées dans [AWS Key Management Service](#). Device Advisor crypte vos données à l'aide de Clés détenues par AWS de. Pour plus d'informations sur Clés détenues par AWS, consultez [Clés détenues par AWS](#).

Gestion des identités et des accès pour AWS IoT

AWS Identity and Access Management (IAM) est un Service AWS qui aide un administrateur à contrôler en toute sécurité l'accès aux ressources AWS. Des administrateurs IAM contrôlent les personnes qui s'authentifient (sont connectées) et sont autorisées (disposent d'autorisations) à utiliser des ressources AWS IoT. IAM est un Service AWS que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- [Public ciblé \(p. 414\)](#)
- [Authentification avec des identités IAM \(p. 414\)](#)
- [Gestion des accès à l'aide de politiques \(p. 417\)](#)
- [Fonctionnement de AWS IoT avec IAM \(p. 419\)](#)
- [Exemples de politiques basées sur l'identité AWS IoT \(p. 438\)](#)
- [Politiques AWS gérées pour AWS IoT \(p. 441\)](#)
- [Résolution des problèmes d'identité et d'accès avec AWS IoT \(p. 451\)](#)

Public ciblé

Votre utilisation d'AWS Identity and Access Management (IAM) diffère selon la tâche que vous accomplissez dans AWS IoT.

Utilisateur du service – Si vous utilisez le service AWS IoT pour effectuer votre tâche, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Plus vous utiliserez de fonctions AWS IoT pour effectuer votre travail, plus vous pourrez avoir besoin d'autorisations supplémentaires. Si vous comprenez la gestion des accès, vous pourrez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité dans AWS IoT, consultez [Résolution des problèmes d'identité et d'accès avec AWS IoT \(p. 451\)](#).

Administrateur du service – Si vous êtes le responsable des ressources AWS IoT de votre entreprise, vous bénéficiez probablement d'un accès total à AWS IoT. Votre responsabilité est de déterminer AWS IoT les fonctionnalités ainsi que les ressources auxquelles les utilisateurs de votre service doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la façon dont votre entreprise peut utiliser IAM avec AWS IoT, veuillez consulter [Fonctionnement de AWS IoT avec IAM \(p. 419\)](#).

Administrateur IAM – Si vous êtes un administrateur IAM, vous souhaiterez peut-être en savoir plus sur la façon d'écrire des politiques pour gérer l'accès à AWS IoT. Pour voir des exemples de politiques AWS IoT basées sur l'identité que vous pouvez utiliser dans IAM, veuillez consulter [Exemples de politiques basées sur l'identité AWS IoT \(p. 438\)](#).

Authentification avec des identités IAM

Dans les AWS IoT identités, il peut s'agir de certificats d'appareils (X.509), d'identités Amazon Cognito ou d'utilisateurs ou de 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 \(p. 320\)](#).

L'authentification correspond au processus par lequel vous vous connectez à AWS avec vos informations d'identification. Vous devez vous authentifier (être connecté à AWS) en tant qu'Utilisateur racine d'un compte AWS, en tant qu'utilisateur IAM ou en endossant un rôle IAM.

Vous pouvez vous connecter à AWS en tant qu'identité fédérée à l'aide des informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center (successor to AWS Single Sign-On) Les utilisateurs (IAM Identity Center), l'authentification de connexion unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. Lorsque vous accédez à AWS en utilisant la fédération, vous endossez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter à la AWS Management Console ou au portail d'accès AWS. Pour plus d'informations sur la connexion à AWS, consultez [Connexion à votre Compte AWS](#) dans le Guide de l'utilisateur Connexion à AWS.

Si vous accédez à AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos requêtes à l'aide de vos informations d'identification. Si vous n'utilisez pas les outils AWS, vous devez signer les requêtes vous-même. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vos demandes vous-même, veuillez consulter la rubrique [Processus de signature Signature Version 4](#) dans la Références générales AWS.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, AWS vous recommande d'utiliser l'authentification multifactorielle (MFA) pour améliorer la sécurité de votre compte. Pour en savoir plus, veuillez consulter [Multi-factor authentication](#) (Authentification multifactorielle) dans le Guide de l'utilisateur AWS IAM Identity Center (successor to AWS Single Sign-On) et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

Utilisateur root Compte AWS

Lorsque vous créez un Compte AWS, vous commencez avec une seule identité de connexion disposant d'un accès complet à tous les Services AWS et ressources du compte. Cette identité est appelée utilisateur root du Compte AWS. Vous pouvez y accéder en vous connectant à l'aide de l'adresse électronique et du mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur root pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur root et utilisez-les pour effectuer les tâches que seul l'utilisateur root peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur root, consultez [Tasks that require root user credentials](#) (Tâches nécessitant des informations d'identification d'utilisateur root) dans le Guide de référence d'AWS Account Management.

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. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour

plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez 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 la AWS Management Console en [changeant 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 plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, veuillez consulter la rubrique [Jeux d'autorisations](#) dans le Guide de l'utilisateur AWS IAM Identity Center (successor to AWS Single Sign-On).
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, certains Services AWS vous permettent d'attacher une politique 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 politiques basées sur les ressources pour l'accès intercompte, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.
- Accès interservices : certains Services AWS utilisent des fonctions dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, une fonction de service ou un rôle lié au service.
 - Autorisations de principal : lorsque vous utilisez un utilisateur ou un rôle IAM afin d'effectuer des actions dans AWS, vous êtes considéré comme le principal. Les politiques accordent des autorisations au principal. 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 des actions dépendantes supplémentaires dans une politique, consultez [Actions, ressources et clés de condition pour AWS IoT](#) dans la Référence 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. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
 - Rôle lié au service : un rôle lié au service est un type de fonction du service lié à un Service AWS. Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service

s'affichent dans votre Compte AWS et sont détenus par le service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications s'exécutant 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 effectuant des demandes d'API AWS CLI ou AWS. 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 plus d'informations, consultez [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, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez les accès dans AWS en créant des politiques et en les attachant à des identités AWS ou à des ressources. Une politique est un objet dans AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit les autorisations de ces dernières. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur racine ou séance de rôle) envoie une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées dans AWS en tant que documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Présentation des politiques JSON](#) dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur avec cette politique peut obtenir des informations utilisateur à partir de la AWS Management Console, de la AWS CLI ou de l'API AWS.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez attacher à plusieurs utilisateurs, groupes et rôles dans votre Compte AWS. Les politiques gérées incluent les politiques gérées par AWS et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance

de rôle IAM et des politiques de compartiment Amazon S3. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez [spécifier un principal](#) dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou des Services AWS.

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques gérées AWS depuis IAM dans une politique basée sur une ressource.

Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3, AWS WAF et Amazon VPC sont des exemples de services prenant en charge les ACL. Pour en savoir plus sur les listes de contrôle d'accès, consultez [Présentation des listes de contrôle d'accès \(ACL\)](#) dans le Guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courantes. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d'autorisations : une limite d'autorisations est une fonction avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations qui en résultent représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ Principal ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.
- Politiques de contrôle des services (SCP) - les SCP sont des politiques 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 politiques de contrôle de service (SCP) à l'un ou à l'ensemble de vos comptes. La politique de contrôle des services limite les autorisations pour les entités dans les comptes membres, y compris dans chaque Utilisateur racine d'un compte AWS. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous passez en tant que paramètre lorsque vous programmez afin de créer une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de la séance obtenue sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations, consultez [Politiques de séance](#) dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques 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 politiques, consultez [Logique d'évaluation de politiques](#) dans le Guide de l'utilisateur IAM.

Fonctionnement de 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 avec AWS IoT Pour obtenir une vue d'ensemble de la façon dont AWS IoT et d'autres services AWS fonctionnent avec IAM, veuillez consulter [Services AWS qui fonctionnent avec IAM](#)dans le Guide de l'utilisateur IAM.

Rubriques

- [AWS IoTPolitiques basées sur l'identité \(p. 419\)](#)
- [AWS IoTPolitiques basées sur les ressources \(p. 437\)](#)
- [Autorisation basée sur les balises AWS IoT \(p. 437\)](#)
- [Rôles IAM AWS IoT \(p. 438\)](#)

AWS IoTPolitiques basées sur l'identité

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 IoT prend en charge des actions, ressources et clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une politique JSON, veuillez consulter [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

Actions

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément Action d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de politique 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 politique. 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, l'AWS IoTAPI associée et la ressource manipulée par l'action.

Actions de politique	API AWS IoT	Ressources
IoT : AcceptCertificateTransfer	AcceptCertificateTransfer	<code>arn:aws:iot:<region>:<account-id>:cert/<cert-id></code> Note Le compte Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.
IoT : AddThingToThingGroup	AddThingToThingGroup	<code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> <code>arn:aws:iot:<region>:<account-id>:thing/<thing-name></code>
IoT : AssociateTargetsWithJob	AssociateTargetsWithJob	

Actions de politique	API AWS IoT	Ressources
IoT : AttachPolicy	AttachPolicy	<p>arn:aws:iot:<i>region:account-id:thinggroup/thing-group-name</i></p> <p>or</p> <p>arn:aws:iot:<i>region:account-id:cert/cert-id</i></p>
IoT : AttachPrincipalPolicy	AttachPrincipalPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : AttachSecurityProfile	AttachSecurityProfile	<p>arn:aws:iot:<i>region:account-id:securityprofile/security-profile-name</i></p> <p>arn:aws:iot:<i>region:account-id:dimension/dimension-name</i></p>
IoT : AttachThingPrincipal	AttachThingPrincipal	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : CancelCertificateTransfer	CancelCertificateTransfer	<p>arn:aws:iot:<i>region:account-id:cert/cert-id</i></p> <p>Note</p> <p>Le compte Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.</p>
IoT : CancelJob	CancelJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT : CancelJobExecution	CancelJobExecution	<p>arn:aws:iot:<i>region:account-id:job/job-id</i></p> <p>arn:aws:iot:<i>region:account-id:thing/thing-name</i></p>
IoT : ClearDefaultAuthorizer	ClearDefaultAuthorizer	Aucune
IoT : CreateAuthorizer	CreateAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i>
IoT : CreateCertificateFromCsr	CreateCertificateFromCsr	
IoT : CreateDimension	CreateDimension	arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
IoT : CreateJob	CreateJob	<p>arn:aws:iot:<i>region:account-id:job/job-id</i></p> <p>arn:aws:iot:<i>region:account-id:thinggroup/thing-group-name</i></p> <p>arn:aws:iot:<i>region:account-id:thing/thing-name</i></p> <p>arn:aws:iot:<i>region:account-id:jobtemplate/job-template-id</i></p>
IoT : CreateJobTemplate	CreateJobTemplate	<p>arn:aws:iot:<i>region:account-id:job/job-id</i></p> <p>arn:aws:iot:<i>region:account-id:jobtemplate/job-template-id</i></p>

Actions de politique	API AWS IoT	Ressources
IoT : CreateKeysAndCertificate	CreateKeysAndCertificate	
IoT : CreatePolicy	CreatePolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : CreatePolicyVersion	CreatePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i> Note Il doit s'agir d'une AWS IoT politique et non d'une politique IAM.
IoT : CreateRoleAlias	CreateRoleAlias	(paramètre : roleAlias) arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
IoT : CreateSecurityProfile	CreateSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
IoT : CreateThing	CreateThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : CreateThingGroup	CreateThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> pour le groupe en cours de création et pour le groupe parent, si utilisé
IoT : CreateThingType	CreateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
IoT : CreateTopicRule	CreateTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : DeleteAuthorizer	DeleteAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-name</i>
iot:DeleteCACertificate	DeleteCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
IoT : DeleteCertificate	DeleteCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DeleteDimension	DeleteDimension	arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
IoT : DeleteJob	DeleteJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT : DeleteJobTemplate	DeleteJobTemplate	arn:aws:iot: <i>region:account-id:job/job-template-id</i>
IoT : DeleteJobExecution	DeleteJobExecution	arn:aws:iot: <i>region:account-id:job/job-id</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>

Actions de politique	API AWS IoT	Ressources
IoT : DeletePolicy	DeletePolicy	arn:aws:iot: <i>region:account-id</i> :policy/ <i>policy-name</i>
IoT : DeletePolicyVersion	DeletePolicyVersion	arn:aws:iot: <i>region:account-id</i> :policy/ <i>policy-name</i>
IoT : DeleteRegistrationCode	DeleteRegistrationCode	
IoT : DeleteRoleAlias	DeleteRoleAlias	arn:aws:iot: <i>region:account-id</i> :rolealias/ <i>role-alias-name</i>
IoT : DeleteSecurityProfile	DeleteSecurityProfile	arn:aws:iot: <i>region:account-id</i> :securityprofile/ <i>security-profile-name</i> arn:aws:iot: <i>region:account-id</i> :dimension/ <i>dimension-name</i>
IoT : DeleteThing	DeleteThing	arn:aws:iot: <i>region:account-id</i> :thing/ <i>thing-name</i>
IoT : DeleteThingGroup	DeleteThingGroup	arn:aws:iot: <i>region:account-id</i> :thinggroup/ <i>thing-group-name</i>
IoT : DeleteThingType	DeleteThingType	arn:aws:iot: <i>region:account-id</i> :thingtype/ <i>thing-type-name</i>
IoT : DeleteTopicRule	DeleteTopicRule	arn:aws:iot: <i>region:account-id</i> :rule/ <i>rule-name</i>
IoT : Supprimer la V2 LoggingLevel	Supprimer la V2 LoggingLevel	arn:aws:iot: <i>region:account-id</i> :thinggroup/ <i>thing-group-name</i>
IoT : DeprecateThingType	DeprecateThingType	arn:aws:iot: <i>region:account-id</i> :thingtype/ <i>thing-type-name</i>
IoT : DescribeAuthorizer	DescribeAuthorizer	arn:aws:iot: <i>region:account-id</i> :authorizer/ <i>authorizer-function-name</i> (paramètre : authorizerName) Aucun(e)
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot: <i>region:account-id</i> :cacert/ <i>cert-id</i>
IoT : DescribeCertificate	DescribeCertificate	arn:aws:iot: <i>region:account-id</i> :cert/ <i>cert-id</i>
IoT : DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Aucune
IoT : DescribeEndpoint	DescribeEndpoint	*
IoT : DescribeEventConfigurations	DescribeEventConfigurations	Action(s)
IoT : DescribelIndex	DescribelIndex	arn:aws:iot: <i>region:account-id</i> :index/ <i>index-name</i>

Actions de politique	API AWS IoT	Ressources
IoT : DescribeJob	DescribeJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT : DescribeJobExecution	DescribeJobExecution	Aucune
IoT : DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot: <i>region:account-id:job/job-template-id</i>
IoT : DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
IoT : DescribeThing	DescribeThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : DescribeThingGroup	DescribeThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : DescribeThingRegistrationTask	DescribeThingRegistrationTask	Aucune
IoT : DescribeThingType	DescribeThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
IoT : DetachPolicy	DetachPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i> or arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DetachSecurityProfile	DetachSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
IoT : DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DisableTopicRule	DisableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : EnableTopicRule	EnableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : GetIndexingConfiguration	GetIndexingConfiguration	Aucune
IoT : GetJobDocument	GetJobDocument	arn:aws:iot: <i>region:account-id:job/job-id</i>

Actions de politique	API AWS IoT	Ressources
IoT : GetLoggingOptions	GetLoggingOptions	*
IoT : GetPolicy	GetPolicy	<code>arn:aws:iot:<region>:<account-id>:policy/<policy-name></code>
IoT : GetPolicyVersion	GetPolicyVersion	<code>arn:aws:iot:<region>:<account-id>:policy/<policy-name></code>
IoT : GetRegistrationCode	GetRegistrationCode	*
IoT : GetTopicRule	GetTopicRule	<code>arn:aws:iot:<region>:<account-id>:rule/<rule-name></code>
IoT : ListAttachedPolicies	ListAttachedPolicies	<code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> or <code>arn:aws:iot:<region>:<account-id>:cert/<cert-id></code>
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	<code>arn:aws:iot:<region>:<account-id>:thinggroup/<thing-group-name></code> si le thingGroupName paramètre est utilisé
IoT : ListJobTemplates	ListJobs	Aucune
IoT : ListOutgoingCertificates	ListOutgoingCertificates	*
IoT : ListPolicies	ListPolicies	*
IoT : ListPolicyPrincipals	ListPolicyPrincipals	*

Actions de politique	API AWS IoT	Ressources
IoT : ListPolicyVersions	ListPolicyVersions	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : ListPrincipalThings	ListPrincipalThings	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : ListRoleAliases	ListRoleAliases	Aucune
IoT : ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : ListThingGroups	ListThingGroups	Aucune
IoT : ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : ListThingPrincipals	ListThingPrincipals	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : ListThingRegistrationTaskReports	ListThingRegistrationTaskReports	Task Reports
IoT : ListThingRegistrationTasks	ListThingRegistrationTasks	Task
IoT : ListThingTypes	ListThingTypes	*
IoT : ListThings	ListThings	*
IoT : ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : ListTopicRules	ListTopicRules	*
IoT : Liste V2 LoggingLevels	Liste V2 LoggingLevels	Aucune
iot:RegisterCACertificate*	RegisterCACertificate*	
IoT : RegisterCertificate	RegisterCertificate	*
IoT : RegisterThing	RegisterThing	Aucune
IoT : RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
		arn:aws:iot: <i>region:account-id:thing/thing-name</i>

Actions de politique	API AWS IoT	Ressources
IoT : ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot: <i>region:account-id</i> :rule/ <i>rule-name</i>
IoT : SearchIndex	SearchIndex	arn:aws:iot: <i>region:account-id</i> :index/ <i>index-id</i>
IoT : SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot: <i>region:account-id</i> :authorizer/ <i>authorizer-function-name</i>
IoT : SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot: <i>region:account-id</i> :policy/ <i>policy-name</i>
IoT : SetLoggingOptions	SetLoggingOptions	arn:aws:iot: <i>region:account-id</i> :role/ <i>role-name</i>
IoT : Set V2 LoggingLevel	Set V2 LoggingLevel	arn:aws:iot: <i>region:account-id</i> :thinggroup/ <i>thing-group-name</i>
IoT : Set V2 LoggingOptions	Set V2 LoggingOptions	arn:aws:iot: <i>region:account-id</i> :role/ <i>role-name</i>
IoT : StartThingRegistrationTask	StartThingRegistrationTask	Aucune
IoT : StopThingRegistrationTask	StopThingRegistrationTask	Aucune
IoT : TestAuthorization	TestAuthorization	arn:aws:iot: <i>region:account-id</i> :cert/ <i>cert-id</i>
IoT : TestInvokeAuthorizer	TestInvokeAuthorizer	Aucune
IoT : TransferCertificate	TransferCertificate	arn:aws:iot: <i>region:account-id</i> :cert/ <i>cert-id</i>
IoT : UpdateAuthorizer	UpdateAuthorizer	arn:aws:iot: <i>region:account-id</i> :authorizerfunction/ <i>authorizer-function-name</i>
iot:UpdateCACertificate	UpdateCACertificate	arn:aws:iot: <i>region:account-id</i> :cacert/ <i>cert-id</i>
IoT : UpdateCertificate	UpdateCertificate	arn:aws:iot: <i>region:account-id</i> :cert/ <i>cert-id</i>
IoT : UpdateDimension	UpdateDimension	arn:aws:iot: <i>region:account-id</i> :dimension/ <i>dimension-name</i>
IoT : UpdateEventConfigurations	UpdateEventConfigurations	Aucune
IoT : UpdateIndexingConfiguration	UpdateIndexingConfiguration	Aucune
IoT : UpdateRoleAlias	UpdateRoleAlias	arn:aws:iot: <i>region:account-id</i> :rolealias/ <i>rolealias-name</i>

Actions de politique	API AWS IoT	Ressources
IoT : UpdateSecurityProfile	UpdateSecurityProfile	arn:aws:iot: <i>region:account-id:securityprofile/security-profile-name</i> arn:aws:iot: <i>region:account-id:dimension/dimension-name</i>
IoT : UpdateThing	UpdateThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : UpdateThingGroup	UpdateThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : UpdateThingGroupsForThing	UpdateThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i> arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>

Les actions de politique dans AWS IoT utilisent le préfixe suivant avant l'action : iot:. Par exemple, pour accorder à une personne l'autorisation de répertorier tous les objets connectés enregistrés dans l'ListThings API, vous incluez l'iot>ListThings action dans sa stratégie. Compte AWS Les déclarations de politique 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 des actions AWS IoT, consultez [Actions définies par AWS IoT](#) dans le Guide de l'utilisateur IAM.

Actions Device Advisor

Le tableau suivant répertorie les actions IAM IoT Device Advisor, l'API AWS IoT Device Advisor associée et la ressource manipulée par l'action.

Actions de politique	API AWS IoT	Ressources
conseiller en matière d'appareils IoT : CreateSuiteDefinition	CreateSuiteDefinition	Aucune
conseiller en matière d'appareils IoT : DeleteSuiteDefinition	DeleteSuiteDefinition	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>

Actions de politique	API AWS IoT	Ressources
conseiller en matière d'appareils IoT : GetSuiteDefinition	GetSuiteDefinition	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
conseiller en matière d'appareils IoT : GetSuiteRun	GetSuiteRun	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-run-id</i>
conseiller en matière d'appareils IoT : GetSuiteRunReport	GetSuiteRunReport	arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
conseiller en matière d'appareils IoT : ListSuiteDefinitions	ListSuiteDefinitions	Aucune
conseiller en matière d'appareils IoT : ListSuiteRuns	ListSuiteRuns	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
conseiller en matière d'appareils IoT : ListTagsForResource	ListTagsForResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
conseiller en matière d'appareils IoT : StartSuiteRun	StartSuiteRun	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
conseiller en matière d'appareils IoT : TagResource	TagResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
conseiller en matière d'appareils IoT : UntagResource	UntagResource	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i> arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>
conseiller en matière d'appareils IoT : UpdateSuiteDefinition	UpdateSuiteDefinition	arn:aws:iotdeviceadvisor: <i>region:account-id:suitedefinition/suite-definition-id</i>
conseiller en matière d'appareils IoT : StopSuiteRun	StopSuiteRun	arn:aws:iotdeviceadvisor: <i>region:account-id:suiterun/suite-definition-id/suite-run-id</i>

Les actions de stratégie dans AWS IoT Device Advisor utilisent le préfixe suivant avant l'action `:iotdeviceadvisor:`. Par exemple, pour accorder à une personne l'autorisation de répertorier toutes les définitions de suite qui y sont enregistrées Compte AWS avec l'ListSuiteDefinitionsAPI, vous incluez l'`:iotdeviceadvisor:ListSuiteDefinitions`action dans sa stratégie.

Ressources

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON Resource indique le ou les objets 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 des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

```
"Resource": "*"
```

Ressources AWS IoT

Actions de politique	API AWS IoT	Ressources
IoT : AcceptCertificateTransfer	AcceptCertificateTransfer	<p><code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code></p> <p>Note</p> <p>Le compte Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.</p>
IoT : AddThingToThingGroup	AddThingToThingGroup	<p><code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code></p> <p><code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code></p>
IoT : AssociateTargetsWithJob	AssociateTargetsWithJob	
IoT : AttachPolicy	AttachPolicy	<p><code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code></p> <p>or</p> <p><code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code></p>
IoT : AttachPrincipalPolicy	AttachPrincipalPolicy	<code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code>
IoT : AttachThingPrincipal	AttachThingPrincipal	<code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code>
IoT : CancelCertificateTransfer	CancelCertificateTransfer	<p><code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code></p> <p>Note</p> <p>Le compte Compte AWS spécifié dans l'ARN doit être le compte vers lequel le certificat est transféré.</p>

Actions de politique	API AWS IoT	Ressources
IoT : CancelJob	CancelJob	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code>
IoT : CancelJobExecution	CancelJobExecution	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code> <code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>
IoT : ClearDefaultAuthorizer	ClearDefaultAuthorizer	Aucune
IoT : CreateAuthorizer	CreateAuthorizer	<code>arn:aws:iot:<i>region:account-id</i>:authorizer/<i>authorizer-function-name</i></code>
IoT : CreateCertificateFromCsr	CreateCertificateFromCsr	
IoT : CreateJob	CreateJob	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code> <code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code> <code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code> <code>arn:aws:iot:<i>region:account-id</i>:jobtemplate/<i>job-template-id</i></code>
IoT : CreateJobTemplate	CreateJobTemplate	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code> <code>arn:aws:iot:<i>region:account-id</i>:jobtemplate/<i>job-template-id</i></code>
IoT : CreateKeysAndCertificate	CreateKeysAndCertificate	
IoT : CreatePolicy	CreatePolicy	<code>arn:aws:iot:<i>region:account-id</i>:policy/<i>policy-name</i></code>
CreatePolicyVersion	IoT : CreatePolicyVersion	<code>arn:aws:iot:<i>region:account-id</i>:policy/<i>policy-name</i></code>
		Note
		Il doit s'agir d'une AWS IoT politique et non d'une politique IAM.
IoT : CreateRoleAlias	CreateRoleAlias	(paramètre : roleAlias) <code>arn:aws:iot:<i>region:account-id</i>:rolealias/<i>role-alias-name</i></code>
IoT : CreateThing	CreateThing	<code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>
IoT : CreateThingGroup	CreateThingGroup	<code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code> pour le groupe en cours de création et pour le groupe parent, si utilisé

Actions de politique	API AWS IoT	Ressources
IoT : CreateThingType	CreateThingType	<code>arn:aws:iot:<i>region:account-id</i>:thingtype/<i>thing-type-name</i></code>
IoT : CreateTopicRule	CreateTopicRule	<code>arn:aws:iot:<i>region:account-id</i>:rule/<i>rule-name</i></code>
IoT : DeleteAuthorizer	DeleteAuthorizer	<code>arn:aws:iot:<i>region:account-id</i>:authorizer/<i>authorizer-name</i></code>
iot:DeleteCACertificate	DeleteCACertificate	<code>arn:aws:iot:<i>region:account-id</i>:cacert/<i>cert-id</i></code>
IoT : DeleteCertificate	DeleteCertificate	<code>arn:aws:iot:<i>region:account-id</i>:cert/<i>cert-id</i></code>
IoT : DeleteJob	DeleteJob	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code>
IoT : DeleteJobExecution	DeleteJobExecution	<code>arn:aws:iot:<i>region:account-id</i>:job/<i>job-id</i></code> <code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>
IoT : DeleteJobTemplate	DeleteJobTemplate	<code>arn:aws:iot:<i>region:account-id</i>:jobtemplate/<i>job-template-id</i></code>
IoT : DeletePolicy	DeletePolicy	<code>arn:aws:iot:<i>region:account-id</i>:policy/<i>policy-name</i></code>
IoT : DeletePolicyVersion	DeletePolicyVersion	<code>arn:aws:iot:<i>region:account-id</i>:policy/<i>policy-name</i></code>
IoT : DeleteRegistrationCode	DeleteRegistrationCode	
IoT : DeleteRoleAlias	DeleteRoleAlias	<code>arn:aws:iot:<i>region:account-id</i>:rolealias/<i>role-alias-name</i></code>
IoT : DeleteThing	DeleteThing	<code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>
IoT : DeleteThingGroup	DeleteThingGroup	<code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code>
IoT : DeleteThingType	DeleteThingType	<code>arn:aws:iot:<i>region:account-id</i>:thingtype/<i>thing-type-name</i></code>
IoT : DeleteTopicRule	DeleteTopicRule	<code>arn:aws:iot:<i>region:account-id</i>:rule/<i>rule-name</i></code>
IoT : Supprimer la V2 LoggingLevel	Supprimer la V2 LoggingLevel	<code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code>
IoT : DeprecateThingType	DeprecateThingType	<code>arn:aws:iot:<i>region:account-id</i>:thingtype/<i>thing-type-name</i></code>
IoT : DescribeAuthorizer	DescribeAuthorizer	<code>arn:aws:iot:<i>region:account-id</i>:authorizer/<i>authorizer-function-name</i></code> (paramètre : authorizerName) Aucun(e)

Actions de politique	API AWS IoT	Ressources
iot:DescribeCACertificate	DescribeCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
IoT : DescribeCertificate	DescribeCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DescribeDefaultAuthorizer	DescribeDefaultAuthorizer	Aucune
IoT : DescribeEndpoint	DescribeEndpoint	*
IoT : DescribeEventConfigurations	DescribeEventConfigurations	Action(s)
IoT : DescribeIndex	DescribeIndex	arn:aws:iot: <i>region:account-id:index/index-name</i>
IoT : DescribeJob	DescribeJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT : DescribeJobExecution	DescribeJobExecution	Aucune
IoT : DescribeJobTemplate	DescribeJobTemplate	arn:aws:iot: <i>region:account-id:jobtemplate/job-template-id</i>
IoT : DescribeRoleAlias	DescribeRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/rolealias-name</i>
IoT : DescribeThing	DescribeThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : DescribeThingGroup	DescribeThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : DescribeThingRegistrationTask	DescribeThingRegistrationTask	Aucune
IoT : DescribeThingType	DescribeThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
IoT : DetachPolicy	DetachPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i> or arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : DetachPrincipalPolicy	DetachPrincipalPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DetachThingPrincipal	DetachThingPrincipal	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : DisableTopicRule	DisableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : EnableTopicRule	EnableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>

Actions de politique	API AWS IoT	Ressources
IoT : GetEffectivePolicies	GetEffectivePolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : GetIndexingConfiguration	GetIndexingConfiguration	Aucune
IoT : GetJobDocument	GetJobDocument	arn:aws:iot: <i>region:account-id:job/job-id</i>
IoT : GetLoggingOptions	GetLoggingOptions	*
IoT : GetPolicy	GetPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : GetPolicyVersion	GetPolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : GetRegistrationCode	GetRegistrationCode	*
IoT : GetTopicRule	GetTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : ListAttachedPolicies	ListAttachedPolicies	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> or arn:aws:iot: <i>region:account-id:cert/cert-id</i>
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: <i>region:account-id:thinggroup/thing-group-name</i> si le thingGroupName paramètre est utilisé
IoT : ListJobTemplates	ListJobTemplates	Aucune

Actions de politique	API AWS IoT	Ressources
IoT : ListOutgoingCertificates	ListOutgoingCertificates	
IoT : ListPolicies	ListPolicies	*
IoT : ListPolicyPrincipals	ListPolicyPrincipals	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : ListPolicyVersions	ListPolicyVersions	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : ListPrincipalPolicies	ListPrincipalPolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : ListPrincipalThings	ListPrincipalThings	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : ListRoleAliases	ListRoleAliases	Aucune
IoT : ListTargetsForPolicy	ListTargetsForPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : ListThingGroups	ListThingGroups	Aucune
IoT : ListThingGroupsForThing	ListThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : ListThingPrincipals	ListThingPrincipals	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : ListThingRegistrationTasks	ListThingRegistrationTasks	TaskReports
IoT : ListThingRegistrationTasks	ListThingRegistrationTasks	Aucune
IoT : ListThingTypes	ListThingTypes	*
IoT : ListThings	ListThings	*
IoT : ListThingsInThingGroup	ListThingsInThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
IoT : ListTopicRules	ListTopicRules	*
IoT : Liste V2 LoggingLevels	Liste V2 LoggingLevels	Aucune
iot:RegisterCACertificate*	RegisterCACertificate*	
IoT : RegisterCertificate	RegisterCertificate	*
IoT : RegisterThing	RegisterThing	Aucune

Actions de politique	API AWS IoT	Ressources
IoT : RejectCertificateTransfer	RejectCertificateTransfer	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : RemoveThingFromThingGroup	RemoveThingFromThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>
IoT : ReplaceTopicRule	ReplaceTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
IoT : SearchIndex	SearchIndex	arn:aws:iot: <i>region:account-id:index/index-id</i>
IoT : SetDefaultAuthorizer	SetDefaultAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i>
IoT : SetDefaultPolicyVersion	SetDefaultPolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
IoT : SetLoggingOptions	SetLoggingOptions	*
IoT : Set V2 LoggingLevel	Set V2 LoggingLevel	*
IoT : Set V2 LoggingOptions	Set V2 LoggingOptions	*
IoT : StartThingRegistrationTask	StartThingRegistrationTask	Aucune
IoT : StopThingRegistrationTask	StopThingRegistrationTask	Aucune
IoT : TestAuthorization	TestAuthorization	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : TestInvokeAuthorizer	TestInvokeAuthorizer	Aucune
IoT : TransferCertificate	TransferCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : UpdateAuthorizer	UpdateAuthorizer	arn:aws:iot: <i>region:account-id:authorizerfunction/authorizer-function-name</i>
iot:UpdateCACertificate	UpdateCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
IoT : UpdateCertificate	UpdateCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
IoT : UpdateEventConfigurations	UpdateEventConfigurations	Aucune
IoT : UpdateIndexingConfiguration	UpdateIndexingConfiguration	Aucune

Actions de politique	API AWS IoT	Ressources
IoT : UpdateRoleAlias	UpdateRoleAlias	<code>arn:aws:iot:<i>region:account-id</i>:rolealias/<i>role-alias-name</i></code>
IoT : UpdateThing	UpdateThing	<code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>
IoT : UpdateThingGroup	UpdateThingGroup	<code>arn:aws:iot:<i>region:account-id</i>:thinggroup/<i>thing-group-name</i></code>
IoT : UpdateThingGroupsForThing	UpdateThingGroupsForThing	<code>arn:aws:iot:<i>region:account-id</i>:thing/<i>thing-name</i></code>

Pour plus d'informations sur le format des ARN, consultez [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 ces cas-là, vous devez utiliser le caractère générique (*).

"Resource": "*"

Pour consulter la liste des types de AWS IoT ressources et de leurs ARN, veuillez consulter [Ressources définies par AWS IoT](#) dans le Guide de l'utilisateur IAM. Pour savoir grâce à quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez [Actions définies par AWS IoT](#).

Ressources Device Advisor

Pour définir des restrictions au niveau des ressources pour les politiques IAM de AWS IoT Device Advisor, utilisez les formats ARN de ressources suivants pour les définitions et les exécutions de suites.

Format ARN des ressources de définition de la suite

`arn:aws:iotdeviceadvisor:region:account-id:suitedefinition/suite-definition-id`

Format ARN des ressources d'exécution Suite

`arn:aws:iotdeviceadvisor:region:account-id:suiterun/suite-definition-id/suite-run-id`

Clés de condition

Les administrateurs peuvent utiliser les politiques JSON AWS pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément Condition (ou le bloc Condition) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément Condition est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des [opérateurs de condition](#), tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments Condition dans une instruction, ou plusieurs clés dans un seul élément Condition, AWS les évalue à l'aide d'une opération AND logique. Si vous spécifiez plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une 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 plus d'informations, consultez [Éléments d'une politique IAM : variables et identifications](#) dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques 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, veuillez consulter la rubrique [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/Composant de clé \${tag-key}	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 consulter la liste des clés de AWS IoT condition, veuillez consulter [Clés de condition AWS IoT](#) dans le Guide de l'utilisateur IAM. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez [Actions définies par AWS IoT](#).

Exemples

Pour voir des exemples de politiques AWS IoT basées sur l'identité, consultez [Exemples de politiques basées sur l'identité AWS IoT \(p. 438\)](#).

AWS IoT Politiques basées sur les ressources

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 IoT ne prend pas en charge les stratégies basées sur les ressources IAM. Elle soutient toutefois les politiques AWS IoT basées sur les ressources. Pour plus d'informations, veuillez consulter [Stratégies AWS IoT Core \(p. 357\)](#).

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 étiquettes, vous devez fournir les

informations d'étiquette dans l'[élément de condition](#) d'une politique utilisant les clés de condition `iot:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`. Pour plus d'informations, veuillez consulter [Utilisation des balises avec des stratégies IAM \(p. 311\)](#). Pour plus d'informations sur le balisage des ressources AWS IoT, consultez [Balisage de vos ressources AWS IoT \(p. 310\)](#).

Pour visualiser un exemple de politique basée sur l'identité permettant de limiter l'accès à une ressource en fonction des balises de cette ressource, consultez [Affichage des ressources en fonction des balises AWS IoT \(p. 440\)](#).

Rôles IAM AWS IoT

Un [rôle IAM](#) est 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'AWS STSAPI telles que [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 fonction permet à un service d'endosser une [fonction du service](#) en votre nom. Ce rôle autorise le service à accéder à des ressources d'autres services pour effectuer une action en votre nom. Les rôles de service s'affichent dans votre compte IAM et sont la propriété du compte. Cela signifie qu'un administrateur IAM peut modifier les autorisations associées à ce rôle. Toutefois, une telle action peut perturber le bon fonctionnement du service.

Exemples de politiques basées sur l'identité AWS IoT

Par défaut, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou modifier les ressources AWS IoT. 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 politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces stratégies aux utilisateurs ou aux groupes ayant besoin de ces autorisations.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de politique JSON, consultez [Création de politiques dans l'onglet JSON](#) dans le Guide de l'utilisateur IAM.

Rubriques

- [Bonnes pratiques en matière de politiques \(p. 439\)](#)
- [Utilisation de la console AWS IoT \(p. 439\)](#)
- [Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations \(p. 440\)](#)
- [Affichage des ressources en fonction des balises AWS IoT \(p. 440\)](#)
- [Affichage des ressources de AWS IoT Device Advisor en fonction des balises \(p. 441\)](#)

Bonnes pratiques en matière de politiques

Les stratégies basées sur l'identité 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 politiques basées sur l'identité, suivez ces instructions et recommandations :

- Démarrer avec AWS gérées et évoluez vers les autorisations de moindre privilège - Pour commencer à accorder des autorisations à vos utilisateurs et charges de travail, utilisez les politiques gérées AWS qui accordent des autorisations dans de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire encore les autorisations en définissant des Politiques gérées par le client AWS qui sont spécifiques à vos cas d'utilisation. Pour de plus amples informations, consultez [Politiques gérées AWS](#) ou [Politiques gérées AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège - Lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez [Politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès - Vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées via un Service AWS spécifique, comme AWS CloudFormation. Pour plus d'informations, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.
- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles - IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour de plus amples informations, consultez [Validation de politique IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Authentification multifactorielle (MFA) nécessaire : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root dans votre Compte AWS, activez l'authentification multifactorielle pour une sécurité renforcée. Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour de plus amples informations, consultez [Configuration de l'accès aux API protégé par MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

Utilisation de la console AWS IoT

Pour accéder à la console AWS IoT, vous devez disposer d'un ensemble minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et consulter les informations relatives aux AWS IoT ressources de votre compteCompte AWS. Si vous créez une stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette 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 plus d'informations, consultez [Ajout d'autorisations à un utilisateur](#) dans le Guide 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 politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations 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 politique 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 politique 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 politique 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 d'étiquette `Owner` correspond à la fois à `Owner` et à `owner`, car les noms de clé de condition ne sont pas sensibles à la casse. Pour plus d'informations, veuillez consulter la rubrique [Éléments de stratégie JSON IAM : Condition](#) dans le Guide de l'utilisateur IAM.

Affichage des ressources de AWS IoT Device Advisor en fonction des balises

Vous pouvez utiliser des conditions dans votre stratégie basée sur l'identité pour contrôler l'accès aux ressources AWS IoT Device Advisor en fonction des balises. L'exemple suivant montre comment créer une politique qui permet d'afficher une définition de suite particulière. Toutefois, l'autorisation est accordée uniquement si la balise de définition de la suite est `SuiteType` définie sur la valeur `MQTT`. Cette politique 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"}  
            }  
        }  
    ]  
}
```

Politiques AWS gérées pour AWS IoT

Pour ajouter des autorisations à des utilisateurs, des groupes et des rôles, il est plus facile d'utiliser des politiques gérées par AWS que d'écrire des politiques vous-même. Il faut du temps et de l'expertise pour [créer des politiques gérées par le client IAM](#) qui ne fournissent à votre équipe que les autorisations dont elle a besoin. Pour démarrer rapidement, vous pouvez utiliser nos politiques gérées par AWS. Ces politiques couvrent des cas d'utilisation courants et sont disponibles dans votre Compte AWS. Pour plus d'informations sur les politiques gérées par AWS, consultez [Politiques gérées par AWS](#) dans le Guide de l'utilisateur IAM.

Les services AWS assurent la maintenance et la mise à jour des politiques gérées AWS. Vous ne pouvez pas modifier les autorisations définies dans les politiques gérées par AWS. Les services ajoutent occasionnellement des autorisations à une politique gérée par AWS pour prendre en charge de nouvelles

fonctions. Ce type de mise à jour affecte toutes les identités (utilisateurs, groupes et rôles) auxquelles la politique est attachée. Les services sont très susceptibles de mettre à jour une politique gérée par AWS quand une nouvelle fonction est lancée ou quand de nouvelles opérations sont disponibles. Les services ne supprimant pas les autorisations d'une politique gérée par AWS, les mises à jour de politique n'interrompent vos autorisations existantes.

En outre, AWS prend en charge des politiques gérées pour des activités professionnelles couvrant plusieurs services. Par exemple, la politique `ReadOnlyAccess` gérée AWS donne accès en lecture seule à l'ensemble des services et ressources AWS. Quand un service lance une nouvelle fonction, AWS ajoute des autorisations en lecture seule pour les nouvelles opérations et ressources. Pour obtenir la liste des politiques de fonctions professionnelles et leurs descriptions, consultez la page [politiques gérées par AWS pour les fonctions de tâche](#) dans le Guide de l'utilisateur IAM.

Note

AWS IoT fonctionne à la fois avec AWS IoT les politiques IAM. Cette rubrique traite uniquement des politiques IAM, qui définissent une action de politique pour les opérations d'API du plan de contrôle et du plan de données. Voir aussi [Stratégies AWS IoT Core](#) (p. 357).

AWS Politique gérée par: AWSIoTConfigAccess

Vous pouvez attacher la politique AWSIoTConfigAccess à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à toutes les opérations AWS IoT de configuration. Cette stratégie peut affecter le traitement et le stockage des données. Pour consulter cette politique dans le AWS Management Console, voir [AWSIoTConfigAccess](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**— Récupérez AWS IoT des données et effectuez des actions de configuration de l'IoT.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:AcceptCertificateTransfer",  
                "iot:AddThingToThingGroup",  
                "iot:AssociateTargetsWithJob",  
                "iot:AttachPolicy",  
                "iot:AttachPrincipalPolicy",  
                "iot:AttachThingPrincipal",  
                "iot:CancelCertificateTransfer",  
                "iot:CancelJob",  
                "iot:CancelJobExecution",  
                "iot:ClearDefaultAuthorizer",  
                "iot>CreateAuthorizer",  
                "iot>CreateCertificateFromCsr",  
                "iot>CreateJob",  
                "iot>CreateKeysAndCertificate".  
            ]  
        }  
    ]  
}
```

```
"iot:CreateOTAUpdate",
"iot:CreatePolicy",
"iot:CreatePolicyVersion",
"iot:CreateRoleAlias",
"iot:CreateStream",
"iot:CreateThing",
"iot:CreateThingGroup",
"iot:CreateThingType",
"iot:CreateTopicRule",
"iot:DeleteAuthorizer",
"iot:DeleteCACertificate",
"iot:DeleteCertificate",
"iot:DeleteJob",
"iot:DeleteJobExecution",
"iot:DeleteOTAUpdate",
"iot:DeletePolicy",
"iot:DeletePolicyVersion",
"iot:DeleteRegistrationCode",
"iot:DeleteRoleAlias",
"iot:DeleteStream",
"iot:DeleteThing",
"iot:DeleteThingGroup",
"iot:DeleteThingType",
"iot:DeleteTopicRule",
"iot:DeleteV2LoggingLevel",
"iot:DeprecateThingType",
"iot:DescribeAuthorizer",
"iot:DescribeCACertificate",
"iot:DescribeCertificate",
"iot:DescribeDefaultAuthorizer",
"iot:DescribeEndpoint",
"iot:DescribeEventConfigurations",
"iot:DescribeIndex",
"iot:DescribeJob",
"iot:DescribeJobExecution",
"iot:DescribeRoleAlias",
"iot:DescribeStream",
"iot:DescribeThing",
"iot:DescribeThingGroup",
"iot:DescribeThingRegistrationTask",
"iot:DescribeThingType",
"iot:DetachPolicy",
"iot:DetachPrincipalPolicy",
"iot:DetachThingPrincipal",
"iot:DisableTopicRule",
"iot:EnableTopicRule",
"iot:GetEffectivePolicies",
"iot:GetIndexingConfiguration",
"iot:GetJobDocument",
"iot:GetLoggingOptions",
"iot:GetOTAUpdate",
"iot:GetPolicy",
"iot:GetPolicyVersion",
"iot:GetRegistrationCode",
"iot:GetTopicRule",
"iot:GetV2LoggingOptions",
"iot>ListAttachedPolicies",
"iot>ListAuthorizers",
"iot>ListCACertificates",
"iot>ListCertificates",
"iot>ListCertificatesByCA",
"iot>ListIndices",
"iot>ListJobExecutionsForJob",
"iot>ListJobExecutionsForThing",
"iot>ListJobs",
"iot>ListOTAUpdates",
```

```
"iot>ListOutgoingCertificates",
"iot>ListPolicies",
"iot>ListPolicyPrincipals",
"iot>ListPolicyVersions",
"iot>ListPrincipalPolicies",
"iot>ListPrincipalThings",
"iot>ListRoleAliases",
"iot>ListStreams",
"iot>ListTargetsForPolicy",
"iot>ListThingGroups",
"iot>ListThingGroupsForThing",
"iot>ListThingPrincipals",
"iot>ListThingRegistrationTaskReports",
"iot>ListThingRegistrationTasks",
"iot>ListThings",
"iot>ListThingsInThingGroup",
"iot>ListThingTypes",
"iot>ListTopicRules",
"iot>ListV2LoggingLevels",
"iot>RegisterCACertificate",
"iot>RegisterCertificate",
"iot>RegisterThing",
"iot>RejectCertificateTransfer",
"iot>RemoveThingFromThingGroup",
"iot>ReplaceTopicRule",
"iot>SearchIndex",
"iot>SetDefaultAuthorizer",
"iot>SetDefaultPolicyVersion",
"iot>SetLoggingOptions",
"iot>SetV2LoggingLevel",
"iot>SetV2LoggingOptions",
"iot>StartThingRegistrationTask",
"iot>StopThingRegistrationTask",
"iot>TestAuthorization",
"iot>TestInvokeAuthorizer",
"iot>TransferCertificate",
"iot>UpdateAuthorizer",
"iot>UpdateCACertificate",
"iot>UpdateCertificate",
"iot>UpdateEventConfigurations",
"iot>UpdateIndexingConfiguration",
"iot>UpdateRoleAlias",
"iot>UpdateStream",
"iot>UpdateThing",
"iot>UpdateThingGroup",
"iot>UpdateThingGroupsForThing",
"iot>UpdateAccountAuditConfiguration",
"iot>DescribeAccountAuditConfiguration",
"iot>DeleteAccountAuditConfiguration",
"iot>StartOnDemandAuditTask",
"iot>CancelAuditTask",
"iot>DescribeAuditTask",
"iot>ListAuditTasks",
"iot>CreateScheduledAudit",
"iot>UpdateScheduledAudit",
"iot>DeleteScheduledAudit",
"iot>DescribeScheduledAudit",
"iot>ListScheduledAudits",
"iot>ListAuditFindings",
"iot>CreateSecurityProfile",
"iot>DescribeSecurityProfile",
"iot>UpdateSecurityProfile",
"iot>DeleteSecurityProfile",
"iot>AttachSecurityProfile",
"iot>DetachSecurityProfile",
"iot>ListSecurityProfiles",
```

```
        "iot>ListSecurityProfilesForTarget",
        "iot>ListTargetsForSecurityProfile",
        "iot>ListActiveViolations",
        "iot>ListViolationEvents",
        "iot>ValidateSecurityProfileBehaviors"
    ],
    "Resource": "*"
}
]
```

AWS Politique gérée par: AWSIoTConfigReadOnlyAccess

Vous pouvez attacher la politique AWSIoTConfigReadOnlyAccess à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent un accès en lecture seule à toutes les opérations de AWS IoT configuration. Pour consulter cette politique dans le AWS Management Console, voir [AWSIoTConfigReadOnlyAccess](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**— Effectuez des opérations en lecture seule sur les actions de configuration de l'IoT.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:DescribeAuthorizer",
                "iot:DescribeCACertificate",
                "iot:DescribeCertificate",
                "iot:DescribeDefaultAuthorizer",
                "iot:DescribeEndpoint",
                "iot:DescribeEventConfigurations",
                "iot:DescribeIndex",
                "iot:DescribeJob",
                "iot:DescribeJobExecution",
                "iot:DescribeRoleAlias",
                "iot:DescribeStream",
                "iot:DescribeThing",
                "iot:DescribeThingGroup",
                "iot:DescribeThingRegistrationTask",
                "iot:DescribeThingType",
                "iot:GetEffectivePolicies",
                "iot:GetIndexingConfiguration",
                "iot:GetJobDocument",
                "iot:GetLoggingOptions",
                "iot:GetOTAUpdate",
                "iot:GetPolicy",
                "iot:GetPolicyVersion",
                "iot:GetRegistrationCode",
                "iot:GetTopicRule",
                "iot:GetV2LoggingOptions",
                "iot>ListAttachedPolicies",
                "iot>ListSecurityProfilesForTarget",
                "iot>ListTargetsForSecurityProfile",
                "iot>ListActiveViolations",
                "iot>ListViolationEvents",
                "iot>ValidateSecurityProfileBehaviors"
            ],
            "Resource": "*"
}
]
```

```
    "iot>ListAuthorizers",
    "iot>ListCACertificates",
    "iot>ListCertificates",
    "iot>ListCertificatesByCA",
    "iot>ListIndices",
    "iot>ListJobExecutionsForJob",
    "iot>ListJobExecutionsForThing",
    "iot>ListJobs",
    "iot>ListOTAUpdates",
    "iot>ListOutgoingCertificates",
    "iot>ListPolicies",
    "iot>ListPolicyPrincipals",
    "iot>ListPolicyVersions",
    "iot>ListPrincipalPolicies",
    "iot>ListPrincipalThings",
    "iot>ListRoleAliases",
    "iot>ListStreams",
    "iot>ListTargetsForPolicy",
    "iot>ListThingGroups",
    "iot>ListThingGroupsForThing",
    "iot>ListThingPrincipals",
    "iot>ListThingRegistrationTaskReports",
    "iot>ListThingRegistrationTasks",
    "iot>ListThings",
    "iot>ListThingsInThingGroup",
    "iot>ListThingTypes",
    "iot>ListTopicRules",
    "iot>ListV2LoggingLevels",
    "iot>SearchIndex",
    "iot>TestAuthorization",
    "iot>TestInvokeAuthorizer",
    "iot>DescribeAccountAuditConfiguration",
    "iot>DescribeAuditTask",
    "iot>ListAuditTasks",
    "iot>DescribeScheduledAudit",
    "iot>ListScheduledAudits",
    "iot>ListAuditFindings",
    "iot>DescribeSecurityProfile",
    "iot>ListSecurityProfiles",
    "iot>ListSecurityProfilesForTarget",
    "iot>ListTargetsForSecurityProfile",
    "iot>ListActiveViolations",
    "iot>ListViolationEvents",
    "iot>ValidateSecurityProfileBehaviors"
],
"Resource": "*"
}
]
```

AWS Politique gérée par: AWSIoTDataAccess

Vous pouvez attacher la politique AWSIoTDataAccess à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à toutes les opérations de AWS IoT données. Les opérations sur les données envoient des données via les protocoles MQTT ou HTTP. Pour consulter cette politique dans le AWS Management Console, voir [AWSIoTDataAccess](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**— Récupérez AWS IoT les données et autorisez un accès complet aux actions AWS IoT de messagerie.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect",  
                "iot:Publish",  
                "iot:Subscribe",  
                "iot:Receive",  
                "iot:GetThingShadow",  
                "iot:UpdateThingShadow",  
                "iot:DeleteThingShadow",  
                "iot>ListNamedShadowsForThing"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

AWS Politique gérée par: AWSIoTFullAccess

Vous pouvez attacher la politique `AWSIoTFullAccess` à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à toutes les opérations AWS IoT de configuration et de messagerie. Pour consulter cette politique dans le AWS Management Console, voir [AWSIoTFullAccess](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**— Récupérez AWS IoT les données et autorisez un accès complet aux actions AWS IoT de configuration et de messagerie.
- **iotjobsdata**— Récupérez les données AWS IoT Jobs et autorisez un accès complet aux opérations de l'API du plan de données AWS IoT Jobs.

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

```
        ]  
    }  
}
```

AWS Politique gérée par: AWSIoTLogging

Vous pouvez attacher la politique AWSIoTLogging à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent de créer des groupes Amazon CloudWatch Logs et de diffuser des journaux vers ces groupes. Cette stratégie est attachée à votre rôle de journalisation CloudWatch. Pour consulter cette politique dans leAWS Management Console, voir [AWSIoTLogging](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- logs— Récupère CloudWatch les journaux. Permet également de créer des groupes de CloudWatch journaux et de diffuser des journaux vers ces groupes.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs>PutLogEvents",  
                "logs>PutMetricFilter",  
                "logs>PutRetentionPolicy",  
                "logs>GetLogEvents",  
                "logs>DeleteLogStream"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

AWS Politique gérée par: AWSIoTOTAUUpdate

Vous pouvez attacher la politique AWSIoTOTAUUpdate à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à la création de AWS IoT tâches, à des tâches de signature de AWS IoT code et à la description des tâches de signature de AWS code. Pour consulter cette politique dans leAWS Management Console, voir [AWSIoTOTAUUpdate](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**— Créez des AWS IoT tâches et des tâches de signature de code.
- **signer**— Procède à la création de tâches de signature de AWS code.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "iot:CreateJob",  
            "signer:DescribeSigningJob"  
        ],  
        "Resource": "*"  
    }  
}
```

AWS Politique gérée par: AWSIoTRuleActions

Vous pouvez attacher la politique AWSIoTRuleActions à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à toutes les Service AWS actions prises en charge par les AWS IoT règles. Pour consulter cette politique dans le AWS Management Console, voir [AWSIoTRuleActions](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot**- Effectuez des actions pour publier des messages d'action sur les règles.
- **dynamodb**- Insérez un message dans une table DynamoDB ou divisez un message en plusieurs colonnes d'une table DynamoDB.
- **s3**- Stockez un objet dans un compartiment Amazon S3.
- **kinesis**- Envoyez un message à un objet de flux Amazon Kinesis.
- **firehose**- Insérez un enregistrement dans un objet de flux Kinesis Data Firehose.
- **cloudwatch**- Modifiez l'état de l'CloudWatch alarme ou envoyez les données du message à la CloudWatch métrique.
- **sns**- Effectuez une opération pour publier une notification à l'aide d'Amazon SNS. Cette opération est limitée aux rubriques du AWS IoT SNS.
- **sqs**- Insérez un message à ajouter à la file d'attente SQS.
- **es**- Envoyez un message au OpenSearch service Service.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": [  
            "dynamodb:PutItem",  
            "kinesis:PutRecord",  
            "iot:Publish",  
        ]  
    }  
}
```

```
        "s3:PutObject",
        "sns:Publish",
        "sns:SendMessage",
        "cloudwatch:SetAlarmState",
        "cloudwatch:PutMetricData",
        "es:ESHttpPut",
        "firehose:PutRecord"
    ],
    "Resource": "*"
}
```

AWS Politique gérée par: AWSIoTThingsRegistration

Vous pouvez attacher la politique AWSIoTThingsRegistration à vos identités IAM.

Cette politique accorde les autorisations d'identité associées qui permettent d'accéder à l'enregistrement groupé d'éléments à l'aide de l'StartThingRegistrationTaskAPI. Cette stratégie peut affecter le traitement et le stockage des données. Pour consulter cette politique dans leAWS Management Console, voir [AWSIoTThingsRegistration](#).

Détails de l'autorisation

Cette politique inclut les autorisations suivantes.

- **iot-** Effectuez des actions pour créer des objets et joindre des politiques et des certificats lors de l'enregistrement groupé.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:AddThingToThingGroup",
                "iot:AttachPolicy",
                "iot:AttachPrincipalPolicy",
                "iot:AttachThingPrincipal",
                "iot>CreateCertificateFromCsr",
                "iot>CreatePolicy",
                "iot>CreateThing",
                "iot:DescribeCertificate",
                "iot:DescribeThing",
                "iot:DescribeThingGroup",
                "iot:DescribeThingType",
                "iot:DetachPolicy",
                "iot:DetachThingPrincipal",
                "iot:GetPolicy",
                "iot>ListAttachedPolicies",
                "iot>ListPolicyPrincipals",
                "iot>ListPrincipalPolicies",
                "iot>ListPrincipalThings",
                "iot>ListTargetsForPolicy",
                "iot>ListThingGroupsForThing",
                "iot>ListThingPrincipals",
                "iot:RegisterCertificate",
                "iot:UpdateThing"
            ]
        }
    ]
}
```

```
        "iot:RegisterThing",
        "iot:RemoveThingFromThingGroup",
        "iot:UpdateCertificate",
        "iot:UpdateThing",
        "iot:UpdateThingGroupsForThing",
        "iot:AddThingToBillingGroup",
        "iot:DescribeBillingGroup",
        "iot:RemoveThingFromBillingGroup"
    ],
    "Resource": [
        "*"
    ]
}
]
```

Mises à jour AWS IoT vers des politiques gérées par AWS

Consultez le détail des mises à jour des politiques gérées par AWS pour AWS IoT depuis que ce service a commencé à suivre ces modifications. Pour obtenir des alertes automatiques concernant les modifications apportées à cette page, abonnez-vous au flux RSS sur la page d'historique du document AWS IoT.

Modification	Description	Date
AWSIoTFullAccess (p. 447) - mise à jour d'une politique existante	AWS IoT a ajouté de nouvelles autorisations pour permettre aux utilisateurs d'accéder aux opérations de l'API du plan de données AWS IoT Jobs à l'aide du protocole HTTP. Un nouveau préfixe de politique IAM vous fournit un contrôle d'accès plus précis pour accéder aux points de terminaison du plan de données de AWS IoT Jobs. <code>iotjobsdata:</code> Pour les opérations de l'API du plan de contrôle, vous utilisez toujours le <code>iot:</code> préfixe. Pour plus d'informations, veuillez consulter AWS IoT Core politiques pour le protocole HTTPS (p. 843) .	11 mai 2022
AWS IoT a démarré le suivi des modifications	AWS IoT a commencé à suivre les modifications pour ses politiques gérées par AWS.	11 mai 2022

Résolution des problèmes d'identité et d'accès avec AWS IoT

Utilisez les informations suivantes pour identifier et résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS IoT et IAM.

Rubriques

- [Je ne suis pas autorisé à effectuer une action dans AWS IoT \(p. 452\)](#)
- [Je ne suis pas autorisé à exécuter iam : PassRole \(p. 452\)](#)
- [Je veux autoriser des personnes extérieures à mon Compte AWS à accéder à mes ressources AWS IoT \(p. 453\)](#)

Je ne suis pas autorisé à effectuer une action dans AWS IoT

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à effectuer une action, vos stratégies doivent être mises à jour afin de vous permettre d'effectuer l'action.

L'exemple d'erreur suivant se produit lorsque l'utilisateur IAM tente d'utiliser la console pour afficher des informations détaillées concernant une ressource objet mais ne dispose pas des iot:[DescribeThing](#) autorisations nécessaires. mateojackson

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
iot:DescribeThing on resource: MyIoTThing
```

Dans ce cas, la politique qui s'applique à l'mateojacksonutilisateur doit être mise à jour pour autoriser l'accès à la ressource objet à l'aide de l'iot:[DescribeThing](#)action.

Si vous avez encore besoin d'aide, contactez votre administrateur AWS. Votre administrateur vous a fourni vos informations de connexion.

Utilisation de AWS IoT Device Advisor

Si vous utilisez AWS IoT Device Advisor, l'exemple d'erreur suivant se produit lorsque l'utilisateur mateojackson tente d'utiliser la console pour afficher des informations détaillées concernant une définition de suite mais ne dispose pas des iotdeviceadvisor:[GetSuiteDefinition](#) autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:  
iotdeviceadvisor:GetSuiteDefinition on resource: MySuiteDefinition
```

Dans ce cas, la politique de l'mateojacksonutilisateur doit être mise à jour pour autoriser l'accès à la [MySuiteDefinition](#)ressource à l'aide de l'iotdeviceadvisor:[GetSuiteDefinition](#)action.

Je ne suis pas autorisé à exécuter iam : PassRole

Si vous recevez une erreur selon laquelle vous n'êtes pas autorisé à exécuter iam:PassRole l'action, vos stratégies doivent être mises à jour afin de vous permettre de transmettre un rôle à AWS IoT.

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, les stratégies de Mary doivent être mises à jour pour lui permettre d'exécuter l'action iam:PassRole.

Si vous avez encore besoin d'aide, contactez votre administrateur AWS. Votre administrateur vous a fourni vos informations de connexion.

Je veux autoriser des personnes extérieures à mon Compte AWS à accéder à mes ressources AWS IoT

Vous pouvez créer un rôle que les utilisateurs provenant d'autres comptes ou les personnes extérieures à votre organisation pourront utiliser pour accéder à vos ressources. Vous pouvez spécifier la personne à qui vous souhaitez confier le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACL), vous pouvez utiliser ces politiques pour 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 de AWS IoT avec IAM \(p. 419\)](#).
- Pour savoir comment octroyer l'accès à vos ressources à des Comptes AWS dont vous êtes propriétaire, consultez la section [Fournir l'accès à un utilisateur IAM dans un autre Compte AWS que vous possédez](#) dans le Guide de l'utilisateur IAM.
- Pour savoir comment octroyer l'accès à vos ressources à des Comptes AWS tiers, consultez [Fournir l'accès aux 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, consultez [Fournir un accès à des utilisateurs authentifiés en externe \(fédération d'identité\)](#) dans le Guide de l'utilisateur IAM.
- Pour découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des stratégies basées sur les ressources pour l'accès à des 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 recueillir les données de surveillance de toutes les parties de votre solution AWS de façon à pouvoir déboguer plus facilement un éventuel échec multipoint. Pour de plus amples informations sur les procédures de journalisation et de surveillance, veuillez consulter [Surveillance des AWS IoT \(p. 466\)](#)

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 :

- Amazon CloudWatch Alarms : surveillez une seule métrique sur une période 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, veuillez consulter [Surveillez les AWS IoT alarmes et les métriques à l'aide d'Amazon CloudWatch \(p. 474\)](#).

- 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 critiques des scénarios de test de AWS IoT Device Advisor, les événements générés et les messages MQTT envoyés depuis vos appareils ou AWS IoT Core pendant l'exécution des tests. Ces journaux permettent de déboguer et de prendre des mesures correctives sur vos appareils. Pour plus d'informations, consultez [Surveiller AWS IoT à l'aide CloudWatch des journaux \(p. 490\)](#). Pour plus d'informations sur l'utilisation d'AmazonCloudWatch, consultez la section [Monitoring Log Files](#) du Guide de CloudWatch l'utilisateur Amazon.
- 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 plus d'informations, consultez la section [Qu'est-ce qu'Amazon CloudWatch Events](#) dans le guide de CloudWatch l'utilisateur Amazon.
- AWS CloudTrailSurveillance de journaux — Partagez les fichiers journaux en temps réel en envoyant à Logs, envoyez-les à Logs, écrivez des applications de traitement de CloudTrail journaux en temps réel en envoyant à CloudWatch Logs, écrivez des applications de traitement de journaux en temps réel en envoyant à Logs, écrivez des applications de traitement de journaux en temps réel en envoyant à Logs, écrivez des applications de traitement de journaux en CloudTrail temps réel en envoyant Pour plus d'informations, voir [Journalisation des appels d'API AWS IoT avec AWS CloudTrail \(p. 521\)](#) et également [Utilisation des fichiers CloudTrail journaux](#) du Guide de l'AWS CloudTrailutilisateur.

Outils de surveillance manuelle

La surveillance de AWS IoT implique également de surveiller manuellement les éléments que les alarmes CloudWatch ne couvrent pas. Les AWS IoT tableaux de bord de la console de AWS serviceCloudWatch,, fournissent une at-a-glance vue de l'état de votre AWS environnement. Nous recommandons de consulter également les fichiers journaux sur AWS IoT.

- Le tableau de bord AWS IoT affiche :
 - Certificats CA
 - Certificats
 - Stratégies
 - Règles
 - 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 tâches 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 vos métriques de ressources AWS.
- Créer et modifier des alarmes pour être informé des problèmes.

Validation de la conformité pour AWS IoT Core

Pour savoir si un Service AWS fait partie du champ d'application de programmes de conformité spécifiques, consultez [Services AWS dans le champ d'application par programme de conformité](#) et choisissez le programme de conformité qui vous intéresse. Pour obtenir des renseignements généraux, consultez [Programmes de conformité AWS](#).

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour plus d'informations, consultez [Téléchargement de rapports dans AWS Artifact](#).

Votre responsabilité de conformité lors de l'utilisation de Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise, ainsi que par la législation et la réglementation applicables. 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 traitent de considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de référence dans AWS centrés sur la sécurité et la conformité.
- [Architecture pour la sécurité et la conformité HIPAA sur Amazon Web Services](#) : ce livre blanc décrit comment les entreprises peuvent utiliser AWS pour créer des applications éligibles à la loi HIPAA.

Note

Tous les Services AWS ne sont pas éligibles à HIPAA. Pour plus d'informations, consultez le [HIPAA Eligible Services Reference](#).

- [Ressources de conformité AWS](#) : cet ensemble de manuels et de guides peut s'appliquer à votre secteur d'activité et à votre emplacement.
- [Évaluation des ressources à l'aide de règles](#) dans le Guide du développeur AWS Config : 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 dans AWS. Security Hub utilise des contrôles de sécurité pour évaluer vos ressources AWS et vérifier votre conformité par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Pour obtenir la liste des services et des contrôles pris en charge, veuillez consulter la [Référence des contrôles Security Hub](#).
- [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.

La résilience au cœur de AWS IoT

L'infrastructure mondiale d'AWS est construite autour de zones de disponibilité et de Région AWS. Les régions 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 de plus amples informations sur Région AWS les et les zones de disponibilité, veuillez consulter [Infrastructure AWS mondiale d'](#).

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 panne matérielle ou réseau, ces données sont automatiquement répliquées entre les zones de disponibilité, mais pas entre les régions.

AWS IoT Core publie les événements MQTT lorsque le registre des appareils est mis à jour. Vous pouvez utiliser ces messages pour sauvegarder les données de votre 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. Device Shadow stocke les données d'état de vos appareils et peut être renvoyée lorsqu'un appareil revient en ligne. AWS IoT Device Advisor stocke les informations relatives à 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 spécifiques à une région et ne sont pas répliquées à Régions AWS moins que vous ne le fassiez spécifiquement.

Pour plus d'informations sur les bonnes pratiques de sécurité, consultez[Bonnes pratiques de sécurité dans AWS IoT Core \(p. 459\)](#).

Utilisation d'AWS IoT Core avec les points de terminaison d'un VPC d'interface

Avec AWS IoT Core, vous pouvez créer des points de [terminaison de données IoT au sein de votre VPC en utilisant des points](#) de terminaison d'[interface VPC](#). Les points de terminaison d'un VPC d'interface reposent sur AWS PrivateLink, une AWS technologie qui vous permet d'accéder aux services exécutés à l'aide AWS d'adresses IP privées. Pour en savoir plus, consultez [Amazon Virtual Private Cloud](#).

Pour connecter des appareils sur le terrain sur des réseaux distants, tels qu'un réseau d'entreprise à votre AWS VPC, consultez les différentes options répertoriées dans la matrice de connectivité [réseau vers Amazon VPC](#).

Sujets des chapitres :

- [Création des points de terminaison d'un VPC pour AWS IoT Core \(p. 456\)](#)
- [Contrôle de l'accès aux points de AWS IoT Core terminaison de plusieurs VPC \(p. 457\)](#)
- [Limites des points de terminaison d'un VPC \(p. 458\)](#)
- [Mise à l'échelle des points de terminaison VPC avec IoT Core \(p. 458\)](#)
- [Utilisation de domaines personnalisés avec des points de terminaison VPC \(p. 458\)](#)
- [Disponibilité des points de terminaison VPC pour AWS IoT Core \(p. 458\)](#)

Création des points de terminaison d'un VPC pour AWS IoT Core

Pour commencer à utiliser les points de terminaison VPC, il vous suffit de [créer un point de terminaison VPC d'interface](#) et de le sélectionner AWS IoT Core comme service. AWS Si vous utilisez la CLI, appelez d'abord [describe-vpc-endpoint-services](#) pour vous assurer que vous choisissez une zone de disponibilité présente dans votre zone particulière. AWS IoT Core Région AWS Par exemple, dans us-east-1, cette commande devrait ressembler à ceci :

```
aws ec2 describe-vpc-endpoint-services --service-name com.amazonaws.us-east-1.iot.data
```

Note

La fonctionnalité VPC permettant de créer automatiquement un enregistrement DNS est désactivée. Pour vous connecter à ces points de terminaison, vous devez créer manuellement un enregistrement DNS privé. Pour de plus amples informations sur les enregistrements DNS privés des VPC, veuillez consulter [DNS privé des points de terminaison d'interface](#). Pour plus d'informations sur les limitations d'un AWS IoT Core VPC, consultez [Limites des points de terminaison d'un VPC \(p. 458\)](#).

Pour connecter les clients MQTT aux interfaces de point de terminaison du VPC, vous devez créer manuellement des enregistrements DNS dans une zone hébergée privée rattachée à votre VPC. Pour commencer, consultez la section [Création d'une zone hébergée privée](#). Dans votre zone hébergée privée, créez un enregistrement d'alias pour chaque adresse IP d'elastic network interface pour le point de terminaison du VPC. Si vous disposez de plusieurs adresses IP d'interface réseau pour plusieurs points de terminaison VPC, créez des enregistrements DNS pondérés avec des poids

égaux pour tous les enregistrements pondérés. Ces adresses IP sont disponibles à partir de l'appel d'[DescribeNetworkInterfaces](#) API lorsqu'elles sont filtrées par l'ID de point de terminaison du VPC dans le champ de description.

Contrôle de l'accès aux points de AWS IoT Core terminaison de plusieurs VPC

Vous pouvez restreindre l'accès aux appareils AWS IoT Core pour qu'il soit autorisé uniquement via le point de terminaison du VPC à l'aide des clés [contextuelles de condition](#) du VPC. AWS IoT Core prend en charge les clés de contexte liées au VPC suivantes :

- [SourceVpc](#)
- [SourceVpce](#)
- [VPC SourceIp](#)

Note

AWS IoT Core ne prend pas en charge les politiques relatives aux points de terminaison des <https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints-access.html#vpc-endpoint-policies> VPC pour le moment.

Par exemple, la politique suivante accorde l'autorisation de se connecter à AWS IoT Core l'aide d'un ID client correspondant au nom de l'objet et de publier sur n'importe quel sujet préfixé par le nom de l'objet, à condition que l'appareil se connecte à un point de terminaison VPC avec un ID de point de terminaison VPC particulier. Cette politique refuserait les tentatives de connexion à votre point de terminaison de données IoT public.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:client/  
${iot:Connection.Thing.ThingName}"  
            ],  
            "Condition": {  
                "StringEquals": {  
                    "aws:SourceVpce": "vpce-1a2b3c4d"  
                }  
            }  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iot:us-east-1:123456789012:topic/  
${iot:Connection.Thing.ThingName}/*"  
            ]  
        }  
    ]  
}
```

Limites des points de terminaison d'un VPC

Cette section décrit les limites des points de terminaison VPC par rapport aux points de terminaison publics.

- Les points de terminaison VPC sont actuellement pris en charge uniquement pour les points de terminaison de [données IoT](#)
- Les périodes de maintien en vie du MQTT sont limitées à 230 secondes. Le fait de rester en vie plus longtemps que cette période sera automatiquement réduite à 230 secondes
- Chaque point de terminaison VPC prend en charge 100 000 appareils connectés simultanément au total. Si vous avez besoin de connexions supplémentaires, consultez[Mise à l'échelle des points de terminaison VPC avec IoT Core \(p. 458\)](#).
- Les points de terminaison d'un VPC prennent en charge le trafic IPv4 uniquement.
- Les points de terminaison VPC ne fourniront que [des certificats ATS](#), à l'exception des domaines personnalisés.
- Les [politiques relatives aux terminaux VPC](#) ne sont pas prises en charge pour le moment.
- Pour les points de terminaison VPC créés pour le plan de AWS IoT Core données, l'utilisation d'enregistrements DNS publics zonaux ou régionaux AWS IoT Core n'est pas prise en charge.

Mise à l'échelle des points de terminaison VPC avec IoT Core

AWS IoT CoreLes points de terminaison d'interface VPC sont limités à 100 000 appareils connectés sur un seul point de terminaison d'interface. Si votre cas d'utilisation nécessite davantage de connexions simultanées au broker, nous vous recommandons d'utiliser plusieurs points de terminaison VPC et de router manuellement vos appareils via les points de terminaison de votre interface. Lorsque vous créez des enregistrements DNS privés pour acheminer le trafic vers les points de terminaison de votre VPC, veillez à créer autant d'enregistrements pondérés que de points de terminaison VPC afin de répartir le trafic entre vos multiples points de terminaison.

Utilisation de domaines personnalisés avec des points de terminaison VPC

Si vous souhaitez utiliser des domaines personnalisés avec des points de terminaison VPC, vous devez créer vos enregistrements de nom de domaine personnalisés dans une zone hébergée privée et créer des enregistrements de routage dans Route53. Pour plus d'informations, consultez [Création d'une zone hébergée privée](#).

Disponibilité des points de terminaison VPC pour AWS IoT Core

AWS IoT CoreLes points de terminaison d'interface VPC sont disponibles dans toutes les régions prises en [AWS IoT Corecharge](#).

Sécurité de l'infrastructure dans AWS IoT

En tant qu'ensemble de services gérés, AWS IoT est protégé par les procédures de sécurité du réseau AWS mondial qui sont décrites dans le livre blanc [Amazon Web Services : Présentation des procédures de sécurité](#).

Vous utilisez les appels d'API publiés 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 aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La plupart des systèmes modernes telles que Java 7 et versions ultérieures prennent en charge ces modes. Pour plus d'informations, veuillez consulter [Sécurité du transport dans AWS IoT Core \(p. 409\)](#).

Les demandes doivent être signées à l'aide d'un identifiant 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 et signer les demandes.

Surveillance de la sécurité des flottes de production ou des appareils avec Core AWS IoT

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 rendent la configuration de la flotte complexe et sujette aux 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 utiliser AWS IoT Device Defender pour analyser, auditer et surveiller les appareils connectés afin de détecter les comportements anormaux, et d'atténuer les risques pour la sécurité. AWS IoT Device Defender peut auditer les parcs d'appareils pour s'assurer qu'ils respectent les meilleures pratiques de sécurité et 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, veuillez consulter [AWS IoT Device Defender \(p. 975\)](#).

AWS IoT Device Advisor diffuse les mises à jour et les correctifs nécessaires à votre parc. AWS IoT Device Advisor met automatiquement à jour les scénarios de test. Les scénarios de test que vous sélectionnez utilisent toujours la version la plus récente. Pour plus d'informations, veuillez consulter [Device Advisor \(p. 1167\)](#).

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 sur les règles de sécurité pour les solutions IoT industrielles, voir [Dix règles d'or de sécurité pour les solutions IoT industrielles](#).

Protection des connexions MQTT dans AWS IoT

[AWS IoT Core](#) est un service cloud géré qui permet aux appareils connectés d'interagir facilement et en toute sécurité avec des applications cloud et d'autres appareils. AWS IoT Core prend en charge le [WebSocketHTTP](#) et le [MQTT](#), un protocole de communication léger spécialement conçu pour tolérer les connexions intermittentes. Si vous vous connectez à AWS IoT l'aide de MQTT, chacune de vos connexions doit être associée à un identifiant appelé 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 chacun Compte AWS d'euxRégion AWS. Cela signifie que vous n'avez pas besoin d'appliquer l'unicité globale des identifiants clients en dehors de votre région Compte AWS ou entre les régions au sein de votreCompte 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 \(p. 94\)](#)).
- Contraintes liées aux ressources de l'appareil.
- La cause première des déconnexions, leur agressivité et leur persistance.

Pour éviter les conflits d'identifiants clients et leurs répercussions négatives potentielles, assurez-vous que chaque appareil ou application mobile dispose d'une politique AWS IoT ou IAM qui limite les identifiants clients pouvant être utilisés pour les connexions MQTT au AWS IoT courtier de messages. Par exemple, vous pouvez utiliser une politique IAM pour empêcher un appareil de fermer involontairement la connexion d'un autre appareil en utilisant un ID client déjà utilisé. Pour plus d'informations, veuillez consulter [Autorisation \(p. 355\)](#).

Tous les appareils de votre parc doivent avoir des identifiants avec des priviléges 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 politiques d'autorisation, vous pouvez utiliser [Variables de stratégie AWS IoT Core \(p. 362\)](#) des [variables de politique 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 le AWS IoT registre, vous pouvez utiliser les [variables de politique des objets dans les \(p. 364\)](#) AWS IoT politiques pour accorder ou refuser des autorisations en fonction des propriétés des objets, telles que les noms, les types d'objets et les valeurs des attributs des objets. 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 politique de l'objet sont remplacées lorsqu'un objet se connecte AWS IoT via MQTT à l'aide de l'authentification mutuelle TLS ou via le WebSocket protocole MQTT à l'aide d'identités Amazon Cognito authentifiées](#). Vous pouvez utiliser l'[AttachThingPrincipalAPI](#) pour associer des certificats et des identités Amazon Cognito authentifiées à un objet. `iot:Connection.Thing.ThingName`est une variable de politique utile pour appliquer les restrictions relatives à l'identification des clients. 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:region:account:thing/iot:Connection.Thing.ThingName"  
      ]  
    }  
  ]  
}
```

```
        "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"  
    ]  
}  
]  
}
```

Si vous souhaitez identifier les conflits d'ID client en cours, vous pouvez activer et utiliser [CloudWatchLogs for AWS IoT \(p. 490\)](#). 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 un [filtre CloudWatch Logs](#), par exemple `$.reason= "DUPLICATE_CLIENT_ID"` pour rechercher des instances de conflits d'ID client ou pour configurer des [filtres CloudWatch métriques](#) et les CloudWatch alarmes correspondantes à des fins de surveillance et de génération de rapports continus.

Vous pouvez utiliser [AWS IoTDevice Defender](#) pour identifier les politiques trop permissives AWS IoT et IAM. AWS IoT Device Defender propose également un audit qui vous avertit si plusieurs appareils de votre parc se connectent au courtier de AWS IoT messages en utilisant le même identifiant client.

Vous pouvez utiliser AWS IoT Device Advisor pour vérifier que vos appareils peuvent se connecter de manière fiable AWS IoT Core et suivre les meilleures pratiques de sécurité.

Consulter aussi

- [AWS IoT Core](#)
- [Fonctions de sécurité AWS IoT \(p. 316\)](#)
- [Variables de stratégie AWS IoT Core \(p. 362\)](#)
- [Variables de politique IAM](#)
- [Amazon Cognito Identity](#)
- [Device Defender AWS IoT](#)
- Journaux [CloudWatch pour AWS IoT \(p. 490\)](#)

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, vous avez une application qui consiste en un téléphone mobile qui reçoit des mises à jour d'état provenant de deux objets domotiques 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 priviléges 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-en un second Région AWS comme sauvegarde

Envisagez de stocker une copie de vos données en une seconde à Région AWS titre de sauvegarde. Pour de plus amples informations, veuillez consulter [Récupération après sinistre pour AWS 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, veuillez consulter [ust-in-timeAppvisionnement en J \(p. 903\)](#).

Autorisations pour exécuter des tests AWS IoT Device Advisor

Le modèle de politique suivant indique les autorisations minimales et l'entité IAM requises pour exécuter des scénarios de test de AWS IoT Device Advisor. Vous devrez le remplacer par le rôle *your-device-role-arn* d'appareil Amazon Resource Name (ARN) que vous avez créé selon les [prérequis](#).

```
{  
    "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": [  
                "execute-api:Invoke*",  
                "iam>ListRoles", // Required to list device roles in the Device Advisor  
                "iot:Connect",  
                "iot>CreateJob",  
                "iot>DeleteJob",  
                "iot:DescribeCertificate",  
                "iot:DescribeEndpoint",  
                "iot:DescribeJobExecution",  
                "iot:DescribeJob",  
                "iot:DescribeThing",  
                "iot:GetPendingJobExecutions",  
                "iot:GetPolicy",  
                "iot>ListAttachedPolicies",  
                "iot>ListCertificates",  
                "iot>ListPrincipalPolicies",  
                "iot>ListThingPrincipals",  
                "iot>ListThings",  
                "iot:Publish",  
                "iot:StartNextPendingJobExecution",  
                "iot:UpdateJobExecution",  
                "iot:UpdateThingShadow",  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs:DescribeLogGroups",  
                "logs:DescribeLogStreams",  
                "logs:PutLogEvents",  
                "logs:PutRetentionPolicy"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

```

        "Resource": "*"
    },
    {
        "Sid": "VisualEditor2",
        "Effect": "Allow",
        "Action": "iotdeviceadvisor:/*",
        "Resource": "*"
    }
]
}

```

Prévention confus entre services pour Device Advisor

Le problème de l'adjoint confus est un problème de sécurité dans lequel une entité qui n'a pas l'autorisation d'effectuer une action peut contraindre une entité plus privilégiée à effectuer cette action. Dans AWS, l'emprunt d'identité entre services peut entraîner le problème de député confus. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé et ses autorisations utilisées pour agir sur les ressources d'un autre client auxquelles on ne serait pas autorisé d'accéder autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Nous vous recommandons d'utiliser les clés de condition [aws:SourceAccount](#) globale [aws:SourceArn](#) les clés de condition globale dans les politiques de ressources afin de limiter les autorisations à la ressource octroyées par Device Advisor à un autre service. Si vous utilisez les deux clés de contexte de condition globale, la valeur `aws:SourceAccount` et le compte de la valeur `aws:SourceArn` doit utiliser le même ID de compte lorsqu'il est utilisé dans la même déclaration de politique.

La valeur de `aws:SourceArn` doit être l'ARN de la ressource de définition de votre suite. La ressource de définition de la suite fait référence à la suite de tests que vous avez créée avec Device Advisor.

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:iotdeviceadvisor:*:account-id:suitedefinition/*`

L'exemple suivant montre comment utiliser les clés de condition `aws:SourceAccount` globale `aws:SourceArn` et les clés de condition globale dans Device Advisor afin d'éviter le problème de l'adjoint confus.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ConfusedDeputyPreventionExamplePolicy",
            "Effect": "Allow",
            "Principal": {
                "Service": "iotdeviceadvisor.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iotdeviceadvisor:us-east-1:123456789012:suitedefinition/ygp6rxax3tzvn"
                },
                "StringEquals": {
                    "aws:SourceAccount": "123456789012"
                }
            }
        }
    ]
}

```

}

Formation et certification AWS

Pour en savoir plus sur les concepts clés relatifs à la sécurité AWS IoT, consultez les informations [d'introduction à la sécurité AWS IoT.](#)

Surveillance des 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 \(p. 467\)](#) et établir vos lignes de base de performances.

- Quels sont les objectifs de la surveillance ?
- Quelles sont les ressources à surveiller ?
- A 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 \(p. 467\)](#) 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 \(p. 481\)](#)
- [PublishOut.Success \(p. 481\)](#)
- [Subscribe.Success \(p. 481\)](#)
- [Ping.Success \(p. 481\)](#)
- [Connect.Success \(p. 481\)](#)
- [GetThingShadow.Accepted \(p. 484\)](#)
- [UpdateThingShadow.Accepted \(p. 484\)](#)
- [DeleteThingShadow.Accepted \(p. 484\)](#)
- [RulesExecuted \(p. 479\)](#)

Les rubriques de cette section peuvent vous aider à démarrer la journalisation et la surveillance de AWS IoT.

Rubriques

- [Configurer la journalisation AWS IoT \(p. 467\)](#)
- [Surveillez les AWS IoT alarmes et les métriques à l'aide d'Amazon CloudWatch \(p. 474\)](#)
- [Surveiller AWS IoT à l'aide CloudWatch des journaux \(p. 490\)](#)
- [Importer les journaux côté appareil sur Amazon CloudWatch \(p. 512\)](#)

- [Journalisation des appels d'API AWS IoT avec AWS CloudTrail \(p. 521\)](#)

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 \(p. 474\)](#) 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 connexionAWS IoT, vous devez créer un rôle IAM et une politique qui vous AWS autorise à surveiller AWS IoT l'activité en votre nom.

Note

Avant d'activer la AWS IoT journalisation, assurez-vous de bien comprendre les autorisations d'accès aux CloudWatch journaux. Les utilisateurs ayant accès aux CloudWatch journaux peuvent consulter les informations de débogage de vos appareils. Pour de plus amples informations, veuillez consulter [Authentification et contrôle d'accès pour Amazon CloudWatch Logs](#).

Si vous vous attendez à des modèles de trafic élevés en AWS IoT Core raison des tests de charge, pensez à désactiver la journalisation de l'IoT pour éviter tout ralentissement. Si un trafic élevé est détecté, notre service peut désactiver la connexion à votre compte.

La section suivante montre comment créer un rôle de journalisation et une politique pour les AWS IoT Core ressources. Pour plus d'informations sur la manière de créer un rôle et une politique de journalisation IAM AWS IoT Core pour le LoRa WAN, consultez[Création d'un rôle et d'une politique de journalisation pourAWS IoT Wireless \(p. 1453\)](#).

Création d'un rôle de journalisation

Pour créer un rôle de journalisation, ouvrez le [hub Rôles de la console IAM](#) et choisissez Créeer un rôle.

1. Sous Sélectionner une entité de confiance, choisissez AWSService. Choisissez ensuite IoT sous Cas d'utilisation. Si l'Internet des IoT n'est pas visible, saisissez et recherchez l'Internet des objets dans le menu déroulant Cas d'utilisation pour d'autres AWS services :. Sélectionnez Suivant.
2. Sur la page Ajouter des autorisations, vous verrez les politiques qui sont automatiquement associées au rôle de service. Choisissez Suivant.
3. Sur la page Nom, révision et création, entrez le nom du rôle et la description du rôle, puis choisissez Créeer un rôle.
4. Dans la liste des rôles, recherchez le rôle que vous avez créé, ouvrez-le et copiez l'ARN du rôle (*logging-role-arn*) à utiliser lorsque vous[Configurez la journalisation par défaut dans AWS IoT \(console\) \(p. 469\)](#).

Stratégie de rôle de journalisation

Les documents de stratégie suivants fournissent la stratégie de rôle et la stratégie d'approbation qui permettent à AWS IoT d'envoyer des entrées de journaux à CloudWatch en votre nom. Si vous avez également autorisé le LoRa WAN AWS IoT Core à envoyer des entrées de journal, vous verrez un document de politique créé pour vous qui enregistre les deux activités. Pour plus d'informations sur la création d'un rôle et d'une politique de journalisation IAM AWS IoT Core pour le LoRa WAN, consultez [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#).

Note

Ces documents ont été créés pour vous lorsque vous avez créé le rôle de journalisation. Les documents contiennent des variables `#{partition}`, `#{region}`, et `#{accountId}`, que vous devez remplacer par vos valeurs.

Stratégie de rôle :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs>CreateLogGroup",  
                "logs>CreateLogStream",  
                "logs>PutLogEvents",  
                "logs>PutMetricFilter",  
                "logs>PutRetentionPolicy",  
                "iot>GetLoggingOptions",  
                "iot>SetLoggingOptions",  
                "iot>SetV2LoggingOptions",  
                "iot>GetV2LoggingOptions",  
                "iot>SetV2LogLevel",  
                "iot>ListV2LoggingLevels",  
                "iot>DeleteV2LogLevel"  
            ],  
            "Resource": [  
                "arn:#{partition}:logs:#{region}:#{accountId}:log-group:AWSIotLogsV2:*"  
            ]  
        }  
    ]  
}
```

Politique de confiance permettant de ne consigner que AWS IoT Core l'activité :

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

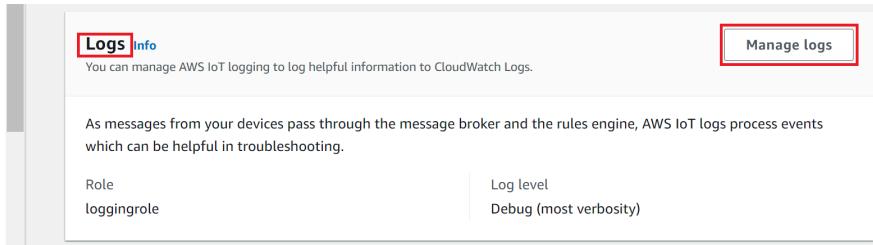
Configurez la journalisation par défaut dans AWS IoT (console)

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 \(interface de ligne de commande\) \(p. 472\)](#).

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, veuillez consulter [Ouvrez la AWS IoT console. \(p. 20\)](#).
2. Dans le panneau de navigation de gauche, choisissez Paramètres. Dans la section Journaux de la page Paramètres, choisissez Gérer les journaux.

La page Journaux affiche le rôle de journalisation et le niveau de verbosité utilisés par tous. AWS IoT



3. Sur la page Journaux, choisissez Sélectionner un rôle pour spécifier un rôle dans [Création d'un rôle de journalisation \(p. 467\)](#) lequel vous avez créé ou Créez un rôle pour créer un nouveau rôle à utiliser pour la journalisation.

Logs Info

Log role Info

Create or select the role you want to use to log information to CloudWatch Logs.

Select role

loggingrole

Create role

Attach policy to IAM role permitting AWS IoT to publish logs to CloudWatch on your behalf.

Log level Info

Select how detailed you want your logs to be. Selecting Error (least verbose) logs only errors and is the least detailed. Selecting Debug (most verbose) creates the most detailed logs. Collecting more detailed logs can increase logging costs.

Log level

Debug (most verbosity)

Cancel

Update

4. Choisissez le niveau de journal qui décrit le [niveau de détail \(p. 474\)](#) des entrées de journal que vous souhaitez voir apparaître dans les CloudWatch journaux.
5. Choisissez Mettre à jour pour enregistrer vos modifications.

Une fois que vous avez activé la journalisation, visitez [Affichage des journaux AWS IoT dans la console CloudWatch \(p. 490\)](#) 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 \(p. 467\)](#) avant de continuer.

Le principal utilisé pour appeler l'API doit avoir [Transmettre les autorisations de rôle \(p. 526\)](#) 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 accorde AWS IoT l'autorisation d'écrire dans vos CloudWatch journaux dans Logs.

--default-log-level

Le [niveau de journalisation \(p. 474\)](#) à utiliser. Les valeurs valides 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 des journaux AWS IoT dans la console CloudWatch \(p. 490\)](#) 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 \(p. 294\)](#) 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.

Outre les groupes d'objets, vous pouvez également enregistrer des cibles telles que l'ID client, l'adresse IP source et l'ID principal d'un appareil.

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 \(p. 467\)](#) avant de continuer.

Le principal utilisé pour appeler l'API doit avoir [Transmettre les autorisations de rôle \(p. 526\)](#) 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-v2LoggingOptions](#) 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 accorde AWS IoT l'autorisation d'écrire dans vos CloudWatch journaux dans Logs.

--default-log-level

Le [niveau de journalisation \(p. 474\)](#) à utiliser. Les valeurs valides 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-v2LoggingLevel](#) 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 `target_type` valeur doit être l'une des suivantes : THING_GROUP | CLIENT_ID | SOURCE_IP |PRINCIPAL_ID. 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

```
aws iot set-v2-logging-level \
    --log-target targetType=CLIENT_ID,targetName=ClientId1 \
    --log-level DEBUG
```

- Utilisez la commande [list-v2-logging-levels](#) pour répertorier les niveaux de journalisation actuellement configurés.

```
aws iot list-v2-logging-levels
```

- Utilisez la [delete-v2-logging-level](#) commande pour supprimer un niveau de journalisation spécifique à une ressource, comme dans les exemples suivants.

```
aws iot delete-v2-logging-level \
    --target-type "THING_GROUP" \
    --target-name "thing_group_name"
```

```
aws iot delete-v2-logging-level \
    --target-type=CLIENT_ID \
    --target-name=ClientId1
```

--targetType

La `target_type` valeur doit être l'une des suivantes : THING_GROUP | CLIENT_ID | SOURCE_IP |PRINCIPAL_ID.

--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 des journaux AWS IoT dans la console CloudWatch \(p. 490\)](#) 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.

Surveillez les AWS IoT alarmes et les métriques à l'aide d'Amazon CloudWatch

Vous pouvez surveiller AWS IoT avec CloudWatch, qui recueille et traite les données brutes de AWS IoT en métriques lisibles et disponibles presque 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, les données des métriques AWS IoT sont automatiquement envoyées à CloudWatch toutes les minutes. Pour plus d'informations, consultez [Qu'est-ce qu'AmazonCloudWatch, Amazon CloudWatch Events et Amazon CloudWatch Logs ?](#) dans le guide de CloudWatch l'utilisateur d'Amazon.

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 ? \(p. 475\)](#)
- [Comment puis-je être informé si mes objets ne publient pas de données chaque jour ? \(p. 476\)](#)
- [Comment puis-je être informé si les mises à jour de mon thing shadow sont rejetées chaque jour ? \(p. 477\)](#)
- [Comment créer une alarme CloudWatch pour les travaux ? \(p. 477\)](#)

En savoir plus sur les alarmes et les métriques CloudWatch

- [Création d'alarmes CloudWatch pour surveiller AWS IoT \(p. 475\)](#)
- [Métriques et dimensions AWS IoT \(p. 478\)](#)

Création d'alarmes CloudWatch pour surveiller AWS IoT

Vous pouvez créer une CloudWatch alarme qui envoie un message d'Amazon SNS lorsque l'alarme change d'état. 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 peut être une notification envoyée à une rubrique Amazon SNS ou à une politique Auto Scaling. Les alertes appellent les actions pour les changements d'état soutenus uniquement pour les changements d'état soutenus. CloudWatchles alarmes n'appellent pas d'actions simplement flotées 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 ? \(p. 475\)](#)
- [Comment puis-je être informé si mes objets ne publient pas de données chaque jour ? \(p. 476\)](#)
- [Comment puis-je être informé si les mises à jour du shadow de mon objet sont rejetées chaque jour ? \(p. 477\)](#)
- [Comment créer une alarme CloudWatch pour les travaux ? \(p. 477\)](#)

Vous pouvez voir toutes les métriques sur lesquelles les alarmes CloudWatch peuvent surveiller [Métriques et dimensions AWS IoT \(p. 478\)](#).

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 Amazon Resource Name (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que `sns-topic-arn`.
Pour plus d'informations sur la façon de créer une notification Amazon SNS, consultez [Premiers pas avec Amazon SNS](#).
2. Créez l'alerte.

```
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 votre [console CloudWatch](#).

Comment puis-je être informé si mes objets ne publient pas de données chaque jour ?

1. Créez une rubrique Amazon SNS nommée `things-not-publishing-data` et enregistrez son Amazon Resource Name (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que *sns-topic-arn*.

Pour plus d'informations sur la façon de créer une notification Amazon SNS, consultez [Premiers pas avec Amazon SNS](#).

2. Créez l'alerte.

```
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 votre [console 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 nommée `things-shadow-updates-rejected` et enregistrez son Amazon Resource Name (ARN). Cette procédure fera référence à l'ARN de votre rubrique en tant que `sns-topic-arn`.

Pour plus d'informations sur la façon de créer une notification Amazon SNS, consultez [Premiers pas avec Amazon SNS](#).

2. Créez l'alerte.

```
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 votre [console 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 les [Métriques de tâches \(p. 485\)](#).

La commande suivante crée une alarme CloudWatch pour surveiller le nombre total d'exécutions de tâches ayant échoué pour `SampleOTAJob` et vous avertit lorsque plus de 20 exécutions de tâches ont échoué. L'alarme surveille la métrique `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 se déclenche, 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-executions
```

La commande suivante crée une alarme CloudWatch pour surveiller le nombre d'exécutions de travaux ayant échoué pour **SampleOTAJob** au cours d'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 se déclenche, 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-executions-per-hour
```

Métriques et dimensions AWS IoT

Lorsque vous interagissez avec AWS 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](#).
2. Dans le panneau de navigation, choisissez Métriques, puis choisissez Toutes les métriques.
3. Dans l'onglet Parcourir, recherchez AWS IoT pour afficher la liste des mesures.

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 pour AWS IoT :

- [Métriques AWS IoT \(p. 479\)](#)
- [AWS IoT Corestatistiques relatives aux fournisseurs d'informations d'identification \(p. 479\)](#)
- [Métriques de règle \(p. 479\)](#)

- [Métriques d'action de règle \(p. 480\)](#)
- [Métriques spécifiques à l'action HTTP \(p. 480\)](#)
- [Métriques d'agent de messages \(p. 481\)](#)
- [Métriques de shadow d'appareil \(p. 484\)](#)
- [Métriques de tâches \(p. 485\)](#)
- [Métriques d'audit Device Defender \(p. 486\)](#)
- [Métriques de détection Device Defender \(p. 486\)](#)
- [Métriques de mise en service d'appareils \(p. 487\)](#)
- [Métriques d'indexation de la flotte \(p. 489\)](#)
- [Dimensions pour les métriques \(p. 489\)](#)

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 nombre 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 Corestatistiques relatives aux fournisseurs d'informations d'identification

Métrique	Description
CredentialExchangeSuccess	Le nombre de AssumeRoleWithCertificate demandes adressées avec succès au fournisseur AWS IoT Core 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.

Métrique	Description
RuleNotFound	La règle à déclencher est introuvable. La dimension RuleName contient le nom de la règle.
RulesExecuted	Nombre de règles AWS IoT exécutées.
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.
ErrorActionFailure	Nombre d'action d'erreur ayant échoué. La dimension RuleName contient le nom de la règle qui spécifie l'action. La dimension ActionType contient le type d'action ayant été appelé.
ErrorActionSuccess	Le nombre d'actions d'erreur réussies. 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://.

Métrique	Description
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 sous Métriques 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 AWS IoTQuotas (p. 1514) . 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

Métrique	Description
	contient le protocole utilisé pour publier le message. HTTP Publish ne prend pas en charge cette métrique.
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 AWS IoTQuotas (p. 1514) . La dimension Protocol contient le protocole utilisé pour publier le message. HTTP Publish ne prend pas en charge cette métrique.
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. HTTP Publish ne prend pas en charge cette métrique.
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 autorisé. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH. HTTP Publish ne prend pas en charge cette métrique.
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 AWS IoTQuotas (p. 1514) . 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 qui ont été limitées parce que le client a dépassé le débit de messages sortants autorisé. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH.
PublishRetained.AuthError	Nombre de demandes de publication avec l'RETAINindicateur défini que le courtier de messages n'a pas pu autoriser. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH.
PublishRetained.ServerError	Nombre de demandes de publication conservées que le gestionnaire de messages n'a pas pu traiter en raison d'une erreur interne. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH.

Métrique	Description
PublishRetained.Success	Nombre de demandes de publication avec l'RETAINindicateur défini qui ont été traitées avec succès par le courtier de messages. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH.
PublishRetained.Throttle	Nombre de demandes de publication avec l'RETAINindicateur activé qui ont été limitées parce que le client a dépassé le débit de messages entrants autorisé. La dimension Protocol contient le protocole utilisé pour envoyer le message PUBLISH.
Queued.Success	Nombre de messages stockés qui ont été traités avec succès par le courtier de messages pour les clients déconnectés de leur session persistante. Les messages dont la QoS service est de 1 sont stockés lorsqu'un client avec une session persistante est déconnecté.
Queued.Throttle	Le nombre de messages qui n'ont pas pu être stockés et qui ont été limités alors que les clients ayant des sessions persistantes étaient déconnectés. Cela se produit lorsque les clients dépassent la limite de messages en file d'attente par seconde et par compte . Les messages dont la QoS service est de 1 sont stockés lorsqu'un client avec une session persistante est déconnecté.
Queued.ServerError	Le nombre de messages qui n'ont pas été stockés pour une session persistante en raison d'une erreur interne. Lorsque les clients disposant d'une session persistante sont déconnectés, les messages dont la qualité de service (QoS) est de 1 sont stockés.
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 AWS IoTQuotas (p. 1514) . 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.

Métrique	Description
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 AWS IoTQuotas (p. 1514) . La dimension Protocol contient le protocole utilisé pour envoyer le message UNSUBSCRIBE.
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, sous Métriques 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 de plus amples informations sur les métriques CloudWatch, veuillez consulter Métriques Amazon CloudWatch .) La dimension JobId contient l'ID de la tâche.
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 de plus amples informations sur les métriques CloudWatch, veuillez consulter 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 de plus amples informations sur les métriques CloudWatch, veuillez consulter 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 de plus amples informations sur les métriques CloudWatch, veuillez consulter 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.

Métrique	Description
RejectedJobExecutionCount	Nombre d'exécutions de tâche dont le statut est passé à REJECTED au cours d'une période définie par CloudWatch. (Pour de plus amples informations sur les métriques CloudWatch, veuillez consulter 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 de plus amples informations sur les métriques CloudWatch, veuillez consulter Métriques Amazon CloudWatch .) La dimension JobId contient l'ID de la tâche.
ServerErrorCount	Nombre d'erreurs de serveur générées pendant l'exécution de la tâche. La dimension JobId contient l'ID de la tâche.
SucceededJobExecutionCount	Nombre d'exécutions de tâche dont le statut est passé à SUCCESS au cours d'une période définie par CloudWatch. (Pour de plus amples informations sur les métriques CloudWatch, veuillez consulter Métriques Amazon CloudWatch .) La dimension JobId contient l'ID de la tâche.
SucceededJobExecutionTotalCount	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.

Métrique	Description
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 spécifique et pour un comportement spécifique d'un profil de sécurité spécifique.

Métriques de mise en service d'appareils

AWS IoT Métriques de provisionnement de la flotte

Métrique	Description
ApproximateNumberOfThingsRegistered	<p>Le nombre de choses qui ont été enregistrées par Fleet Provisioning.</p> <p>Bien que le décompte soit généralement précis, l'architecture distribuée de AWS IoT Core rend difficile le maintien d'un décompte précis des objets enregistrés.</p> <p>La statistique à utiliser pour cette métrique est la suivante :</p> <ul style="list-style-type: none"> • Maximum pour signaler le nombre total d'objets enregistrés. Pour connaître le nombre d'éléments enregistrés pendant la fenêtre CloudWatch d'agrégation, consultez la RegisterThingFailed métrique. <p>Dimensions : ClaimCertificateId (p. 489)</p>
CreateKeysAndCertificateFailed	<p>Le nombre d'échecs survenus lors d'appels à l'API CreateKeysAndCertificate MQTT.</p> <p>La métrique est émise à la fois en cas de réussite (valeur = 0) et en cas d'échec (valeur = 1). Cette métrique peut être utilisée pour suivre le nombre de certificats créés et enregistrés pendant les fenêtres d'agrégation CloudWatch prises en charge, par exemple 5 minutes ou 1 heure.</p> <p>Les statistiques disponibles pour cette métrique sont les suivantes :</p>

Métrique	Description
	<ul style="list-style-type: none"> • Somme pour signaler le nombre d'appels ayant échoué. • SampleCountpour signaler le nombre total d'appels réussis et échoués.
CreateCertificateFromCsrFailed	<p>Le nombre d'échecs survenus lors d'appels à l'API <code>CreateCertificateFromCsr</code> MQTT.</p> <p>La métrique est émise à la fois en cas de réussite (valeur = 0) et en cas d'échec (valeur = 1). Cette métrique peut être utilisée pour suivre le nombre d'éléments enregistrés pendant les fenêtres d'agrégation CloudWatch prises en charge, par exemple 5 minutes ou 1 heure.</p> <p>Les statistiques disponibles pour cette métrique sont les suivantes :</p> <ul style="list-style-type: none"> • Somme pour signaler le nombre d'appels ayant échoué. • SampleCountpour signaler le nombre total d'appels réussis et échoués.
RegisterThingFailed	<p>Le nombre d'échecs survenus lors d'appels à l'API <code>RegisterThing</code> MQTT.</p> <p>La métrique est émise à la fois en cas de réussite (valeur = 0) et en cas d'échec (valeur = 1). Cette métrique peut être utilisée pour suivre le nombre d'éléments enregistrés pendant les fenêtres d'agrégation CloudWatch prises en charge, par exemple 5 minutes ou 1 heure. Pour le nombre total d'objets enregistrés, consultez la <code>ApproximateNumberOfThingsRegistered</code> métrique.</p> <p>Les statistiques disponibles pour cette métrique sont les suivantes :</p> <ul style="list-style-type: none"> • Somme pour signaler le nombre d'appels ayant échoué. • SampleCountpour signaler le nombre total d'appels réussis et échoués. <p>Dimensions : TemplateName (p. 489)</p>

Métriques de `ust-in-time` provisionnement J

Métrique	Description
<code>ProvisionThing.ClientError</code>	Le nombre de fois où un appareil n'a pas pu être approvisionné en raison d'une erreur du client. Par exemple, la politique spécifiée dans le modèle n'existe pas.
<code>ProvisionThing.ServerError</code>	Le nombre de fois où un appareil n'a pas pu être approvisionné en raison d'une erreur du serveur. Les clients peuvent réessayer de provisionner l'appareil

Métrique	Description
	après avoir attendu et ils peuvent contacter AWS IoT si le problème persiste.
ProvisionThing.Success	Le nombre de fois qu'un appareil a été approvisionné avec succès.

Métriques d'indexation de la flotte

AWS IoT Métriques d'indexation de la flotte

Métrique	Description
NamedShadowCountForDynamicGroupQuery	Un maximum de 25 ombres nommées par objet sont traitées pour les termes de requête qui ne sont pas spécifiques à une source de données dans les groupes d'objets dynamiques. Lorsque cette limite est dépassée pour un objet, le type NamedShadowCountForDynamicGroupQueryLimitExceeded d'événement est émis.

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. 531) spécifié par la règle déclenchée par la demande.
BehaviorName	Nom du comportement du profil de sécurité de détection Device Defender qui est surveillé.
ClaimCertificateId	La certificateId partie de la réclamation utilisée pour fournir les appareils.
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.

Dimension	Description
TemplateName	Nom du modèle de mise en service.

Surveiller AWS IoT à l'aide CloudWatch des journaux

Lorsque la [journalisation AWS IoT est activée \(p. 467\)](#), 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. Dans la [CloudWatchconsole](#), CloudWatch les journaux apparaissent dans un groupe de journaux nommé AWSLogs.

Pour plus d'informations sur les CloudWatch journaux, consultez la section [CloudWatchJournaux](#). Pour plus d'informations sur les AWS IoT CloudWatch journaux pris en charge, consultez[CloudWatch AWS IoT entrées du journal \(p. 491\)](#).

Affichage des journaux AWS IoT dans la console CloudWatch

Note

Le groupe de AWSIotLogsV2 journaux n'est pas visible dans la CloudWatch console tant que :

- Vous avez activé la connexionAWS IoT. Pour plus d'informations sur la façon d'activer la connexionAWS IoT, voir [Configurer la journalisation AWS IoT \(p. 467\)](#)
- Certaines entrées du journal ont été écrites par AWS IoT des opérations.

Vous pouvez afficher vos journaux AWS IoT dans la console CloudWatch

1. Accédez à <https://console.aws.amazon.com/cloudwatch/>. Dans le panneau de navigation, choisissez 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. Choisissez Rechercher 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 plus d'informations sur la création d'expressions de filtre, veuillez consulter [CloudWatchLogs et requêtes](#).

CloudWatch AWS IoT entrées 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 AWS IoT composants suivants. Pour plus d'informations sur AWS IoT Core la surveillance du LoRa WAN, reportez-vous à la section [Afficher les entrées du CloudWatch AWS IoT Wireless journal \(p. 1464\)](#).

Rubriques

- [Entrées du journal du courtier de messages \(p. 491\)](#)
- [Entrées de journal de shadow d'appareil \(p. 499\)](#)
- [Entrées de journal du moteur de règles \(p. 501\)](#)
- [Entrées du journal des tâches \(p. 505\)](#)
- [Entrées de journal de provisionnement des appareils \(p. 509\)](#)
- [Entrées dynamiques du journal des groupes d'objets \(p. 510\)](#)
- [entrée de entrée de la flotte \(p. 511\)](#)
- [Attributs courants CloudWatch des journaux \(p. 512\)](#)

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 \(p. 491\)](#)
- [Déconnecter l'entrée du journal \(p. 492\)](#)
- [Entrée de journal GetRetainedMessage \(p. 493\)](#)
- [Entrée de journal ListRetainedMessage \(p. 493\)](#)
- [Entrée de journal Publish-In \(p. 494\)](#)
- [Entrée du journal Publish-Out \(p. 495\)](#)
- [entrée de entrée de entrée de entrée de entrée de entrée \(p. 495\)](#)
- [Souscrire l'entrée de journal \(p. 497\)](#)

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 courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer 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,  
    "reason": "DUPLICATE_CLIENT_ID",  
    "details": "A new connection was established with the same client ID",  
    "disconnectReason": "CLIENT_INITIATED_DISCONNECT"  
}
```

En plus de [Attributs courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer 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.

reason

La raison pour laquelle le client se déconnecte.

détails

Une brève explication de l'erreur.

disconnectReason

La raison pour laquelle le client se déconnecte.

Entrée de journal GetRetainedMessage

Le courtier de AWS IoT messages génère une entrée de journal avec eventType la GetRetainedMessage date à laquelle il [GetRetainedMessage](#) est appelé.

[GetRetainedMessage](#) entrée de entrée de entrée de entrée

```
{  
    "timestamp": "2017-08-07 18:47:56.664",  
    "logLevel": "INFO",  
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",  
    "accountId": "123456789012",  
    "status": "Success",  
    "eventType": "GetRetainedMessage",  
    "protocol": "HTTP",  
    "topicName": "a/b/c",  
    "qos": "1",  
    "lastModifiedDate": "2017-08-07 18:47:56.664"  
}
```

En plus de [Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal GetRetainedMessage contiennent les attributs suivants :

lastModifiedDate

Date et heure d'Epoch, en millisecondes, auxquelles le message conservé a été stocké. AWS IoT protocole ;

Protocole utilisé pour effectuer la demande. Valeur valide : HTTP.

qos

Le niveau de qualité de service (QoS) utilisé dans la demande de publication. Les valeurs valides sont 0 ou 1.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Entrée de journal ListRetainedMessage

Le courtier de AWS IoT messages génère une entrée de journal avec eventType la ListRetainedMessage date à laquelle il [ListRetainedMessages](#) est appelé.

ListRetainedMessageentrée de entrée de entrée

```
{  
    "timestamp": "2017-08-07 18:47:56.664",  
    "logLevel": "INFO",  
    "traceId": "1a60d02e-15b9-605b-7096-a9f584a6ad3f",  
    "accountId": "123456789012",  
    "status": "Success",  
    "eventType": "ListRetainedMessage",  
    "protocol": "HTTP"  
}
```

Outre le [Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées du ListRetainedMessage journal contiennent l'attribut suivant :

protocole ;

Protocole utilisé pour effectuer la demande. Valeur valide : HTTP.

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,  
    "retain": "True"  
}
```

En plus de [Attributs courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer la demande. Les valeurs valides sont MQTT ou HTTP.

conserver

Attribut utilisé lorsque l'indicateur RETAIN d'un message est défini avec une valeur de True. Si l'indicateur RETAIN n'est pas activé dans le message, cet attribut n'apparaît pas dans l'entrée du journal. Pour plus d'informations, veuillez consulter [MQTT retenus \(p. 97\)](#).

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 courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal Publish-Out contiennent les attributs suivants :

clientId

L'ID du client abonné qui reçoit des messages sur ce sujet MQTT.

principalId

L'ID du mandataire formulant la demande.

protocole ;

Protocole utilisé pour effectuer 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 de entrée de entrée de entrée de entrée de entrée

Lorsqu'un appareil doté d'une session persistante est déconnecté, le courtier de messages MQTT stocke les messages de l'appareil et AWS IoT génère des entrées de journal avec un eventType de Queued

Pour de plus amples informations sur les sessions persistantes MQTT, consultez[Sessions permanentes MQTT \(p. 94\)](#).

Exemple d'entrée dans le journal des erreurs du serveur en file d'attente

```
{  
    "timestamp": "2022-08-10 15:39:30.961",  
    "logLevel": "ERROR",  
    "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",  
    "accountId": "123456789012",  
    "topicName": "$aws/things/MyThing/get",  
    "clientId": "123123123",  
    "qos": "1",  
    "protocol": "MQTT",  
    "eventType": "Queued",  
    "status": "Failure",  
    "details": "Server Error"  
}
```

Outre les[Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées du journal des erreurs Queued du serveur contiennent les attributs suivants :

clientId

L'ID du client auquel le message est mis en file d'attente.

détails

Server Error

Une erreur du serveur a empêché le stockage du message.

protocole ;

Protocole utilisé pour effectuer la demande. La valeur sera toujours MQTT.

qos

Le niveau de qualité de service (QoS) de la demande. La valeur sera toujours 1 car les messages avec une QoS de 0 ne sont pas stockés.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Exemple d'entrée dans le journal de réussite en file d'attente

```
{  
    "timestamp": "2022-08-10 15:39:30.961",  
    "logLevel": "INFO",  
    "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",  
    "accountId": "123456789012",  
    "topicName": "$aws/things/MyThing/get",  
    "clientId": "123123123",  
    "qos": "1",  
    "protocol": "MQTT",  
    "eventType": "Queued",  
    "status": "Success"  
}
```

Outre les[Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées du journal de Queueed réussite contiennent les attributs suivants :

clientId

L'ID du client auquel le message est mis en file d'attente.

protocole ;

Protocole utilisé pour effectuer la demande. La valeur sera toujours MQTT.

qos

Le niveau de qualité de service (QoS) de la demande. La valeur sera toujours 1 car les messages avec une QoS de 0 ne sont pas stockés.

topicName

Le nom de la rubrique faisant l'objet de l'abonnement.

Exemple d'entrée de journal limitée en file d'attente

```
{  
    "timestamp": "2022-08-10 15:39:30.961",  
    "logLevel": "ERROR",  
    "traceId": "672ec480-31ce-fd8b-b5fb-22e3ac420699",  
    "accountId": "123456789012",  
    "topicName": "$aws/things/MyThing/get",  
    "clientId": "123123123",  
    "qos": "1",  
    "protocol": "MQTT",  
    "eventType": "Queued",  
    "status": "Failure",  
    "details": "Throttled while queueing offline message"  
}
```

Outre les [Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées Queued de journal limitées contiennent les attributs suivants :

clientId

L'ID du client auquel le message est mis en file d'attente.

détails

Throttled while queueing offline message

Le client ayant dépassé la [Queued messages per second per account](#) limite, le message n'a pas été enregistré.

protocole ;

Protocole utilisé pour effectuer la demande. La valeur sera toujours MQTT.

qos

Le niveau de qualité de service (QoS) de la demande. La valeur sera toujours 1 car les messages avec une QoS de 0 ne sont pas stockés.

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 dans le journal des abonnements MQTT 3

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

`protocol` ;

Protocole utilisé pour effectuer 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.

Exemple d'entrée dans le journal des abonnements MQTT 5

```
{  
    "timestamp": "2022-11-30 16:24:15.628",  
    "logLevel": "INFO",  
    "traceId": "7aa5c38d-1b49-3753-15dc-513ce4ab9fa6",  
    "accountId": "123456789012",  
    "status": "Success",  
    "eventType": "Subscribe",  
    "protocol": "MQTT",  
    "topicName": "test/topic1,$invalid/reserved/topic",  
    "subscriptions": [  
        {  
            "topicName": "test/topic1",  
            "reasonCode": 1  
        },  
        {  
            "topicName": "$invalid/reserved/topic",  
            "reasonCode": 143  
        }  
    ]  
}
```

```
],
"clientId": "abf27092886e49a8a5c1922749736453",
"principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
"sourceIp": "205.251.233.181",
"sourcePort": 13490
}
```

Pour les opérations d'abonnement à MQTT 5, outre les [attributs d'entrée du Attributs courants CloudWatch des journaux \(p. 512\)](#) [journal d'abonnement de MQTT 3](#), les [entrées de journal \(p. 498\)](#) de MQTT 5 Subscribe contiennent l'attribut suivant :

abonnements

Une liste de correspondances entre les sujets demandés dans la demande d'abonnement et le code de raison individuel du MQTT 5. Pour plus d'informations, consultez la section [Codes de raison MQTT](#).

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 \(p. 499\)](#)
- [Entrée de journal GetThingShadow \(p. 500\)](#)
- [Entrée de journal UpdateThingShadow \(p. 500\)](#)

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.

DeleteThingShadowentrée de entrée de entrée

```
{
  "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 courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal DeleteThingShadow contiennent les attributs suivants :

deviceShadowName

Nom du shadow à mettre à jour.

protocole ;

Protocole utilisé pour effectuer 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.

GetThingShadowentrée de entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal GetThingShadow contiennent les attributs suivants :

deviceShadowName

Le nom du shadow demandé.

protocole ;

Protocole utilisé pour effectuer 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.

UpdateThingShadowentrée de entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal UpdateThingShadow contiennent les attributs suivants :

deviceShadowName

Nom du shadow à mettre à jour.

protocole ;

Protocole utilisé pour effectuer 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 \(p. 501\)](#)
- [Entrée de journal RuleExecution \(p. 502\)](#)
- [Entrée de journal RuleMatch \(p. 502\)](#)
- [Entrée de journal RuleExecutionThrottled \(p. 503\)](#)
- [Entrée de journal RuleNotFound \(p. 504\)](#)
- [Entrée de journal StartingRuleExecution \(p. 505\)](#)

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).

FunctionExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

RuleExecutionentrée de entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

RuleMatch entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 RuleExecutionThrottled

Lorsqu'une exécution est limitée, le moteur de AWS IoT règles génère une entrée de journal avec un eventType de RuleExecutionThrottled

RuleExecutionThrottled entrée de entrée de entrée

```
{  
    "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": "Exection of Rule example_rule throttled"  
}
```

En plus de [Attributs courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal RuleExecutionThrottled 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 « RuleExecutionThrottled ».

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.

RuleNotFoundentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

StartingRuleExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 \(p. 506\)](#)
- [Entrée de journal GetPendingJobExecution \(p. 506\)](#)
- [Entrée de journal ReportFinalJobExecutionCount \(p. 507\)](#)
- [Entrée de journal StartNextPendingJobExecution \(p. 508\)](#)

- [Entrée de journal UpdateJobExecution \(p. 508\)](#)

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.

DescribeJobExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 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 ;

Protocole utilisé pour effectuer 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.

GetPendingJobExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer 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

Le service AWS IoT Jobs génère une entrée de journal avec un entryType de ReportFinalJobExecutionCount lorsqu'une tâche est terminée.

ReportFinalJobExecutionCount entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

StartNextPendingJobExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer 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.

UpdateJobExecutionentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 ;

Protocole utilisé pour effectuer 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 \(p. 509\)](#)
- [Entrée de journal ProvisionDevice \(p. 510\)](#)

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.

GetDeviceCredentialsentrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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.

ProvisionDeviceentrée de entrée de entrée de entrée

```
{  
    "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 courants CloudWatch des journaux \(p. 512\)](#), 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 dynamiques AWS IoT génèrent des journaux pour l'événement suivant.

Rubriques

- [Entrée de journal AddThingToDynamicThingGroupsFailed \(p. 510\)](#)

Entrée de journal AddThingToDynamicThingGroupsFailed

Lorsque AWS IoT n'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 \(p. 307\)](#).

AddThingToDynamicThingGroupsFailedentrée de entrée de entrée

Cet exemple montre une entrée de journal correspondant à une erreur AddThingToDynamicThingGroupsFailed. Dans cet exemple, TestThingrépondait aux critères pour figurer dans les groupes d'objets dynamiques répertoriés dansdynamicThingGroupNames, mais n'a pas pu être ajouté à ces groupes dynamiques, comme décrit dansreason.

```
{  
  "timestamp": "2020-03-16 22:24:43.804",  
  "logLevel": "ERROR",  
  "traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",  
  "accountId": "57EXAMPLE833",  
  "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 courants CloudWatch des journaux \(p. 512\)](#), les entrées de journal AddThingToDynamicThingGroupsFailed contiennent les attributs suivants :

dynamicThingGroupNoms

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.

entrée de entrée de la flotte

AWS IoT l'indexation du parc génère des entrées de journal pour les événements suivants.

Rubriques

- [Entrée de journal NamedShadowCountForDynamicGroupQueryLimitExceeded \(p. 511\)](#)

Entrée de journal NamedShadowCountForDynamicGroupQueryLimitExceeded

Un maximum de 25 ombres nommées par objet sont traitées pour les termes de requête qui ne sont pas spécifiques à une source de données dans les groupes dynamiques. Lorsque cette limite est dépassée

pour un objet, le type `NamedShadowCountForDynamicGroupQueryLimitExceeded` d'événement est émis.

[NamedShadowCountForDynamicGroupQueryLimitExceeded](#) exemple d'entrée de journal

Cet exemple montre l'entrée d'une `NamedShadowCountForDynamicGroupQueryLimitExceeded` erreur dans le journal. Dans cet exemple, les `DynamicGroup` résultats basés sur toutes les valeurs peuvent être inexacts, comme décrit dans le `reason` champ.

```
{  
  "timestamp": "2020-03-16 22:24:43.804",  
  "logLevel": "ERROR",  
  "traceId": "70b1f2f5-d95e-f897-9dcc-31e68c3e1a30",  
  "accountId": "571032923833",  
  "status": "Failure",  
  "eventType": "NamedShadowCountForDynamicGroupQueryLimitExceeded",  
  "thingName": "TestThing",  
  "reason": "A maximum of 25 named shadows per thing are processed for non-data source  
  specific query terms in dynamic groups."  
}
```

Attributs courants CloudWatch des journaux

Toutes les entrées du journal des CloudWatch journaux incluent les attributs suivants :

`accountId`

Votre Compte AWS carte d'identité.

`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, veuillez consulter [the section called "Niveaux de journalisation" \(p. 474\)](#).

`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.

Importer les journaux côté appareil sur Amazon CloudWatch

Vous pouvez charger des journaux historiques côté appareil sur Amazon CloudWatch pour surveiller et analyser l'activité d'un appareil sur le terrain. Les journaux côté appareil peuvent inclure des fichiers journaux du système, des applications et des appareils. [Ce processus utilise un paramètre d'action CloudWatch Logs rules pour publier les journaux côté appareil dans un groupe de journaux défini par le client.](#)

Comment ça marche

Le processus commence lorsqu'un AWS IoT appareil envoie des messages MQTT contenant des fichiers journaux formatés à une AWS IoT rubrique. Une AWS IoT règle surveille le sujet du message et envoie les fichiers journaux à un groupe de CloudWatch journaux que vous définissez. Vous pouvez ensuite passer en revue et analyser les informations.

Rubriques

- [Rubriques du MQTT \(p. 513\)](#)
- [Action de la règle \(p. 513\)](#)

Rubriques du MQTT

Choisissez un espace de nom de rubrique MQTT que vous utiliserez pour publier les journaux. Nous vous recommandons d'utiliser ce format pour l'espace réservé aux rubriques communes et ce format pour les rubriques contenant des erreurs \$aws/rules/things/*thing_name*/logs/errors. \$aws/rules/things/*thing_name*/logs La structure de dénomination pour les journaux et les rubriques d'erreur est recommandée, mais pas obligatoire. Pour de plus amples informations, veuillez consulter [Conception de rubrique MQTT pour AWS IoT Core](#).

En utilisant l'espace réservé aux rubriques communes recommandé, vous utilisez les rubriques réservées AWS IoT de Basic Ingest. AWS IoT Basic Ingest envoie en toute sécurité les données des appareils aux AWS services pris en charge par les actions des AWS IoT règles. Il supprime le courtier de messages de publication/d'abonnement du chemin d'ingestion, ce qui le rend plus rentable. Pour plus d'informations, consultez [la section Réduction des coûts de messagerie grâce à Basic Ingest](#).

Si vous utilisez BatchMode pour charger des fichiers journaux, vos messages doivent respecter un format spécifique qui inclut un horodatage et un message UNIX. Pour plus d'informations, consultez la rubrique relative [aux exigences relatives au format de message MQTT pour BatchMode](#) dans la section Actions relatives aux règles de [CloudWatchjournalisation](#).

Action de la règle

Lorsque AWS IoT vous recevez les messages MQTT des appareils clients, une AWS IoT règle surveille le sujet défini par le client et publie le contenu dans un groupe de CloudWatch journaux que vous définissez. Ce processus utilise une action de règle CloudWatch Logs pour surveiller MQTT pour détecter des lots de fichiers journaux. Pour plus d'informations, consultez l'action relative à la AWS IoT règle [CloudWatchLogs](#).

Mode Batch

batchMode est un paramètre booléen intégré à l'action de la règle AWS IoT CloudWatch Logs. Ce paramètre est facultatif et est désactivé (`false`) par défaut. Pour charger des fichiers journaux côté appareil par lots, vous devez activer ce paramètre (`true`) lorsque vous créez la règle. AWS IoT Pour plus d'informations, consultez la section [CloudWatchJournaux](#) dans la section [des actions relatives aux AWS IoT règles](#).

Chargement des journaux côté appareil à l'aide de règles AWS IoT

Vous pouvez utiliser le moteur de AWS IoT règles pour charger des enregistrements de journaux à partir de fichiers journaux existants côté appareil (journaux système, applications et appareils clients) vers Amazon. CloudWatch Lorsque les journaux côté appareil sont publiés dans une rubrique MQTT, l'action Règles des CloudWatch journaux transfère les messages vers les journaux. CloudWatch Ce processus explique comment charger les journaux des appareils par lots à l'aide du batchMode paramètre d'action des règles activé (défini sur `true`).

Pour commencer à charger des journaux côté appareilCloudWatch, vous devez remplir les conditions préalables suivantes.

Prérequis

Avant de commencer, vous devez exécuter les actions suivantes :

- Créez au moins un appareil IoT cible enregistré en AWS IoT Core tant qu'AWS IoT object. Pour de plus amples informations, veuillez consulter [Création d'un objet](#).
- Déterminez l'espace thématique du MQTT relatif à l'ingestion et aux erreurs. Pour plus d'informations sur les rubriques MQTT et les conventions de dénomination recommandées, consultez la section [Rubriques Rubriques du MQTT \(p. 513\) MQTT](#) dans [Charger des journaux côté appareil sur Amazon CloudWatch](#).

Pour plus d'informations sur ces conditions préalables, veuillez consulter [Charger des journaux côté appareil dans CloudWatch](#)

Création d'un groupe de CloudWatch journaux

Pour créer un groupe de CloudWatch journaux, effectuez les opérations suivantes. Choisissez l'onglet approprié selon que vous préférez effectuer les étapes par le biais du AWS Management Console ou du AWS Command Line Interface (AWS CLI).

AWS Management Console

Pour créer un groupe de CloudWatch journaux à l'aide du AWS Management Console

1. Ouvrez AWS Management Console et accédez à [CloudWatch](#).
2. Dans la barre de navigation, choisissez Journaux, puis Groupes de journaux.
3. Sélectionnez Créer un groupe de journaux.
4. Mettez à jour le nom du groupe de journaux et, éventuellement, mettez à jour les champs des paramètres de rétention.
5. Sélectionnez Create (Créer).

AWS CLI

Pour créer un groupe de CloudWatch journaux à l'aide du AWS CLI

1. Pour créer le groupe de journaux, exécutez la commande suivante. Pour de plus amples informations, [create-log-group](#) veuillez consulter la référence des commandes AWS CLI v2.

Remplacez le nom du groupe de journaux dans l'exemple (`uploadLogsGroup`) par votre nom préféré.

```
aws logs create-log-group --log-group-name uploadLogsGroup
```

2. Pour confirmer que le groupe de journaux a bien été créé, exécutez la commande suivante.

```
aws logs describe-log-groups --log-group-name-prefix uploadLogsGroup
```

Exemple de sortie :

```
{
```

```
"logGroups": [
    {
        "logGroupName": "uploadLogsGroup",
        "creationTime": 1674521804657,
        "metricFilterCount": 0,
        "arn": "arn:aws:logs:us-east-1:111122223333:log-
group:uploadLogsGroup:*",
        "storedBytes": 0
    }
]
```

Création d'une règle de entrée de rubrique

Pour créer une AWS IoT règle, procédez comme suit pour créer une règle, procédez comme suit pour créer une règle. Choisissez l'onglet approprié selon que vous préférez effectuer les étapes par le biais du AWS Management Console ou du AWS Command Line Interface (AWS CLI).

AWS Management Console

Pour créer une règle de sujet à l'aide du AWS Management Console

1. Ouvrez le hub de règles.
 - a. Ouvrez le AWS Management Console et accédez à [AWS IoT](#).
 - b. Dans la barre de navigation, choisissez Routage des messages, puis Règles.
 - c. Choisissez Create rule (Créer une règle).
2. Entrez les propriétés de la règle.
 - a. Entrez un nom de règle alphanumérique.
 - b. (Facultatif) Entrez une description de la règle et des balises.
 - c. Choisissez Suivant.
3. Entrée de entrée de entrée de entrée de entrée
 - a. Entrez une instruction SQL à l'aide de la rubrique MQTT que vous avez définie pour l'ingestion.
Par exemple, `SELECT * FROM '$aws/rules/things/thing_name/logs'`
 - b. Choisissez Suivant.
4. Entrez les actions des règles.
 - a. Dans le menu Action 1, choisissez CloudWatchjournals.
 - b. Choisissez le nom du groupe de journaux et choisissez le groupe de journaux que vous avez créé.
 - c. Sélectionnez Utiliser le mode batch.
 - d. Spécifiez le rôle IAM pour la règle.

Si vous avez un rôle IAM pour la règle, procédez comme suit pour la règle.

1. Dans le menu du rôle IAM, choisissez votre rôle IAM.

Si vous n'avez pas de rôle IAM pour la règle, procédez comme suit pour la règle.

1. Choisissez Créer un nouveau rôle.
2. Pour le nom du rôle, entrez un nom unique et choisissez Créer.

3. Vérifiez que le nom du rôle IAM est correct dans le champ du rôle IAM.
- e. Choisissez Suivant.
5. Vérifier la configuration du modèle.
 - a. Vérifiez les paramètres du modèle de Job pour vérifier qu'ils sont corrects.
 - b. Lorsque vous avez terminé, cliquez sur Create (Créer).

AWS CLI

Pour créer un rôle IAM et une règle de entrée à l'aide du rôle IAM et une règle de entrée à l'aide du AWS CLI

1. Créez un rôle IAM qui accorde des droits à la AWS IoT règle.

- a. Créez une politique IAM.

Pour créer une politique IAM, exécutez la commande suivante. Assurez-vous de mettre à jour la valeur du `policy-name` paramètre. Pour plus d'informations, consultez [create-policy](#) la Référence des commandes AWS CLI v2.

Note

Si vous utilisez un système d'exploitation Microsoft Windows, vous devrez peut-être remplacer le marqueur de fin de ligne (\) par une coche (') ou un autre caractère.

```
aws iam create-policy \
--policy-name uploadLogsPolicy \
--policy-document \
'{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:CreateTopicRule",
                "iot:Publish",
                "logs>CreateLogGroup",
                "logs>CreateLogStream",
                "logs:PutLogEvents",
                "logs:GetLogEvents"
            ],
            "Resource": "*"
        }
    ]
}'
```

- b. Copiez l'ARN de la politique depuis votre sortie dans un éditeur de texte.

Exemple de sortie :

```
{
    "Policy": {
        "PolicyName": "uploadLogsPolicy",
        "PermissionsBoundaryUsageCount": 0,
        "CreateDate": "2023-01-23T18:30:10Z",
        "AttachmentCount": 0,
        "IsAttachable": true,
        "PolicyId": "AAABBBCCDDDEEEFFFGGG",
        "DefaultVersionId": "v1",
        "Path": "/",
        "Arn": "arn:aws:iam::111122223333:policy/uploadLogsPolicy",
        "UpdateDate": "2023-01-23T18:30:10Z"
```

```

    }
}
```

- c. Création d'un rôle IAM et d'une politique d'approbation.

Pour créer une politique IAM, exécutez la commande suivante. Assurez-vous de mettre à jour la valeur du `role-name` paramètre. Pour plus d'informations, consultez [create-role](#) Référence des commandes AWS CLI v2.

```

aws iam create-role \
--role-name uploadLogsRole \
--assume-role-policy-document \
'{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "iot.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}'
```

- d. Attachez la politique IAM à la règle à la règle.

Pour créer une politique IAM, exécutez la commande suivante. Assurez-vous de mettre à jour les valeurs `role-name` des `policy-arn` paramètres et. Pour plus d'informations, consultez [attach-role-policy](#) la référence des commandes AWS CLI v2.

```

aws iam attach-role-policy \
--role-name uploadLogsRole \
--policy-arn arn:aws:iam::111122223333:policy/uploadLogsPolicy
```

- e. Vérifier le rôle.

Pour confirmer que le rôle IAM a bien été créé, exécutez la commande suivante. Assurez-vous de mettre à jour la valeur du `role-name` paramètre. Pour plus d'informations, consultez [get-role](#) la Référence des commandes AWS CLI v2.

```
aws iam get-role --role-name uploadLogsRole
```

Exemple de sortie :

```
{
    "Role": {
        "Path": "/",
        "RoleName": "uploadLogsRole",
        "RoleId": "AAABBBCCDDDEEEFFFGGG",
        "Arn": "arn:aws:iam::111122223333:role/uploadLogsRole",
        "CreateDate": "2023-01-23T19:17:15+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Sid": "Statement1",
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "iot.amazonaws.com"
```

```
        },
        "Action": "sts:AssumeRole"
    ]
},
"Description": "",
"MaxSessionDuration": 3600,
"RoleLastUsed": {}
}
}
```

2. Créez une règle de AWS IoT sujet dans le AWS CLI.

- Pour créer une règle de entrée de AWS IoT rubrique, exécutez la commande suivante. Assurez-vous de mettre à jour les valeurs de l'sqlinstruction --rule-name descriptionroleARN , et des logGroupName paramètres. Pour plus d'informations, consultez [create-topic-rule](#) la référence des commandes AWS CLI v2.

```
aws iot create-topic-rule \
--rule-name uploadLogsRule \
--topic-rule-payload \
'{
  "sql":"SELECT * FROM 'rules/things/thing_name/logs'",
  "description":"Upload logs test rule",
  "ruleDisabled":false,
  "awsIotSqlVersion":"2016-03-23",
  "actions":[
    {"cloudwatchLogs":
      {"roleArn":"arn:aws:iam::111122223333:role/uploadLogsRole",
       "logGroupName":"uploadLogsGroup",
       "batchMode":true}
    }
  ]
}'
```

- Pour confirmer que la règle a bien été créée, exécutez la commande suivante. Assurez-vous de mettre à jour la valeur du role-name paramètre. Pour plus d'informations, consultez [get-topic-rule](#) la Référence des commandes AWS CLI v2.

```
aws iot get-topic-rule --rule-name uploadLogsRule
```

Exemple de sortie :

```
{
  "ruleArn": "arn:aws:iot:us-east-1:111122223333:rule/uploadLogsRule",
  "rule": {
    "ruleName": "uploadLogsRule",
    "sql": "SELECT * FROM rules/things/thing_name/logs",
    "description": "Upload logs test rule",
    "createdAt": "2023-01-24T16:28:15+00:00",
    "actions": [
      {
        "cloudwatchLogs": {
          "roleArn": "arn:aws:iam::111122223333:role/uploadLogsRole",
          "logGroupName": "uploadLogsGroup",
          "batchMode": true
        }
      }
    ],
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23"
  }
}
```

}

Envoi des journaux côté appareil à AWS IoT

Pour envoyer des journaux côté appareil à AWS IoT

1. Pour envoyer des journaux historiques à AWS IoT, communiquez avec vos appareils pour vous assurer de ce qui suit.
 - Les informations du journal sont envoyées à l'espace de noms de rubrique approprié, comme indiqué dans la section Prérequis de cette procédure.

Par exemple, `$aws/rules/things/thing_name/logs`

 - La charge utile du message MQTT est correctement formatée. Pour plus d'informations sur le sujet MQTT et la convention de dénomination recommandée, consultez la [Rubriques du MQTT \(p. 513\)](#) section ci-dessous. [Importer les journaux côté appareil sur Amazon CloudWatch \(p. 512\)](#)
2. Vérifiez que les messages MQTT sont reçus dans le client AWS IoT MQTT.
 - a. Ouvrez AWS Management Console et accédez à [AWS IoT](#).
 - b. Pour afficher le client de test MQTT, dans la barre de navigation, choisissez Test, client de test MQTT.
 - c. Pour S'abonner à une rubrique, Filtre de rubrique, entrez l'espace de nommage de la rubrique.
 - d. Choisissez Subscribe.

Les messages MQTT apparaissent dans le tableau des abonnements et des rubriques, comme indiqué ci-dessous. Ces messages peuvent mettre jusqu'à cinq minutes avant de s'afficher.

The screenshot shows the AWS IoT Core Publish interface. At the top, there are two tabs: "Subscribe to a topic" (grayed out) and "Publish to a topic" (highlighted in orange). Below the tabs, the "Topic name" field contains "topic/test/" and the "Message payload" field is empty. A "Publish" button is visible. In the main area, under "Subscriptions", there is a list with one item: "topic/test/" followed by a heart icon and a delete "X". To the right, the contents of the topic are displayed as a JSON array:

```
[{"timestamp": 1673520691123, "message": "Test message 1"}, {"timestamp": 1673520692321, "message": "Test message 2"}, {"timestamp": 1673520693322, "message": "Test message 3"}]
```

Affichage des données du journal

Pour consulter vos enregistrements de journal dans CloudWatch Logs

1. Ouvrez le AWS Management Console, puis accédez à [CloudWatch](#).
2. Dans la barre de navigation, choisissez Logs, Logs Insights.

3. Dans le menu Sélectionner un ou plusieurs groupes de journaux, choisissez le groupe de journaux que vous avez spécifié dans la AWS IoT règle.
4. Sur la page Logs insights, choisissez Exécuter une requête.

Journalisation des appels d'API AWS IoT avec AWS CloudTrail

AWS IoT est intégré avec AWS CloudTrail un service qui fournit un registre des actions prises par un utilisateur, un rôle ou un service AWS dans AWS IoT. CloudTrail capture tous les appels d'API pour AWS IoT en tant qu'événements, y compris les appels émis par la console AWS IoT et les appels de code transmis aux API AWS IoT. Si vous créez un journal d'activité, vous pouvez activer la livraison continue d'CloudTrail événements à un compartiment Amazon S3, y compris des événements pour AWS IoT. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Historique des événements. À partir des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été envoyée à AWS IoT, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur de la demande, sa date, ainsi que d'autres détails.

Pour en savoir plus sur CloudTrail, veuillez consulter le [Guide de l'utilisateur AWS CloudTrail](#).

Informations AWS IoT dans CloudTrail

CloudTrail est activé sur votre Compte AWS lorsque vous créez le compte. Lorsqu'une activité a lieu dans AWS IoT, cette activité est enregistrée dans un événement CloudTrail avec d'autres événements de service AWS dans Event history (Historique des événements). Vous pouvez afficher, rechercher et télécharger les événements récents dans votre Compte AWS. Pour plus d'informations, consultez [Affichage des événements avec l'historique des événements CloudTrail](#).

Pour un enregistrement continu des événements dans votre Compte AWS, y compris les événements pour AWS IoT, créez un journal d'activité. Un journal de suivi permet CloudTrail à un compartiment Amazon S3 de livrer des fichiers journaux vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions Région AWS. Le journal d'activité consigne les événements de toutes les régions Région AWS dans la partition AWS et transfère les fichiers journaux dans le compartiment Amazon S3 de votre choix. Vous pouvez configurer d'autres services AWS afin d'analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et d'agir sur celles-ci. Pour plus d'informations, reportez-vous à :

- [Présentation de la création d'un journal d'activité](#)
- [Intégrations et services supportés par CloudTrail](#)
- [Configuration des Notifications de Amazon SNS pour CloudTrail](#)
- [Réception de fichiers journaux CloudTrail de plusieurs régions](#) et [Réception de fichiers journaux CloudTrail de plusieurs comptes](#)

Note

Les actions de plan de données AWS IoT (côté appareil) ne sont pas consignées par CloudTrail. Utilisez CloudWatch pour surveiller ces actions.

D'une manière générale, les actions du plan de AWS IoT contrôlent qui apportent des modifications sont enregistrées CloudTrail. Les appels tels que CreateThingCreateKeysAndCertificate, et UpdateCertificate laissent des CloudTrail entrées, tandis que les appels tels que ListThingset ListTopicRules n'en laissent pas.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les autorisations utilisateur root ou IAM .
- Si la demande a été effectuée avec des autorisations de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour plus d'informations, consultez la section [Élément userIdentity CloudTrail](#).

AWS IoT Les actions sont documentées dans la [référence de AWS IoT l'API](#). AWS IoT Les actions sans fil sont documentées dans le [AWS IoTWireless API Reference](#).

Présentation des AWS IoT entrées des fichiers journaux

Un journal d'activité est une configuration qui permet d'envoyer les événements dans des fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Les fichiers journaux CloudTrail contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la requête, etc. Les fichiers journaux CloudTrail ne constituent pas une série 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": "8bfff4fed-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/528ce36e8047f6a75ee51ab7bedb4eb268ad41d2ea881a10b67e8e76924d894",  
        "policyName": "ExamplePolicyForIoT"  
    },  
    "Resources": "",  
    "ResponseElements": "",  
    "SourceIpAddress": "52.90.213.26",  
    "UserAgent": "aws-internal/3",  
    "UserIdentity": {  
        "type": "AssumedRole",  
        "principalId": "AKIAI44QH8DHBEEXAMPLE",  
        "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",  
                "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 permettent à vos appareils 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 effectuer les tâches suivantes :

- Augmenter ou filtrer les données reçues d'un appareil.
- Écrivez les données reçues d'un appareil dans une base de données Amazon DynamoDB.
- Enregistrer un fichier dans Amazon S3.
- Envoyez une notification push à tous les utilisateurs qui utilisent Amazon SNS.
- Publie des données dans une file Amazon SQS.
- Appeler une fonction Lambda pour extraire des données.
- Traitez les messages d'un grand nombre d'appareils avec Amazon Kinesis.
- Envoyez des données à Amazon OpenSearch Service.
- Saisir une métrique CloudWatch.
- Modifier une alarme CloudWatch.
- Envoyez les données d'un message MQTT à Amazon SageMaker pour établir des prévisions basées sur un modèle d'apprentissage automatique (ML).
- Envoyer un message à un flux d'entrée Salesforce IoT
- Envoyer des données de message à un canal AWS IoT Analytics.
- Lance du processus d'une machine d'état Step Functions.
- Envoyer des données de message à une entrée AWS IoT Events.
- Envoyer les données d'un message à une propriété de ressource 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 le. [the section called "Protocoles de communication des appareils" \(p. 89\)](#) Vous pouvez également utiliser la fonctionnalité Basic Ingest (p. 617) pour envoyer en toute sécurité les données de l'appareil aux adresses Services AWS répertoriées précédemment, sans encourir de frais de messagerie. La fonctionnalité Basic Ingest (p. 617) optimise le flux de données en supprimant le courtier de messages de publication/d'abonnement du chemin d'ingestion. Cela le rend rentable tout en conservant les fonctionnalités de sécurité et de traitement des données de AWS IoT.

Avant de AWS IoT pouvoir effectuer ces actions, vous devez lui accorder l'autorisation d'accéder à vos AWS ressources en votre nom. Lorsque les actions sont effectuées, vous devez payer les frais standard pour ceux Services AWS que vous utilisez.

Table des matières

- [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#)
- [Transmettre les autorisations de rôle \(p. 526\)](#)
- [Création d'une règle AWS IoT \(p. 527\)](#)
- [Affichage des règles \(p. 531\)](#)
- [Suppression d'une règle \(p. 531\)](#)
- [Actions de règle AWS IoT \(p. 531\)](#)
- [Résolution des problèmes d'une règle \(p. 609\)](#)
- [Accès aux ressources multicomptes à l'aide AWS IoT de règles \(p. 609\)](#)
- [Gestion des erreurs \(action d'erreur\) \(p. 615\)](#)

- [Réduire les coûts de messagerie grâce à Basic Ingest \(p. 617\)](#)
- [Référence SQL AWS IoT \(p. 618\)](#)

Accorder à une AWS IoT règle l'accès dont elle a besoin

Utilisez les rôles IAM pour contrôler les AWS ressources auxquelles chaque règle a accès. Avant de créer une règle, vous devez créer un rôle IAM avec une stratégie qui permet d'accéder aux AWS ressources requises. AWS IoT assume ce rôle lors de la mise en œuvre d'une règle.

Procédez comme suit pour créer le rôle et la AWS IoT politique IAM qui accordent à une AWS IoT règle l'accès dont elle a besoin (AWS CLI).

1. Enregistrez le document de politique de confiance suivant, qui accorde AWS IoT l'autorisation d'assumer le rôle, dans un fichier nommé `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 politique qui permet aux AWS IoT administrateurs d'accéder 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 manière d'accorder l'accès Services AWS à des politiques d'intégration pour AWS IoT, consultez [Création d'une règle AWS IoT \(p. 527\)](#).

La sortie de la commande [create-policy](#) contient l'ARN de la stratégie. Attachez la politique à un rôle.

```
{  
  "Policy": {  
    "PolicyName": "my-iot-policy",  
    "CreateDate": "2015-09-30T19:31:18.620Z",  
    "AttachmentCount": 0,  
    "IsAttachable": true,  
    "PolicyId": "ZXR6A36LYANPAI7NJ5UV",  
    "DefaultVersionId": "v1",  
    "Path": "/",  
    "Arn": "arn:aws:iam::123456789012:policy/my-iot-policy",  
    "UpdateDate": "2015-09-30T19:31:18.620Z"  
  }  
}
```

3. Utilisez la [attach-role-policy](#) commande pour attacher votre politique 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 partie de la définition d'une règle comprend un rôle IAM qui accorde l'autorisation d'accéder aux ressources spécifiées dans l'action de la règle. Le moteur de règles assume ce rôle lorsque l'action de la règle est invoquée. Le rôle doit être défini de la même manière Compte AWS que la règle.

Lorsque vous créez ou remplacez une règle, vous transférez en fait un rôle au moteur de règles. L'iam:PassRole autorisation est requise pour effectuer cette opération. Pour vérifier que vous disposez de cette autorisation, créez une politique qui accorde l'iam:PassRole autorisation et associez-la 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 politique, l'autorisation `iam:PassRole` est accordée pour le rôle `myRole`. Le rôle a été spécifié à l'aide de son ARN. Attachez cette politique à votre utilisateur IAM ou au rôle auquel appartient votre utilisateur. Pour plus d'informations, consultez [Utilisation des politiques gérées](#).

Note

Les fonctions Lambda utilisent des politiques basées sur des ressources, où la politique est attachée directement à la fonction Lambda. Quand 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 d'autorisation. `iam:PassRole` Pour de plus amples informations sur l'autorisation de fonctions Lambda, veuillez consulter [Accord d'autorisations à l'aide d'une stratégie de ressources](#).

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

Le 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, veuillez consulter [Référence SQL AWS IoT \(p. 618\)](#).

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. La AWS IoT Core console définit cette propriété `2016-03-23` par défaut. Si cette propriété n'est pas définie, par exemple dans une AWS CLI commande ou un AWS CloudFormation modèle, elle `2015-10-08` est utilisée. Pour plus d'informations, veuillez consulter [Versions de SQL \(p. 688\)](#).

Une ou plusieurs actions

Les actions sont exécutées AWS IoT lors de la promulgation de la règle. Par exemple, vous pouvez insérer des données dans une table DynamoDB, écrire des données dans un compartiment Amazon S3, les publier dans une rubrique Amazon SNS ou appeler une fonction Lambda.

Une action d'erreur

L'action s'AWS IoT exécute lorsqu'elle n'est pas en mesure d'exécuter l'action d'une règle.

Lorsque vous créez une règle, tenez compte de la quantité de données que vous publiez sur des sujets. Si vous créez des règles qui incluent un modèle de sujet générique, elles peuvent correspondre à un

pourcentage élevé de vos messages. Dans ce cas, vous devrez peut-être augmenter la capacité des AWS ressources 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 [create-topic-rule](#) commande 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 la `iot/test` rubrique dans la table DynamoDB spécifiée. L'instruction SQL filtre les messages et le rôle ARN AWS IoT autorise l'écriture 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 AWS IoT autorise l'écriture 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 contenant une règle qui transmet des données à Amazon OpenSearch Service :

```
{
```

```
"sql": "SELECT * , timestamp() as timestamp FROM 'iot/test'",  
"ruleDisabled": false,  
"awsIotSqlVersion": "2016-03-23",  
"actions": [  
    {  
        "OpenSearch": {  
            "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 invoque 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 contenant 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 contenant une règle qui envoie des données vers 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 la SageMaker machinelearning_predict fonction Amazon pour republier dans une rubrique si les données de la charge utile MQTT sont classées comme 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 [list-topic-rules](#) commande pour répertorier vos règles :

```
aws iot list-topic-rules
```

Utilisez la [get-topic-rule](#) commande 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 [delete-topic-rule](#) commande pour supprimer une règle :

```
aws iot delete-topic-rule --rule-name myrule
```

Actions de règle AWS IoT

AWS IoTLes actions des règles indiquent ce qu'il faut faire lorsqu'une règle est invoquée. Vous pouvez définir des actions pour envoyer des données vers une base de données Amazon DynamoDB, envoyer des données vers Amazon Kinesis Data Streams, invoquer une AWS Lambda fonction, etc. AWS IoT prend en charge les actions suivantes Régions AWS lorsque le service de l'action est disponible.

Action de la règle	Description	Nom dans l'API
Apache Kafka (p. 533)	Envoie un message à un cluster Apache Kafka.	kafka
Alarmes CloudWatch (p. 541)	Modifie l'état d'une CloudWatch alarme Amazon.	cloudwatchAlarm
Journaux CloudWatch (p. 542)	Envoie un message à Amazon CloudWatch Logs.	cloudwatchLogs
Métriques CloudWatch (p. 544)	Envoie un message à une CloudWatch métrique.	cloudwatchMetric
DynamoDB (p. 546)	Envoie un message à une table DynamoDB.	dynamoDB
DynamoDBv2 (p. 548)	Envoie les données des messages à plusieurs colonnes d'une table DynamoDB.	dynamoDBv2
Elasticsearch (p. 549)	Envoie un message à un OpenSearch point de terminaison.	OpenSearch

Action de la règle	Description	Nom dans l'API
HTTP (p. 551)	Publie un message sur un point de terminaison sur un point de terminaison sur	http
IoT Analytics (p. 578)	Envoie un message à une AWS IoT Analytics chaîne.	iotAnalytics
AWS IoT Events (p. 580)	Envoie un message à une AWS IoT Events entrée.	iotEvents
AWS IoT SiteWise (p. 582)	Envoie des données de message aux propriétés de la AWS IoT SiteWise ressource.	iotSiteWise
Kinesis Data Firehose (p. 586)	Envoie un message à un flux de diffusion Kinesis Data Firehose.	firehose
Kinesis Data Streams (p. 587)	Envoie un message à un flux de données Kinesis.	kinesis
Lambda (p. 589)	.Invoke une fonction Lambda avec des données de message en entrée.	lambda
Emplacement (p. 591)	Envoie des données de localisation à Amazon Location Service.	location
OpenSearch (p. 594)	Envoie un message à un point de terminaison Amazon OpenSearch Service.	OpenSearch
Republish (p. 595)	Republie un message dans une autre rubrique MQTT.	republish
S3 (p. 597)	Stocke un message dans un compartiment Amazon Simple Storage Service (Amazon S3).	s3
Salesforce IoT (p. 599)	Envoie un message à un flux d'entrée Salesforce IoT.	salesforce
SNS (p. 600)	Publie un message sous forme de notification push Amazon Simple Notification Service (Amazon SNS).	sns
SQS (p. 601)	Envoie un message à une file d'attente Amazon Simple Queue Service (Amazon SQS).	sqrs
Step Functions (p. 603)	Démarre une machine AWS Step Functions d'état.	stepFunctions
the section called "Timestream" (p. 604)	Envoie un message à une table de base de données Amazon Timestream.	timestream

Remarques

- Définissez la règle de la même manière Région AWS que la ressource d'un autre service afin que l'action de la règle puisse interagir avec cette ressource.
- Le moteur de AWS IoT règles peut effectuer plusieurs tentatives pour effectuer une action si des erreurs intermittentes se produisent. Si toutes les tentatives échouent, le message est ignoré et l'erreur est disponible dans vos CloudWatch journaux. Vous pouvez spécifier une action en cas d'erreur pour chaque règle appelée après une défaillance. Pour plus d'informations, veuillez consulter [Gestion des erreurs \(action d'erreur\) \(p. 615\)](#).
- Certaines actions liées aux règles activent des actions dans les services qui s'intègrent à AWS Key Management Service (AWS KMS) pour prendre en charge le chiffrement des données au repos. Si vous utilisez une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer les données au repos, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour savoir comment gérer les autorisations relatives à votre clé KMS gérée par le client, consultez les rubriques relatives au chiffrement des données dans le guide de service approprié. Pour plus d'informations sur les clés KMS gérées par le client, consultez les [AWS Key Management Serviceconcepts](#) du Guide du AWS Key Management Servicedéveloppeur.

Apache Kafka

L'action Apache Kafka (Kafka) ou à des clusters Managed Streaming for Apache Kafka (Amazon MSK) ou à des clusters Apache Kafka autogérés.

Note

Cette rubrique suppose que vous êtes familiarisé avec la plateforme Apache Kafka et les concepts associés. Pour plus d'informations sur Apache Kafka, consultez [Apache Kafka](#). Vous ne pouvez pas invoquer une action Apache Kafka dans une action [Error \(p. 615\)](#).

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut assumer l'exécution des ec2:DescribeSecurityGroups opérations ec2>CreateNetworkInterface ec2:DescribeNetworkInterfacesec2>CreateNetworkInterfacePermission,ec2>DeleteNetworkInterface. Ce rôle crée et gère des interfaces réseau élastiques vers votre Amazon Virtual Private Cloud pour atteindre votre courtier Kafka. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT Core de cette action de règle.

Pour de plus amples informations sur les interfaces réseau, veuillez consulter [Interfaces réseau Elastic](#) dans le Guide de l'utilisateur Amazon EC2.

La politique associée au rôle que vous spécifiez doit ressembler à celle de l'exemple suivant :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2>CreateNetworkInterface",  
                "ec2:DescribeNetworkInterfaces",  
                "ec2>CreateNetworkInterfacePermission",  
                "ec2:DeleteNetworkInterface"  
            ]  
        }  
    ]  
}
```

```
        "ec2:DeleteNetworkInterface",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeVpcAttribute",
        "ec2:DescribeSecurityGroups"
    ],
    "Resource": "*"
}
]
}
```

- Si vous utilisez AWS Secrets Manager pour stocker les informations d'identification requises pour vous connecter à votre courtier Kafka, vous devez créer un rôle IAM AWS IoT Core capable d'effectuer les opérations `secretsmanager:GetSecretValue` et `secretsmanager:DescribeSecret`.

La politique associée au rôle que vous spécifiez doit ressembler à celle de l'exemple suivant :

```
{
    "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 pouvez exécuter vos clusters Apache Kafka dans Amazon Virtual Private Cloud (Amazon VPC). Vous devez créer une destination Amazon VPC et utiliser une passerelle NAT dans vos sous-réseaux pour transférer les messages AWS IoT vers un cluster Kafka public. Le moteur de AWS IoT règles crée une interface réseau dans chacun des sous-réseaux répertoriés dans la destination du VPC pour acheminer le trafic directement vers le VPC. Lorsque vous créez une destination VPC, le moteur de AWS IoT règles crée automatiquement une action de règle VPC. Pour plus d'informations sur les actions des règles VPC, consultez [Destination de cloud privé virtuel \(VPC\) \(p. 538\)](#).
- Si vous utilisez une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer les données au repos, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour de plus amples informations, [veuillez consulter le Manuel du](#) développeur Managed Streaming for Apache Kafka.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

Destination ARN

Amazon Resource Name (ARN) de la destination VPC. Pour plus d'informations sur la création d'une destination VPC, consultez [Destination de cloud privé virtuel \(VPC\) \(p. 538\)](#).

topic

La rubrique Kafka pour les messages à envoyer à l'agent Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, veuillez consulter [the section called “Modèles de substitution” \(p. 681\)](#).

clé (facultatif)

La clé de message Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, veuillez consulter [the section called “Modèles de substitution” \(p. 681\)](#).

partition (facultatif)

La partition de message Kafka.

Vous pouvez remplacer ce champ à l'aide d'un modèle de substitution. Pour plus d'informations, veuillez consulter [the section called “Modèles de substitution” \(p. 681\)](#).

Propriétés du client

Objet qui définit les propriétés du client producteur Apache Kafka.

Sacs (optionnels)

Le nombre d'accusés de réception que le producteur exige que le serveur ait reçus avant qu'une demande ne soit considérée comme complète.

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.

serveurs bootstrap

Liste des paires d'hôtes et de ports (host1:port1host2:port2,, etc.) utilisées 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.

security.protocol

Le protocole de sécurité utilisé pour se connecter à votre courtier Kafka.

Valeurs valides : SSL, SASL_SSL. La valeur par défaut est SSL.

key.serializer

Spécifie comment transformer en octets les objets clés que vous fournissez avec `ProducerRecord`

Valeur valide : `StringSerializer`.

value.serializer

Spécifie comment transformer en octets les objets de valeur que vous fournissez avec `ProducerRecord`

Valeur valide : `ByteBufferSerializer`.

ssl.truststore

Le fichier truststore au format base64 ou l'emplacement du fichier truststore dans [AWS Secrets Manager](#). Cette valeur n'est pas requise si les autorités de certification (CA) d'Amazon font confiance à votre boutique de confiance.

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 broker 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” \(p. 681\)](#). Pour plus d'informations sur la fonction SQL get_secret, consultez [the section called “get_secret \(secretId, SecretType, clé, roleArn\)” \(p. 650\)](#). Si le truststore se présente sous la forme d'un fichier, utilisez le SecretBinary paramètre. Si le truststore se présente sous la forme d'une chaîne, utilisez le SecretString paramètre.

La taille maximale de cette valeur est de 65 Ko.

ssl.truststore.password

Le mot de passe du truststore. Cette valeur n'est requise que si vous avez créé un mot de passe pour le truststore.

ssl.keystore

Le fichier keystore. Cette valeur est requise lorsque vous la spécifiez SSL comme valeur pour security.protocol.

Ce champ prend en charge les modèles de substitution. Utilisez 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” \(p. 681\)](#). Pour plus d'informations sur la fonction SQL get_secret, consultez [the section called “get_secret \(secretId, SecretType, clé, roleArn\)” \(p. 650\)](#). Utilisez le paramètre SecretBinary.

ssl.keystore.password

Le mot de passe de stockage pour le fichier keystore. Cette valeur est obligatoire 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. Utilisez 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” \(p. 681\)](#). Pour plus d'informations sur la fonction SQL get_secret, consultez [the section called “get_secret \(secretId, SecretType, clé, roleArn\)” \(p. 650\)](#). Utilisez le paramètre SecretString.

ssl.key.password

Le mot de passe de la clé privée enregistrée dans votre fichier de keystore.

Ce champ prend en charge les modèles de substitution. Utilisez 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” \(p. 681\)](#). Pour plus d'informations sur la fonction SQL get_secret, consultez [the section called “get_secret \(secretId, SecretType, clé, roleArn\)” \(p. 650\)](#). Utilisez le paramètre SecretString.

sasl.mechanism

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, -1 et -1, us-gov-east, us-gov-west

sasl/plain/nom d'utilisateur

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/nom d'utilisateur

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

Le 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. Utilisez 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" \(p. 681\)](#). Pour plus d'informations sur la fonction SQL get_secret, consultez [the section called "get_secret\(secretId, SecretType, clé, roleArn\)" \(p. 650\)](#). Utilisez le paramètre SecretBinary.

sasl/kerberos/service.name

Le nom principal de Kerberos sous lequel Apache Kafka s'exécute. Cette valeur est requise lorsque vous spécifiez SASL_SSL pour security.protocol et GSSAPI pour sasl.mechanism.

sasl/kerberos/krb5.kdc

Le 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

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

Identité Kerberos unique à laquelle Kerberos peut attribuer des tickets pour accéder aux services compatibles Kerberos. Cette valeur est requise lorsque vous spécifiez SASL_SSL pour security.protocol et GSSAPI pour sasl.mechanism.

Exemples

L'exemple JSON suivant définit une action Apache Kafka dans une AWS IoT règle.

```
{
```

```

"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 la configuration de Kerberos

- Votre centre de distribution de clés (KDC) doit pouvoir être résolu via un système de noms de domaine (DNS) privé au sein de 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, veuillez [consulter Utilisation 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 de destination du VPC doivent autoriser le trafic en provenance de votre VPC sur les ports suivants.
 - Trafic TCP sur le port d'écoute du broker bootstrap (souvent 9092, mais doit être compris entre 9 000 et 9 100)
 - Trafic TCP et UDP sur le port 88 pour le KDC
- SCRAM-SHA-512est le seul mécanisme de sécurité pris en charge dans les régions cn-north-1, cn-northwest-1, -1 et -1. us-gov-east us-gov-west

Destination de cloud privé virtuel (VPC)

L'action de règle Apache Kafka achemine les données vers un cluster Apache Kafka dans un Amazon Virtual Private Cloud (Amazon VPC). La configuration VPC utilisée par l'action de règle Apache Kafka est automatiquement activée lorsque vous spécifiez la destination du VPC pour votre action de règle.

Une destination VPC contient une liste de sous-réseaux au sein du VPC. Le moteur de règles crée une elastic network interface dans chaque sous-réseau que vous spécifiez dans cette liste. Pour de plus amples informations sur les interfaces réseau, veuillez consulter [Interfaces réseau Elastic](#) dans le Guide de l'utilisateur Amazon EC2.

Exigences et considérations

- Si vous utilisez un cluster Apache Kafka autogéré auquel vous pourrez accéder via un point de terminaison public via Internet :
- Créez une passerelle NAT pour les instances de vos sous-réseaux. La passerelle NAT possède une adresse IP publique qui peut se connecter à Internet, ce qui permet au moteur de règles de transférer vos messages au cluster Kafka public.
- Allouez une adresse IP Elastic avec les interfaces réseau Elastic (ENI) créées par la destination VPC. Les groupes de sécurité que vous utilisez doivent être configurés pour bloquer le trafic entrant.

Note

Si la destination VPC est désactivée puis réactivée, vous devez réassocier les adresses IP élastiques aux nouveaux ENI.

- Si la destination d'une règle thématique VPC ne reçoit aucun trafic pendant 30 jours consécutifs, elle sera désactivée.
- Si des ressources utilisées par la destination du VPC changent, la destination sera désactivée et ne pourra plus être utilisée.
- Parmi les modifications susceptibles de désactiver une destination VPC, citons : la suppression du VPC, des sous-réseaux, des groupes de sécurité ou du rôle utilisé ; la modification du rôle pour qu'il ne dispose plus des autorisations nécessaires ; et la désactivation de la destination.

Tarification

À des fins de tarification, l'action d'une règle du VPC est mesurée en plus de l'action qui envoie un message à une ressource lorsque celle-ci se trouve dans votre VPC. Pour en savoir plus sur la tarification, consultez [Tarification AWS IoT Core](#).

Création de destinations de règles thématiques de Virtual Private Cloud (VPC)

Vous créez une destination de cloud privé virtuel (VPC) à l'aide de l'[CreateTopicRuleDestination](#) API ou de la AWS IoT Core console.

Lorsque vous créez une destination VPC, vous devez spécifier les informations suivants :

vpclId

L'ID unique de la destination VPC.

subnetIds

Liste de 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.

Groupes de sécurité (facultatif)

Liste des groupes de sécurité à appliquer aux interfaces réseau.

roleArn

L'Amazon Resource Name (ARN) d'un rôle qui est autorisé à créer des interfaces réseau en votre nom.

Cet ARN doit être associée à une politique semblable à celle de l'exemple suivant.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iot:Connect",  
            "Resource": "arn:aws:iot:  
                [REDACTED]:[REDACTED]/[REDACTED]"  
        }  
    ]  
}
```

```
"Action": [
    "ec2>CreateNetworkInterface",
    "ec2>DescribeNetworkInterfaces",
    "ec2>DescribeVpcs",
    "ec2>DeleteNetworkInterface",
    "ec2>DescribeSubnets",
    "ec2>DescribeVpcAttribute",
    "ec2>DescribeSecurityGroups"
],
"Resource": "*"
},
{
    "Effect": "Allow",
    "Action": "ec2>CreateNetworkInterfacePermission",
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2:ResourceTag/VPCDestinationENI": "true"
        }
    }
},
{
    "Effect": "Allow",
    "Action": [
        "ec2>CreateTags"
    ],
    "Resource": "*",
    "Condition": {
        "StringEquals": {
            "ec2>CreateAction": "CreateNetworkInterface",
            "aws:RequestTag/VPCDestinationENI": "true"
        }
    }
}
]
```

Création d'une destination VPC à l'aide de AWS CLI

L'exemple suivant montre comment créer une destination VPC à l'aide AWS CLI de.

```
aws --region regions iot create-topic-rule-destination --destination-configuration
'vpcConfiguration={subnetIds=["subnet-123456789101230456123456789101230456123456789012:role/role-name"'}
```

Une fois que vous avez exécuté cette commande, l'état de destination du VPC sera IN_PROGRESS. Au bout de quelques minutes, son état passera soit à ERROR (si la commande échoue), soit à ENABLED. Lorsque le statut de destination est ENABLED, il est prêt à l'emploi.

Vous pouvez utiliser la commande suivante pour obtenir l'état de votre destination VPC.

```
aws --region region iot get-topic-rule-destination --arn "VPCDestinationARN"
```

Création d'une destination VPC à l'aide de la console AWS IoT Core

Les étapes suivantes décrivent comment créer une destination VPC à l'aide de la AWS IoT Core console.

1. Accédez à la console AWS IoT Core. Dans le volet de gauche, sous l'onglet Agir, choisissez Destinations.
2. Entrez des valeurs pour les champs suivants.
 - ID d'VPC
 - ID de sous-réseau
 - Security Group
3. Sélectionnez un rôle disposant des autorisations requises pour créer des interfaces réseau. L'exemple de politique précédent contient ces autorisations.

Lorsque l'état de destination du VPC est ACTIVÉ, il est prêt à être utilisé.

Alarmes CloudWatch

L'action CloudWatch alarm (`cloudWatchAlarm`) modifie l'état d'une CloudWatch alarme Amazon. Vous pouvez spécifier la raison du changement d'état et la valeur dans cet appel.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`Cloudwatch: SetAlarmState` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`alarmName`

Nom de l'alarme CloudWatch.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`stateReason`

Raisons de la modification de l'alarme.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`stateValue`

Valeur de l'état de l'alarme. Valeurs valides : OK, ALARM, INSUFFICIENT_DATA.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`roleArn`

Rôle IAM qui autorise l'accès à l'CloudWatch alarme. Pour plus d'informations, veuillez consulter [Prérequis \(p. 541\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action d'CloudWatch alarme dans une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce qu'Amazon CloudWatch ?](#) dans le guide de CloudWatch l'utilisateur d'Amazon
- [Utilisation des CloudWatch alarmes Amazon](#) dans le guide de CloudWatch l'utilisateur Amazon

Journaux CloudWatch

L'action CloudWatch Logs (`cloudwatchLogs`) envoie des données à Amazon CloudWatch Logs. Vous pouvez l'utiliser `batchMode` pour télécharger et horodater plusieurs enregistrements du journal de l'appareil dans un seul message. Vous pouvez également spécifier le groupe de journaux auquel l'action envoie les données.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut assumer la responsabilité d'exécuter les `logs:PutLogEvents` opérations `logs>CreateLogStream`, `logs:DescribeLogStreams`, et. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer les données du journal dans les CloudWatch journaux, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour de plus amples informations, veuillez consulter [Chiffrement des données de journal dans les CloudWatch journaux à l'aide d'une AWS KMS clé](#) dans le Guide de l'utilisateur Amazon CloudWatch Logs.

Exigences relatives au format de message MQTT pour batchMode

Si vous utilisez l'action de la règle CloudWatch Logs avec `batchMode` Désactivé (où se trouve la valeur du paramètre `false`), aucune exigence de formatage des messages MQTT n'est requise. Remarque :

la valeur par défaut du batchMode paramètre est `false`. Toutefois, si vous utilisez l'action relative à la règle CloudWatch Logs avec batchMode turned on (`true`), les messages MQTT contenant des journaux côté appareil doivent être formatés de manière à contenir un horodatage et une charge utile du message. Remarque : timestamp représente l'heure à laquelle l'événement s'est produit et est exprimée en nombre de millisecondes après le 1er janvier 1970 00:00:00 UTC.

Voici un exemple du format de publication :

```
[  
  {"timestamp": 1673520691093, "message": "Test message 1"},  
  {"timestamp": 1673520692879, "message": "Test message 2"},  
  {"timestamp": 1673520693442, "message": "Test message 3"}]
```

Selon la manière dont les journaux côté appareil sont générés, il peut être nécessaire de les filtrer et de les reformater avant d'être envoyés afin de se conformer à cette exigence. Pour plus d'informations, consultez Charge [utile du message MQTT](#).

Quel que soit le batchMode paramètre, message le contenu doit respecter les limites de taille des AWS IoT messages. Pour de plus amples informations, consultez [Points de terminaison et quotas AWS IoT Core](#).

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`logGroupName`

Le groupe de CloudWatch journaux dans lequel l'action envoie des données.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`roleArn`

Rôle IAM qui fournit un accès au groupe de CloudWatch journaux. Pour plus d'informations, veuillez consulter [Prérequis \(p. 542\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non
(facultatif) `batchMode`

Indique si des lots d'enregistrements de journal seront extraits et téléchargés dans CloudWatch. Les valeurs incluent `true` ou `false` (par défaut). Pour plus d'informations, veuillez consulter [Prérequis \(p. 542\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action CloudWatch Logs dans une AWS IoT règle.

```
{  
  "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",
        "batchMode": false
    }
}
}
```

Consulter aussi

- [Qu'est-ce qu'Amazon CloudWatch Logs ?](#) dans le guide de l'utilisateur d'Amazon CloudWatch Logs

Métriques CloudWatch

L'action CloudWatch metric (`cloudwatchMetric`) capture une CloudWatch métrique Amazon. Vous pouvez spécifier le namespace, le nom, la valeur, l'unité et l'horodatage de la métrique.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`cloudwatch:PutMetricData` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`metricName`

Nom de la métrique CloudWatch.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`metricNamespace`

Le nom de l'espace de nommage de la CloudWatch métrique.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`metricUnit`

Unité de métrique prise en charge par CloudWatch.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`metricValue`

Chaîne contenant la valeur de la CloudWatch métrique.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`metricTimestamp`

(Facultatif) Chaîne contenant l'horodatage, exprimé en secondes à l'époque Unix. La valeur par défaut est l'heure de l'époque Unix actuelle.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleArn

Rôle IAM qui autorise l'accès à la CloudWatch métrique. Pour plus d'informations, veuillez consulter [Prérequis \(p. 544\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action CloudWatch métrique dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "cloudwatchMetric": {  
                    "metricName": "IoMetric",  
                    "metricNamespace": "IoNamespace",  
                    "metricUnit": "Count",  
                    "metricValue": "1",  
                    "metricTimestamp": "1456821314",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw"  
                }  
            }  
        ]  
    }  
}
```

L'exemple JSON suivant définit une action CloudWatch métrique avec des modèles de substitution dans une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce qu'Amazon CloudWatch ?](#) dans le guide de CloudWatch l'utilisateur d'Amazon
- [Utilisation des CloudWatch métriques Amazon](#) dans le guide de CloudWatch l'utilisateur d'Amazon

DynamoDB

L'action DynamoDB (dynamoDB) écrit tout ou partie d'un message MQTT dans une table Amazon DynamoDB.

Vous pouvez suivre un didacticiel qui explique comment créer et tester une règle avec une action DynamoDB. Pour plus d'informations, veuillez consulter [Tutoriel : Stockage des données de l'appareil dans une table DynamoDB \(p. 229\)](#).

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.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`dynamodb:PutItem` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer des données au repos dans DynamoDB, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour plus d'informations, consultez la section [Clé KMS gérée par le client](#) dans le guide de démarrage Amazon DynamoDB.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

tableName

Le nom de la table DynamoDB.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

hashKeyField

Nom de la clé de hachage (également appelée clé de partition).

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

hashKeyType

(Facultatif) Type de données de la clé de hachage (également appelée clé de partition). Valeurs valides : STRING, NUMBER.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

hashKeyValue

Valeur de la clé de hachage. Envisagez d'utiliser un modèle de substitution tel que \${topic()} ou \${timestamp()}.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

rangeKeyField

(Facultatif) Nom de la clé de plage (également appelée clé de tri).

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

rangeKeyType

(Facultatif) Type de données de la clé de plage (également appelée clé de tri). Valeurs valides : STRING, NUMBER.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

rangeKeyValue

(Facultatif) Valeur de la clé de plage. Envisagez d'utiliser un modèle de substitution tel que \${topic()} ou \${timestamp()}.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

payloadField

(Facultatif) Nom de la colonne dans laquelle la charge utile est écrite. Si vous omettez cette valeur, la charge utile est écrite dans la colonne nommée payload

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

operation

(Facultatif) Le type d'opération à effectuer. Valeurs valides : INSERT, UPDATE, DELETE.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleARN

Rôle IAM qui autorise l'accès à la table DynamoDB. Pour plus d'informations, veuillez consulter [Prérequis \(p. 546\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Les données écrites dans la table DynamoDB sont le résultat de l'instruction SQL de la règle.

Exemples

L'exemple JSON suivant définit une action DynamoDB dans une AWS IoT règle.

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

```
}
```

Consulter aussi

- [Qu'est-ce qu'Amazon DynamoDB ?](#) dans le Manuel du développeur Amazon DynamoDB
- [Démarrez avec DynamoDB](#) dans le guide du développeur Amazon DynamoDB
- [Tutoriel : Stockage des données de l'appareil dans une table DynamoDB \(p. 229\)](#)

DynamoDBv2

L'action DynamoDBv2 (dynamoDBv2) écrit tout ou partie d'un message MQTT dans une table Amazon DynamoDB. Chaque attribut de la charge utile est écrit dans une colonne distincte de la base de données DynamoDB.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'dynamodb : PutItem opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de 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 une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer des données au repos dans DynamoDB, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour plus d'informations, consultez la section [Clé KMS gérée par le client](#) dans le guide de démarrage Amazon DynamoDB.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`putItem`

Objet qui spécifie 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.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`roleARN`

Rôle IAM qui autorise l'accès à la table DynamoDB. Pour plus d'informations, veuillez consulter [Prérequis \(p. 548\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Les données écrites dans la table DynamoDB sont le résultat de l'instruction SQL de la règle.

Exemples

L'exemple JSON suivant définit une action DynamoDBv2 dans une règle AWS IoT

```
{  
    "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 une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce qu'Amazon DynamoDB ?](#) dans le Manuel du développeur Amazon DynamoDB
- [Démarrez avec DynamoDB](#) dans le guide du développeur Amazon DynamoDB

Elasticsearch

L'action Elasticsearch (elasticsearch) écrit les données des messages MQTT dans un domaine Amazon OpenSearch Service. Vous pouvez ensuite utiliser des outils tels que OpenSearch les tableaux de bord pour interroger et visualiser des données dans OpenSearch Service.

Warning

L'action Elasticsearch ne peut être utilisée que par les actions de règle existantes. Pour créer une nouvelle action de règle ou pour mettre à jour une action de règle existante, utilisez l'action de règle OpenSearch à la place. Pour plus d'informations, veuillez consulter [OpenSearch \(p. 594\)](#).

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'opération `ESHttpPut`. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez une clé gérée par le client AWS KMS key (clé KMS) pour chiffrer les données au repos `OpenSearch`, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour plus d'informations, consultez la section [Chiffrement des données au repos pour Amazon OpenSearch Service](#) dans le manuel [Amazon OpenSearch Service Developer Guide](#).

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`endpoint`

Le point de terminaison de votre domaine de service.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement
`index`

Index dans lequel vous souhaitez stocker vos données.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui
`type`

Type de document que vous stockez.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui
`id`

Identifiant unique de chaque document.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui
`roleARN`

Rôle IAM qui fournit un accès au domaine OpenSearch Service. Pour plus d'informations, veuillez consulter [Prérequis \(p. 550\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action Elasticsearch dans une AWS IoT règle et explique comment spécifier les champs de cette action. `elasticsearch` Pour plus d'informations, veuillez consulter [ElasticsearchAction](#).

```
{  
    "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 une AWS IoT règle.

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

Consulter aussi

- [OpenSearch \(p. 594\)](#)
- [Qu'est-ce qu'Amazon OpenSearch Service ?](#)

HTTP

L'action HTTPS (http) envoie les données d'un message MQTT à une application ou à un service Web.

Prérequis

Cette action de règle présente les exigences suivantes :

- Vous devez confirmer et activer les points de terminaison HTTPS avant que le moteur de règles puisse les utiliser. Pour plus d'informations, veuillez consulter [Utilisation des destinations de règle de rubrique \(p. 554\)](#).

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`url`

Point de terminaison HTTPS auquel 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.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`confirmationUrl`

(Facultatif) Si spécifié, AWS IoT utilise l'URL de confirmation pour créer une destination de règle de sujet correspondante. Vous devez activer la destination de la règle de rubrique avant de l'utiliser dans une action HTTP. Pour plus d'informations, veuillez consulter [Utilisation des destinations de règle de rubrique \(p. 554\)](#). 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`.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`headers`

(Facultatif) La liste des en-têtes à inclure dans les requêtes HTTP adressées au point de terminaison. Chaque en-tête doit contenir les informations suivantes :

`key`

La clé de l'en-tête.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

`value`

Valeur de l'en-tête.

Supporte les [modèles de substitution \(p. 681\)](#) : 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 l'`url` argument. Actuellement, Signature Version 4 est le seul type d'authentification pris en charge. Pour de plus amples informations, veuillez consulter [Autorisation HTTP](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une AWS IoT règle avec une action HTTP.

```
{  
    "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

Le moteur de AWS IoT règles réessaie l'action HTTP conformément aux 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 :
 - L'essai précédent a fourni une réponse supérieure à 16 384 octets.
 - Le service web ou l'application en aval ferme la connexion TCP après la tentative.
 - Le temps total nécessaire pour traiter une demande avec de 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.

Consulter aussi

- [Utilisation des destinations de règle de rubrique \(p. 554\)](#)
- [Acheminez les données AWS IoT Core directement depuis vos services Web](#) dans l'Internet des objets sur le AWS blog

Utilisation des destinations de règle de rubrique

Une destination de règle de rubrique HTTP est un service Web vers lequel le moteur de règles peut acheminer des données à partir d'une règle de rubrique. Une AWS IoT Core ressource décrit le service Web pour AWS IoT. Les ressources de destination des règles de rubrique peuvent être partagées par différentes règles.

Avant de AWS IoT Core pouvoir envoyer des données à un autre service Web, celui-ci doit confirmer qu'il peut accéder au point de terminaison du service.

Vue d'ensemble des règles de rubrique HTTP

Une destination de règle thématique HTTP fait référence à un service Web qui prend en charge une URL de confirmation et une ou plusieurs URL de collecte de données. La ressource de destination des règles thématiques HTTP contient l'URL de confirmation de votre service Web. Lorsque vous configurez une action de règle de rubrique HTTP, vous spécifiez l'URL réelle du point de terminaison qui doit recevoir les données ainsi que l'URL de confirmation du service Web. Une fois votre destination confirmée, la règle de rubrique envoie le résultat de l'instruction SQL au point de terminaison HTTPS (et non à l'URL de confirmation).

Une destination de règle de rubrique HTTP peut avoir l'un des états suivants :

ENABLED

La destination a été confirmée et peut être utilisée par une action de règle. L'état d'une destination doit être ENABLED (ACTIVÉ) pour qu'elle soit utilisée dans une règle. Vous ne pouvez activer qu'une destination dont le statut est DÉSACTIVÉ.

DISABLED

La destination a été confirmée mais elle ne peut pas être utilisée par une action de règle. 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 ne pouvez désactiver qu'une destination dont le statut est ACTIVÉ.

IN_PROGRESS

La confirmation de la destination est en cours.

ERROR

La confirmation de la destination a expiré.

Une fois qu'une destination de règle de rubrique HTTP a été confirmée et activée, elle peut être utilisée avec n'importe quelle règle de votre compte.

Les sections suivantes décrivent les actions courants sur les destinations des règles de rubrique HTTP.

Création et confirmation des destinations des règles de rubrique HTTP

Vous créez une destination de règle de rubrique HTTP en appelant l'`CreateTopicRuleDestination` opération ou en utilisant la AWS IoT console.

Lorsque vous créez une destination, AWS IoT envoie une demande de confirmation à l'URL de confirmation. 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"  
}
```

Le contenu de la demande de confirmation inclut les informations suivants :

arn

Amazon Resource Name (ARN) de destination de la règle de rubrique à confirmer.
confirmationToken

Le jeton de confirmation envoyé par AWS IoT Core. Dans l'exemple, le jeton est tronqué. Votre jeton sera plus long. Vous aurez besoin de ce jeton pour confirmer votre destination AWS IoT Core.

enableUrl

L'URL à laquelle vous accédez pour confirmer la destination d'une règle de rubrique.
messageType

Type du message.

Pour terminer le processus de confirmation du point de terminaison, vous devez effectuer l'une des opérations suivantes une fois que votre URL de confirmation a reçu la demande de confirmation.

- Appelez le enableUrl dans la demande de confirmation, puisappelez UpdateTopicRuleDestination pour définir le statut de la règle de sujet sur ENABLED.
- Appelez l'ConfirmTopicRuleDestinationopération et transmettez le résultat confirmationToken de la demande de confirmation.
- Copiez le confirmationToken et collez-le dans la boîte de dialogue de confirmation de la destination dans la AWS IoT console.

Envoi d'une nouvelle demande de confirmation

Pour activer un nouveau message de confirmation pour une destination,appelez UpdateTopicRuleDestination et définissez le statut de la destination de la règle thématique sur IN_PROGRESS.

Répétez le processus de confirmation après avoir envoyé une nouvelle demande de confirmation.

Désactivation et suppression 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. Une règle de rubrique dont l'état est DÉSACTIVÉ peut être réactivée sans qu'il soit nécessaire d'envoyer une nouvelle demande de confirmation.

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 des règles thématiques

Les autorités de certification suivantes sont prises en charge par les points de terminaison HTTPS dans les destinations des règles thématiques. Vous pouvez choisir l'une de ces autorités de certification prises

en charge. Les signatures sont fournies à titre de référence. Notez que vous ne pouvez pas utiliser de certificats autosignés car ils ne fonctionneront pas.

Aidez-nous à améliorer ce sujet

[Dites-nous ce que vous en pensez](#)

```
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: utndatacorpsgcc  
Certificate fingerprints:  
MD5: B3:A5:3E:77:21:60: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:E: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: staatderlandenevrootca
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: swissignssilverg2ca
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: swissignssilvercag2
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: atotrustedroot2011
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: soneraaclass1ca
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:FA:4C:15:64:E1:84

Alias name: certplusclass3ppprimaryca
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: swissigngoldg2ca
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: securitycommunicationenvrootca1
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: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: globalsecureeccrootcar5  
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: tubitakkamussslkoksertifikasisurum1
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: trustcorecal
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: secomsrootca2
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: eecertificationcentreroottca
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: equifaxsecurebusinessca1
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: soneraaclass2ca
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: deutschetelkomrootca2
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: visaecommceroot
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:D5:5C: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: camerfirmachambersofcommerceroott  
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
```

IoT Analytics

L'action AWS IoT Analytics (`iotAnalytics`) envoie les données d'un message MQTT vers un AWS IoT Analytics canal.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`iotanalytics:BatchPutMessage` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

La politique associée au rôle que vous spécifiez doit ressembler à celle de l'exemple suivant :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iotanalytics:BatchPutMessage",  
            "Resource": [  
                "arn:aws:iotanalytics:us-west-2:account-id:channel/mychannel"  
            ]  
        }  
    ]  
}
```

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

batchMode

(Facultatif) Pour traiter ou non l'action en tant que lot. La valeur par défaut est `false`.

Quand `batchMode` est `true` et que l'instruction SQL de la règle est évaluée dans un tableau, chaque élément du tableau est fourni sous la forme d'un message distinct lorsqu'il est transmis par [BatchPutMessage](#) au AWS IoT Analytics canal. Le tableau résultant ne peut pas contenir plus de 100 messages.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

channelName

Nom du canal AWS IoT Analytics dans lequel les données doivent être écrites.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

roleArn

Rôle IAM qui autorise l'accès au AWS IoT Analytics canal. Pour plus d'informations, veuillez consulter [Prérequis \(p. 578\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

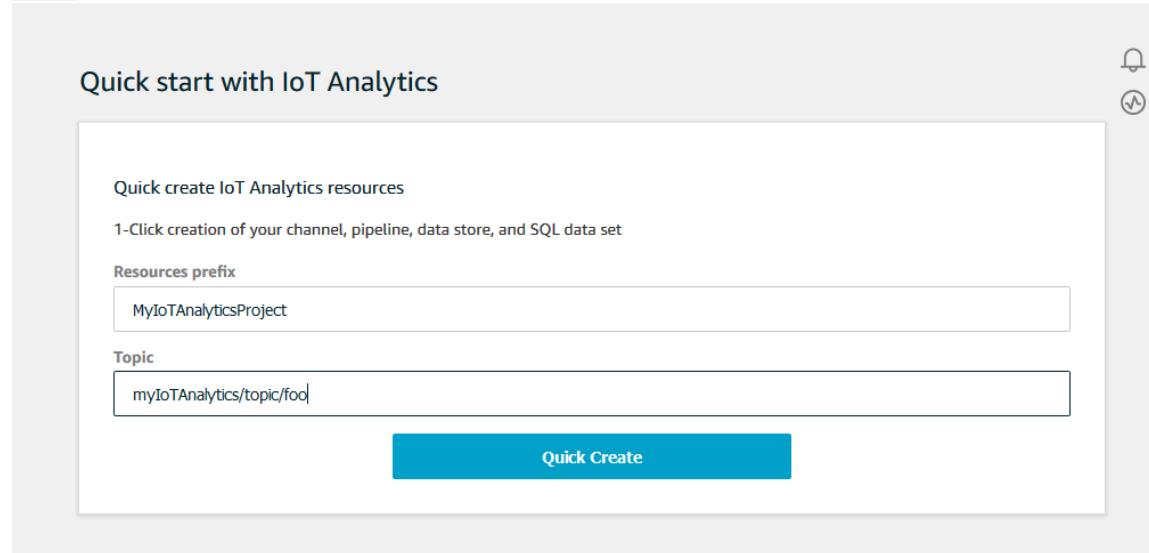
L'exemple JSON suivant définit une AWS IoT Analytics action dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "iotAnalytics": {  
                    "channelName": "mychannel",  
                    "roleArn": "arn:aws:iam::123456789012:role/analyticsRole",  
                }  
            }  
        ]  
    }  
}
```

}

Consulter aussi

- [Qu'est-ce que AWS IoT Analytics ?](#) dans le Guide de l'utilisateur AWS IoT Analytics
- La AWS IoT Analytics console dispose également d'une fonction de démarrage rapide qui vous permet de créer un canal, un magasin de données, un pipeline et un magasin de données en un seul clic. Pour plus d'informations, consultez le guide de [démarrage rapide de la AWS IoT Analytics console dans le Guide de l'AWS IoT Analyticsutilisateur.](#)



AWS IoT Events

L'action AWS IoT Events (`iotEvents`) envoie les données d'un message MQTT à une AWS IoT Events entrée.

Important

Si la charge utile est envoyée AWS IoT Core sans le `input` attribut `Key`, ou si la clé ne se trouve pas dans le même chemin JSON spécifié dans la clé, la règle IoT échouera avec l'erreur `Failed to send message to IoT Events`.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`iotevents:BatchPutMessage` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

batchMode

(Facultatif) Pour traiter ou non l'action d'événement en tant que lot. La valeur par défaut est `false`.

Quand `batchMode` est `true` et que l'instruction SQL de la règle est évaluée dans un tableau, chaque élément du tableau est traité comme un message distinct lorsqu'il est envoyé à AWS IoT Events 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`.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

inputName

Nom de l'entrée AWS IoT Events.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

messageId

(Facultatif) Utilisez cette option pour vérifier qu'une seule entrée (message) avec un donné `messageId` est traitée par un AWS IoT Events détecteur. Vous pouvez utiliser le modèle de `${newuuid() }` substitution pour générer un identifiant unique pour chaque demande.

Quand `batchMode` équivaut à `true`, vous ne pouvez pas spécifier de valeur `messageId` ; une nouvelle valeur UUID sera attribuée.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleArn

Rôle IAM qui permet d'AWS IoT envoyer une entrée à un AWS IoT Events détecteur. Pour plus d'informations, veuillez consulter [Prérequis \(p. 580\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action IoT Events dans une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce que AWS IoT Events ?](#) dans le Guide du développeur AWS IoT Events

AWS IoT SiteWise

L'action AWS IoT SiteWise (`iotSiteWise`) envoie les données d'un message MQTT aux propriétés des actifs dans AWS IoT SiteWise.

Vous pouvez suivre un didacticiel qui vous montre comment ingérer des données à partir d'objets AWS IoT. Pour plus d'informations, consultez le didacticiel [Ingestion de données AWS IoT SiteWise depuis des AWS IoT objets](#) ou la section [Ingestion de données à l'aide des règles de AWS IoT base](#) du Guide de l'AWS IoT SiteWise utilisateur.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`iotsitewise:BatchPutAssetPropertyValue` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Vous pouvez associer l'exemple de politique de confiance 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 à l'AWS IoT SiteWise à l'aide de cette action, celles-ci doivent répondre aux exigences de l'`BatchPutAssetPropertyValue` opération. Pour plus d'informations, consultez [BatchPutAssetPropertyValue](#) dans la Référence d'API AWS IoT SiteWise.

Paramètres

Lorsque vous créez une AWS IoT règle avec 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) Alias de propriété associé à votre propriété de ressources. Spécifiez un `propertyAlias` ou les deux `propertyId.assetId`. Pour de plus amples informations sur les alias de propriétés, veuillez consulter [Mappage de flux de données industrielles avec des propriétés de ressource](#) dans le Guide de AWS IoT SiteWise l'utilisateur.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

assetId

(Facultatif) ID de la AWS IoT SiteWise ressource. Spécifiez un `propertyAlias` ou les deux `propertyId.assetId`

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

propertyId

(Facultatif) ID de la propriété de la ressource. Spécifiez un `propertyAlias` ou les deux `propertyId.assetId`

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

entryId

(Facultatif) Identifiant unique pour cette entrée. Définissez le `entryId` pour mieux suivre le message à l'origine d'une erreur en cas d'échec. La valeur par défaut est un nouvel UUID.

Supporte les [modèles de substitution \(p. 681\)](#) : 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\(\) \(p. 673\)](#), qui renvoie l'heure actuelle en millisecondes. Pour convertir cette heure en secondes, vous pouvez utiliser le modèle de substitution suivant : `${\bf \{ floor(timestamp() / 1E3) \}}`.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

offsetInNanos

(Facultatif) Chaîne contenant le décalage temporel en nanosecondes par rapport au temps en secondes. Si votre charge utile de message n'a pas d'horodatage, vous pouvez utiliser [timestamp\(\) \(p. 673\)](#), 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 : `${\bf \{ (timestamp() \% 1E3) * 1E6 \}}`.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

En ce qui concerne l'heure d'Unix, n'AWS IoT SiteWise accepte que les entrées dont l'horodatage est compris entre 7 jours dans le passé et 5 minutes dans le futur.

quality

(Facultatif) Une chaîne qui décrit la qualité de la valeur. Valeurs valides : GOOD, BAD, UNCERTAIN.

Supporte les [modèles de substitution \(p. 681\)](#) : 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) Une chaîne qui contient la valeur booléenne de l'entrée de valeur.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

doubleValue

(Facultatif) Une chaîne qui contient la valeur double de la valeur saisie.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

integerValue

(Facultatif) Une chaîne qui contient la valeur entière de l'entrée de valeur.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

stringValue

(Facultatif) La valeur de chaîne de l'entrée de valeur.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleArn

ARN du rôle IAM qui accorde l'AWS IoT autorisation d'envoyer la valeur de propriété de ressource à AWS IoT SiteWise. Pour plus d'informations, veuillez consulter [Prérequis \(p. 582\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une SiteWise action IoT de base dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "iotSiteWise": {  
                    "putAssetPropertyValueEntries": [  
                        {  
                            "propertyAlias": "/some/property/alias",  
                            "propertyValues": [  
                                {  
                                    "value": "12345"  
                                }  
                            ]  
                        }  
                    ]  
                }  
            }  
        ]  
    }  
}
```

```

        "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 SiteWise action IoT dans une AWS IoT règle. Cet exemple utilise la rubrique comme alias de propriété et la fonction `timestamp()`. Par exemple, si vous publiez des données sur/`company/windfarm/3/turbine/7/rpm`, cette action envoie les données à la propriété de la 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"
                    }
                }
            }
        ]
    }
}
```

Consulter aussi

- [Qu'est-ce que AWS IoT SiteWise ?](#) dans le Guide de l'utilisateur AWS IoT SiteWise
- [Ingestion des données à l'aide AWS IoT Core des règles](#) du guide de l'AWS IoT SiteWiseutilisateur
- [Ingestion de données AWS IoT SiteWise depuis des AWS IoT objets du Guide de l'AWS IoT SiteWiseutilisateur](#)
- [Résolution des problèmes liés à une action liée à une AWS IoT SiteWise règle](#) dans le Guide de AWS IoT SiteWise l'utilisateur

Kinesis Data Firehose

L'action Kinesis Data Firehose (`firehose`) envoie les données d'un message MQTT vers un flux Amazon Kinesis Data Firehose.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`firehose:PutRecord` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez Kinesis Data Firehose pour envoyer des données vers un compartiment Amazon S3 et que vous utilisez un compartiment AWS KMS géré par le client AWS KMS key pour chiffrer les données au repos dans Amazon S3, Kinesis Data Firehose doit avoir accès à votre compartiment et être autorisé à l'utiliser pour le compte de l'appelant. AWS KMS key Pour de plus amples informations, [veuillez consulter Accord à Kinesis Data Firehose dans le Manuel du développeur Amazon Kinesis Data Firehose](#).

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`batchMode`

(Facultatif) Indique s'il convient de diffuser le flux Kinesis Data Firehose sous forme de lot en tant que lot en tant que. [PutRecordBatch](#) La valeur par défaut est `false`.

Quand `batchMode` est `true` et que l'instruction SQL de la règle est évaluée dans un tableau, chaque élément du tableau forme un enregistrement dans la demande `PutRecordBatch`. Le tableau résultant ne peut pas contenir plus de 500 enregistrements.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

`deliveryStreamName`

Le flux Kinesis Data Firehose dans lequel les données du message doivent être écrites.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`separator`

(Facultatif) Séparateur de caractères utilisé pour séparer les enregistrements écrits dans le flux Kinesis Data Firehose. Si vous ne spécifiez pas ce paramètre, le flux n'utilise aucun séparateur. Valeurs valides : , (virgule), \t (tab), \n (nouvelle ligne), \r\n (nouvelle ligne Windows).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

`roleArn`

Rôle IAM qui fournit un accès au flux Kinesis Data Firehose. Pour plus d'informations, veuillez consulter [Prérequis \(p. 586\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action Kinesis Data Firehose dans une AWS IoT règle.

```
{  
    "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 une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce qu'Amazon Kinesis Data Firehose ?](#) dans le Guide du développeur Amazon Kinesis Data Firehose

Kinesis Data Streams

L'action Kinesis Data Streams (kinesis) écrit les données d'un message MQTT dans Amazon Kinesis Data Streams.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`kinesis:PutRecord` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez une clé AWS KMS gérée par le client AWS KMS key (clé KMS) pour chiffrer des données au repos dans Kinesis Data Streams, le service doit être autorisé à l'utiliser pour le compte de l'AWS KMS key appelant. Pour de plus amples informations, veuillez consulter [Autorisations d'utilisation générées par l'utilisateur AWS KMS keys](#) dans le Manuel du développeur Amazon Kinesis Data Streams.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

stream

Le flux de données Kinesis dans lequel les données doivent être écrites.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement
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()}).

Supporte les [modèles de substitution \(p. 681\)](#) : Oui
roleArn

ARN du rôle IAM qui accorde l'AWS IoT autorisation d'accéder au flux de données Kinesis. Pour plus d'informations, veuillez consulter [Prérequis \(p. 587\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action Kinesis Data Streams dans une AWS IoT règle.

```
{  
    "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 une AWS IoT règle.

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

```
        ]  
    }  
}
```

Consulter aussi

- [Qu'est-ce qu'Amazon Kinesis Data Streams ?](#) dans le Guide du développeur Amazon Kinesis Data Streams

Lambda

Une action Lambda (lambda) invoque une AWS Lambda fonction et transmet un message MQTT. AWS IoT appelle des fonctions Lambda de façon asynchrone.

Vous pouvez suivre un didacticiel qui explique comment créer et tester une règle avec une action Lambda. Pour plus d'informations, veuillez consulter [Tutorial : Formatage d'une notification à l'aide d'une AWS Lambda fonction \(p. 235\)](#).

Prérequis

Cette action de règle présente les exigences suivantes :

- AWS IoT Pour appeler une fonction Lambda, vous devez configurer une politique qui accorde l'lambda:InvokeFunction autorisation à AWS IoT. Vous pouvez appeler une fonction Lambda définie dans la même fonction Région AWS lorsque votre politique Lambda existe. Les fonctions Lambda utilisent des politiques basées sur des ressources, vous devez donc associer la politique à la fonction Lambda.

Utilisez la AWS CLI commande suivante pour joindre une politique qui accorde l'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 commande a les paramètres suivants :

--function-name

Nom de la fonction Lambda. Vous ajoutez une nouvelle autorisation pour mettre à jour la politique de ressources de la fonction.

--region

Le Région AWS de la fonction.

--principal

Le principal qui obtient l'autorisation. Cela devrait iot.amazonaws.com permettre d'appeler AWS IoT la fonction Lambda.

--source-arn

ARN de la règle. Vous pouvez utiliser la get-topic-rule AWS CLI commande pour obtenir l'ARN d'une règle.

--source-account

L'Compte AWS dont où la règle est définie.

--statement-id

Identifiant unique de l'instruction.

--action

Action Lambda que vous souhaitez autoriser dans cette instruction. Pour autoriser AWS IoT l'appel d'une fonction Lambda, spécifiez. lambda:InvokeFunction

Important

Si vous ajoutez une autorisation pour un AWS IoT principal sans fournir le source-arn ousource-account, toute personne Compte AWS qui crée une règle avec votre action Lambda peut activer des règles à partir desquelles votre fonction Lambda est invoquée. AWS IoT

Pour plus d'informations, consultez [Autorisations AWS Lambda](#).

- Si vous utilisez une solution AWS KMS gérée par le client AWS KMS key pour chiffrer des données au repos dans Lambda, le service doit être autorisé à l'utiliser pour le compte de AWS KMS key l'appelant. Pour plus d'informations, veuillez consulter la rubrique [Chiffrement au repos](#) dans le Guide du développeur AWS Lambda.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

functionArn

ARN de la fonction Lambda à appeler. AWS IoT doit avoir l'autorisation d'invoquer la fonction. Pour plus d'informations, veuillez consulter [Prérequis \(p. 589\)](#).

Si vous ne spécifiez pas de version ou d'alias pour votre fonction Lambda, la version la plus récente de la fonction est fermée. Vous pouvez spécifier une version ou un alias si vous souhaitez arrêter une version spécifique de votre fonction Lambda. Pour spécifier une version 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 le versionnement et les alias, consultez la section [Versionnage des AWS Lambda fonctions et alias](#).

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

Exemples

L'exemple JSON suivant définit une action Lambda dans une AWS IoT règle.

```
{  
    "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 une AWS IoT règle.

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "lambda": {
                    "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:${topic()}"
                }
            }
        ]
    }
}
```

Consulter aussi

- [Qu'est-ce que AWS Lambda ?](#)dans le Guide du développeur AWS Lambda
- [Tutoriel : Formatage d'une notification à l'aide d'uneAWS Lambda fonction \(p. 235\)](#)

Emplacement

L'action Location (location) envoie vos données de localisation géographique à [Amazon Location Service](#).

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'geo:BatchUpdateDevicePositionopération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

deviceId

ID unique de l'appareil fournissant les données de localisation. Pour plus d'informations, consultez la référence [DeviceId](#)de l'API Amazon Location Service.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

latitude

Chaîne qui correspond à une valeur double représentant la latitude de l'emplacement de l'appareil.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

longitude

Chaîne qui correspond à une valeur double représentant la longitude de l'emplacement de l'appareil.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleArn

Rôle IAM qui fournit un accès au domaine Amazon Location Service. Pour plus d'informations, veuillez consulter [Prérequis \(p. 591\)](#).

timestamp

Heure à laquelle les données de localisation ont été échantillonnées. La valeur par défaut correspond à l'heure à laquelle le message MQTT a été traité.

La timestamp valeur se compose des deux valeurs suivants :

- **value**: expression qui renvoie une valeur temporelle de longue époque. Vous pouvez utiliser cette [the section called “time_to_epoch \(Chaîne, Chaîne\)” \(p. 672\)](#) fonction pour créer un horodatage valide à partir d'une valeur de date ou d'heure transmise dans la charge utile du message. Supporte les [modèles de substitution \(p. 681\)](#) : Oui.
- **unit**: (Facultatif) Précision de la valeur d'horodatage résultant de l'expression décrite dans. **value** Valeurs valides : SECONDS | MILLISECONDS | MICROSECONDS |NANOSECONDS. La valeur par défaut est MILLISECONDS. Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement.

trackerName

Nom de la ressource de suivi Amazon Location dans laquelle l'emplacement est mis à jour. Pour plus d'informations, consultez [Tracker](#) dans le guide du développeur Amazon Location Service.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

Exemples

L'exemple JSON suivant définit une action de localisation dans une AWS IoT règle.

```
{  
  "topicRulePayload": {  
    "sql": "SELECT * FROM 'some/topic'",  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23",  
    "actions": [  
      {  
        "location": {  
          "roleArn": "arn:aws:iام::123454962127:role/service-role/ExampleRole",  
          "trackerName": "MyTracker",  
          "deviceId": "001",  
          "sampleTime": {  
            "value": "${timestamp()}",  
            "unit": "SECONDS"  
          },  
          "latitude": "-12.3456",  
          "longitude": "65.4321"  
        }  
      }  
    ]  
  }  
}
```

```
}
```

L'exemple JSON suivant définit une action Location avec des modèles de substitution dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "location": {  
                    "roleArn": "arn:aws:iam::123456789012:role/service-role/ExampleRole",  
                    "trackerName": "${TrackerName}",  
                    "deviceId": "${DeviceID}",  
                    "timestamp": {  
                        "value": "${timestamp()}",  
                        "unit": "SECONDS"  
                    },  
                    "latitude": "${get(position, 0)}",  
                    "longitude": "${get(position, 1)}"  
                }  
            ]  
        }  
    }  
}
```

L'exemple de charge utile MQTT suivant montre comment les modèles de substitution de l'exemple précédent accèdent aux données. Vous pouvez utiliser la commande [get-device-position-history](#) CLI pour vérifier que les données de charge utile MQTT sont fournies dans votre système de localisation.

```
{  
    "TrackerName": "mytracker",  
    "DeviceID": "001",  
    "position": [  
        "-12.3456",  
        "65.4321"  
    ]  
}
```

```
aws location get-device-position-history --device-id 001 --tracker-name mytracker
```

```
{  
    "DevicePositions": [  
        {  
            "DeviceId": "001",  
            "Position": [  
                -12.3456,  
                65.4321  
            ],  
            "ReceivedTime": "2022-11-11T01:31:54.464000+00:00",  
            "SampleTime": "2022-11-11T01:31:54.308000+00:00"  
        }  
    ]  
}
```

Consulter aussi

- [Qu'est-ce qu'Amazon Location Service ?](#) dans le Guide du développeur d'Amazon Location Service.

OpenSearch

L'action OpenSearch (openSearch) écrit les données des messages MQTT dans un domaine Amazon OpenSearch Service. Vous pouvez ensuite utiliser des outils tels que OpenSearch les tableaux de bord pour interroger et visualiser des données dans OpenSearch Service.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'opération : `ESHttpPut`. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez une solution gérée par le client AWS KMS key pour chiffrer les données au repos dans le OpenSearch Service, le service doit être autorisé à utiliser la clé KMS pour le compte de l'appelant. Pour plus d'informations, consultez la section [Chiffrement des données au repos pour Amazon OpenSearch Service](#) dans le manuel Amazon OpenSearch Service Developer Guide.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

endpoint

Le point de terminaison de votre domaine Amazon OpenSearch Service.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

index

Index OpenSearch dans lequel vous souhaitez stocker vos données.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

type

Type de document que vous stockez.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

id

Identifiant unique de chaque document.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleARN

Rôle IAM qui fournit un accès au domaine OpenSearch Service. Pour plus d'informations, veuillez consulter [Prérequis \(p. 594\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Limites

L'action OpenSearch (openSearch) ne peut pas être utilisée pour fournir des données aux clusters VPC Elasticsearch.

Exemples

L'exemple JSON suivant définit une OpenSearch action dans une AWS IoT règle et explique comment spécifier les champs de cette OpenSearch action. Pour plus d'informations, veuillez consulter [OpenSearchAction](#).

```
{  
    "topicRulePayload": {  
        "sql": "SELECT *, timestamp() as timestamp FROM 'iot/test'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "openSearch": {  
                    "endpoint": "https://my-endpoint",  
                    "index": "my-index",  
                    "type": "my-type",  
                    "id": "${newuuid()}",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_os"  
                }  
            }  
        ]  
    }  
}
```

L'exemple JSON suivant définit une OpenSearch action avec des modèles de substitution dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "openSearch": {  
                    "endpoint": "https://my-endpoint",  
                    "index": "${topic()}",  
                    "type": "${type}",  
                    "id": "${newuuid()}",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_os"  
                }  
            }  
        ]  
    }  
}
```

Consulter aussi

[Qu'est-ce qu'Amazon OpenSearch Service ?](#) dans le manuel Amazon OpenSearch Service Developer Guide

Republish

L'action republier (republish) republie un message MQTT dans une autre rubrique MQTT.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'iot:Publish opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

headers

Informations sur les en-têtes MQTT version 5.0.

Pour plus d'informations, veuillez consulter les sections [RepublishAction](#) et [MqttHeaders](#) (français non garanti) de la Référence d'API AWS.

topic

Rubrique MQTT dans laquelle republier le message.

Pour republier dans une rubrique réservée, qui commence par \$, utilisez \$\$ plutôt. Par exemple, pour republier dans le sujet secondaire de l'appareil \$aws/things/MyThing/shadow/update, spécifiez le sujet sous \$\$aws/things/MyThing/shadow/update la forme.

Note

La republication vers des [sujets de travail réservés \(p. 124\)](#) n'est pas prise en charge. AWS IoT Device Defender les rubriques réservées ne prennent pas en charge la publication HTTP.

Supporte les [modèles de substitution \(p. 681\)](#) : 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 plus d'informations sur la QoS, veuillez consulter [MQTT \(p. 92\)](#)

Supporte les [modèles de substitution \(p. 681\)](#) : Non

roleArn

Le rôle IAM qui permet de AWS IoT publier dans la rubrique MQTT. Pour plus d'informations, veuillez consulter [Prérequis \(p. 595\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action de republication dans une AWS IoT règle.

```
{  
    "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 une AWS IoT règle.

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

L'exemple JSON suivant définit une action de republication à l'aide headers d'une AWS IoT règle.

```
{
  "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",
          "headers": {
            "payloadFormatIndicator": "UTF8_DATA",
            "contentType": "rule/contentType",
            "correlationData": "cnVsZSBjb3JyZWxhdGlvbiBkYXRh",
            "userProperties": [
              {
                "key": "ruleKey1",
                "value": "ruleValue1"
              },
              {
                "key": "ruleKey2",
                "value": "ruleValue2"
              }
            ]
          }
        }
      }
    ]
  }
}
```

S3

L'action S3 (s3) écrit les données d'un message MQTT dans un compartiment Amazon Simple Storage Service (Amazon S3).

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l's3:PutObject opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez un service AWS KMS géré par le client AWS KMS key pour chiffrer des données au repos dans Amazon S3, le service doit être autorisé à l'utiliser pour AWS KMS key le compte de l'appelant. Pour de plus amples informations, veuillez consulter [AWS Gestion AWS KMS keys et Gestion par le client AWS KMS keys](#) dans le Manuel du développeur Simple Storage Service.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

bucket

Le Compartiment Amazon S3 dans lequel écrire des données.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement
cannedacl

(Facultatif) Liste ACL prédéfinie Amazon S3 qui contrôle l'accès à l'objet identifié par la clé d'objet.
Pour plus d'informations, notamment sur les valeurs autorisées, consultez la section [ACL standardisée](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non
key

Chemin d'accès au fichier dans lequel les données sont écrites.

Prenons un exemple où ce paramètre se trouve \${topic()}/\${timestamp()} et où la règle reçoit un message indiquant le sujet some/topic. Si l'horodatage actuel est 1460685389, 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 IoT remplace un seul fichier chaque fois que la règle est invoquée. Nous vous recommandons d'utiliser l'horodatage du message ou un autre identifiant de message unique afin qu'un nouveau fichier soit enregistré dans Amazon S3 pour chaque message reçu.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui
roleArn

Rôle IAM qui autorise l'accès au compartiment Amazon S3. Pour plus d'informations, veuillez consulter [Prérequis \(p. 598\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action S3 dans une AWS IoT règle.

```
{
```

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

Consulter aussi

- [Qu'est-ce qu'Amazon S3 ?](#) dans le Guide de l'utilisateur Amazon Simple Storage Service

Salesforce IoT

L'action Salesforce IoT (`salesforce`) envoie les données du message MQTT qui a déclenché la règle à un flux d'entrée Salesforce IoT.

Paramètres

Lorsque vous créez une AWS IoT règle avec 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.

Supporte les [modèles de substitution \(p. 681\)](#) : 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.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action Salesforce IoT dans une AWS IoT règle.

```
{  
    "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

L'action SNS (sns) envoie les données d'un message MQTT sous la forme d'une notification push Amazon Simple Notification Service (Amazon SNS).

Vous pouvez suivre un didacticiel qui explique comment créer et tester une règle avec une action SNS. Pour plus d'informations, veuillez consulter [Tutorial : Envoi d'une notification Amazon SNS \(p. 221\)](#).

Note

L'action SNS ne prend pas en charge les rubriques [FIFO \(premier entré, premier sorti\)](#). Le moteur de règles étant un service entièrement distribué, l'ordre des messages n'est pas garanti lorsque l'action SNS est invoquée.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'sns:Publish opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez un service AWS KMS géré par le client AWS KMS key pour chiffrer les données au repos dans Amazon SNS, le service doit être autorisé à les utiliser pour le compte de AWS KMS key l'appelant. Pour de plus amples informations, veuillez consulter [Gestion des clés](#) dans le Manuel du développeur Simple Notification Service.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

targetArn

Rubrique SNS ou appareil individuel auxquels les notifications push sont envoyées.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

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 de travail spécifiques à la plateforme doivent être extraites. Valeurs valides : JSON, RAW. La valeur par défaut est RAW.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

roleArn

Rôle IAM qui fournit un accès au SNS. Pour plus d'informations, veuillez consulter [Prérequis \(p. 600\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action SNS dans une AWS IoT règle.

```
{  
    "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 une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce qu'Amazon Simple Notification Service ?](#) dans le Guide du développeur Amazon Simple Notification Service
- [Tutoriel : Envoi d'une notification Amazon SNS \(p. 221\)](#)

SQS

L'action SQS (sqsh) envoie des données d'un message MQTT vers une file d'attente Amazon Simple Queue Service (Amazon SQS).

Note

L'action SQS ne prend pas en [Amazon SQS les files d'attente FIFO \(premier entré, premier sorti\)](#). Le moteur de règles étant un service entièrement distribué, l'ordre des messages n'est pas garanti lorsque l'action SQS est déclenchée.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'sqs : SendMessage opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

- Si vous utilisez un service AWS KMS géré par le client AWS KMS key pour chiffrer les données au repos dans Amazon SQS, le service doit être autorisé à les utiliser pour le compte de AWS KMS key l'appelant. Pour plus d'informations, veuillez consulter [Gestion des clés](#) dans le Guide du développeur Amazon Simple Queue Service.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`queueUrl`

URL de la file d'attente Amazon SQS dans laquelle les données doivent être écrites.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`useBase64`

Définissez ce paramètre `true` pour configurer l'action de la règle afin de coder en base64 les données du message avant de les écrire dans la file d'attente Amazon SQS. La valeur par défaut est `false`.

Supporte les [modèles de substitution \(p. 681\)](#) : Non

`roleArn`

Le rôle IAM qui permet d'accéder à la file d'attente Amazon SQS. Pour plus d'informations, veuillez consulter [Prérequis \(p. 601\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action SQS dans une AWS IoT règle.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "sq": {  
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/  
my_sq_queue",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sq"  
                }  
            }  
        ]  
    }  
}
```

L'exemple JSON suivant définit une action SQS avec des modèles de substitution dans une AWS IoT règle.

```
{
```

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

Consulter aussi

- [Présentation d'Amazon Simple Queue Service](#) dans le Manuel du développeur Amazon Simple Queue Service

Step Functions

L'action Step Functions (`stepFunctions`) démarre une machine à AWS Step Functions états.

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut être assumé pour effectuer l'`states:StartExecution` opération. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir ou créer un rôle pour autoriser l'exécution AWS IoT de cette action de règle.

Paramètres

Lorsque vous créez une AWS IoT règle avec cette action, vous devez spécifier les informations suivantes :

`stateMachineName`

Nom de la machine d'état Step Functions à démarrer.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

`executionNamePrefix`

(Facultatif) Le nom donné à l'exécution de la machine d'état consiste en ce préfixe suivi d'un UUID. S'il n'est pas fourni, Step Functions crée un nom unique pour chaque exécution de machine d'état.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

`roleArn`

ARN du rôle qui accorde l'AWS IoT autorisation de démarrer la machine d'état. Pour plus d'informations, veuillez consulter [Prérequis \(p. 603\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

Exemples

L'exemple JSON suivant définit une action Step Functions dans une AWS IoT règle.

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

Consulter aussi

- [Qu'est-ce que AWS Step Functions ?](#)dans le Guide du développeur AWS Step Functions

Timestream

L'action relative à la règle Timestream écrit les attributs (mesures) d'un message MQTT dans une table Amazon Timestream. Pour plus d'informations sur Amazon Timestream, consultez [Qu'est-ce qu'Amazon Timestream ?](#).

Note

Amazon Timestream n'est pas disponible dans toutes les régions AWS. Si Amazon Timestream n'est pas disponible dans votre région, il n'apparaîtra pas dans la liste des actions relatives aux règles.

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 du résultat de l'instruction de requête est analysée pour en déduire son type de données (comme dans le cas d'une [the section called "DynamoDBv2" \(p. 548\)](#) 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 la [cast \(\) \(p. 638\)](#)fonction dans l'instruction de requête. Pour plus d'informations sur le contenu de chaque enregistrement Timestream, veuillez consulter. [the section called "Contenu d'un enregistrement de flux" \(p. 606\)](#)

Note

Avec SQL V2 (23/03/2016), les valeurs numériques qui sont des nombres entiers, par exemple 10.0, sont converties en leur représentation entière (10). Leur conversion explicite en une Decimal valeur, par exemple à l'aide de la fonction [cast \(\) \(p. 638\)](#), n'empêche pas ce comportement : le résultat reste une Integer valeur. Cela peut provoquer des erreurs de non-concordance de type qui empêchent l'enregistrement des données dans la base de données Timestream. Pour traiter des valeurs numériques entières en tant que Decimal valeurs, utilisez SQL V1 (08/10/2015) pour l'instruction de requête de règle.

Note

Le nombre maximum de valeurs qu'une action de règle Timestream peut écrire dans une table Amazon Timestream est de 100. Pour plus d'informations, consultez Référence [Amazon Timestream Quota](#).

Prérequis

Cette action de règle présente les exigences suivantes :

- Un rôle IAM qui AWS IoT peut assumer la responsabilité d'effectuer les `timestream:WriteRecords` opérations `timestream:DescribeEndpoints` et. Pour plus d'informations, veuillez consulter [Accorder à une AWS IoT règle l'accès dont elle a besoin \(p. 525\)](#).

Dans la AWS IoT console, vous pouvez choisir, mettre à jour ou créer un rôle pour autoriser l'exécution AWS IoT de cette action sur les règles.

- Si vous utilisez une solution gérée par le client AWS KMS pour crypter les données au repos dans Timestream, le service doit être autorisé à les utiliser pour le compte de l'AWS KMS keyappelant. Pour plus d'informations, voir [Comment les AWS services utilisent AWS KMS](#).

Paramètres

Lorsque vous créez une AWS IoT règle avec 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**.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

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`, `outime`. Ces noms sont réservés. Les noms de dimension ne peuvent pas commencer par `ts_` ni `measure_value` contenir le caractère deux-points (`:`).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

value

Valeur à écrire dans cette colonne de l'enregistrement de base de données.

Supporte les [modèles de substitution \(p. 681\)](#) : Oui

roleArn

L'Amazon Resource Name (ARN) du rôle qui accorde à AWS IoT l'autorisation d'écrire dans la table de la base de données Timestream. Pour plus d'informations, veuillez consulter [Prérequis \(p. 605\)](#).

Supporte les [modèles de substitution \(p. 681\)](#) : Non

tableName

Nom de la table de base de données dans laquelle écrire les enregistrements de mesures. Voir aussi **databaseName**.

Supporte les [modèles de substitution \(p. 681\)](#) : API et AWS CLI uniquement

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 cette [the section called “time_to_epoch \(Chaîne, Chaîne\)” \(p. 672\)](#) fonction pour créer un horodatage valide à partir d'une valeur de date ou d'heure transmise dans la charge utile du message.

Contenu d'un enregistrement de flux

Les données écrites dans la table Amazon Timestream par cette action incluent un horodatage, des métadonnées issues de l'action de la règle Timestream et le résultat de l'instruction de requête de la règle.

Pour chaque attribut (mesure) du résultat de l'instruction de 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
<i>nom de la dimension</i>	DIMENSION	La valeur spécifiée dans l'entrée d'action de la règle Timestream.	Chaque dimension spécifiée dans l'entrée d'action de la 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	Le nom de l'attribut dans le résultat de l'instruction de requête dont la valeur est spécifiée dans la measure_value: <i>:datatype</i> colonne.
<i>measure_value : type de données</i>	VALEUR_MESURE	La valeur de l'attribut dans le résultat de l'instruction de requête. Le nom de l'attribut figure dans la measure_name colonne.	La valeur est interprétée* et définie comme la correspondance la plus appropriée entre :bigint, boolean, double, ou varchar Amazon

Nom de la colonne	Type d'attribut	Valeur	Commentaires
			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 cast() (p. 638) fonction figurant dans l'instruction de requête de la règle.
time	TIMESTAMP	Date et heure de l'enregistrement dans la base de données.	Cette valeur est attribuée par le moteur de règles ou par la timestamp propriété, si elle est définie.

* La valeur d'attribut lire à partir de la charge utile du message est interprétée comme suit. Reportez-vous au [the section called “Exemples” \(p. 607\)](#) pour une illustration de chacun de ces cas.

- Une valeur sans guillemets de true ou false est interprétée comme un boolean type.
- Un nombre décimal est interprété comme un double type.
- Une valeur numérique sans virgule décimale est interprétée comme un bigint type.
- Une chaîne entre guillemets est interprétée comme un varchar type.
- Les objets et les valeurs des tableaux sont convertis en chaînes JSON et stockés en tant que varchar type.

Exemples

L'exemple JSON suivant définit une action de règle Timestream avec un modèle de substitution dans une règle AWS IoT

```
{
  "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 la rubrique Timestream définie dans l'exemple précédent avec la charge utile de message suivante génère les enregistrements Amazon Timestream écrits dans le tableau suivant.

```
{
    "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 et les enregistrements de la base de données créés à l'aide de l'action de règle de rubrique spécifiée pour traiter la charge utile du message précédent. Les colonnes `device_firmware_sku` et `device_id` sont les DIMENSIONS définies dans l'action de la règle du sujet. L'action de règle de rubrique Timestream crée la `time` colonne et les `measure_value` colonnes `measure_name` et, qu'elle remplit avec les valeurs issues du résultat de l'instruction de requête de l'action de règle de rubrique.

device_firmware_sku	device_id	nom_mesure	value_mesure	value_mesure	value_mesure	value_mesure	measure_value	time
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_complexe		{"simple_element": 42, "array_of_integers": [23,36,56,72], "tableau_de_chaînes": ["rouge", "vert", "bleu"]}	-	-	-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_entière sous forme de chaîne		123456789012	-	-	-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_booléenne		-	-	-	TRUE	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_entière	123456789012	-	-	-	-	2020

device_firmv	device_id	nom_mesure	value_mesur	value_mesur	value_mesur	measure_val	time
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_chaîne		Valeur de chaîne	-	-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	tableau_de_entiers		[23,36,56,72]-		-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	tableau de chaînes	-	["rouge ", " vert ", " bleu"]	-	-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	valeur_booléenne sous forme de chaîne		TRUE	-	-	2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	double_value-		-	123.456789012		2020
Mes métadonnées IoT - 159 statiques	Console Exemple 738-0	double valeur_sous forme de chaîne	-	123,45679	-	-	2020

Résolution des problèmes d'une règle

Si vous rencontrez un problème avec vos règles, nous vous recommandons d'activer CloudWatch les journaux. Vous pouvez analyser vos journaux pour déterminer si le problème est lié à l'autorisation ou si, par exemple, une condition de clause WHERE ne correspond pas. Pour plus d'informations, consultez [Configuration CloudWatch des journaux](#).

Accès aux ressources multicomptes à l'aide AWS IoT de règles

Vous pouvez configurer des AWS IoT règles d'accès entre comptes afin que les données ingérées sur les rubriques MQTT d'un compte puissent être acheminées vers les AWS services, tels qu'Amazon SQS et Lambda, d'un autre compte. Ce qui suit explique comment configurer des AWS IoT règles pour l'ingestion de données entre comptes, à partir d'une rubrique MQTT d'un compte vers une destination d'un autre compte.

Les règles multicomptes peuvent être configurées à l'aide d'[autorisations basées sur les ressources](#) sur 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 multicompte avec des règles. AWS IoT Les destinations prises en charge incluent Amazon SQS, Amazon SNS, Amazon S3 et. AWS Lambda

Note

Vous devez définir la règle de la même manière Région AWS que la ressource d'un autre service afin que l'action de la règle puisse interagir avec cette ressource. Pour plus d'informations sur les actions des AWS IoT règles, consultez la section [Actions des AWS IoT règles \(p. 531\)](#).

Prérequis

- [Connaissance des règles AWS IoT](#)
- Compréhension des [utilisateurs, des rôles et des autorisations basées sur les ressources d'IAM](#)
- Après avoir [AWS CLI installé](#)

Configuration multicompte pour Amazon SQS

Scénario : le compte A envoie les données d'un message MQTT à la file d'attente Amazon SQS du compte B.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action relative à la règle : sqs:SendMessage
2222-2222-2222	Compte B	File d'attente Amazon SQS <ul style="list-style-type: none">• ARN : arn:aws:sqs:region:2222-2222-2222:ExampleQueue• Adresse URL : https://sqs.region.amazonaws.com/2222-2222-2222/ExampleQueue

Tâches du compte A

Remarque

Pour exécuter les commandes suivantes, votre utilisateur IAM doit être autorisé `iot:CreateTopicRule` à utiliser le nom de ressource Amazon (ARN) de la règle comme ressource et à `iam:PassRole` agir avec une ressource comme ARN du rôle.

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte A.
2. Créez un rôle IAM qui fait confiance au moteur de AWS IoT règles et qui associe une politique autorisant l'accès à la file d'attente Amazon SQS du compte B. Consultez des exemples de commandes et de documents de politique dans la section [Octroi AWS IoT de l'accès requis](#).
3. Pour créer une règle associée à une rubrique, exécutez la [create-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 la `iot/test` rubrique dans la file d'attente Amazon SQS spécifiée. L'instruction SQL filtre les messages et le rôle ARN accorde AWS IoT les autorisations nécessaires pour ajouter le message à la file d'attente Amazon SQS.

```
{  
    "sql": "SELECT * FROM 'iot/test'",
```

```

    "ruleDisabled": false,
    "awsIoTSqlVersion": "2016-03-23",
    "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 une AWS IoT règle, consultez la section [Actions de AWS IoT règle - Amazon SQS](#).

Tâches du compte B

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte B.
2. Pour autoriser le compte A à accéder à la ressource de file d'attente Amazon SQS, exécutez la commande [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 intercompte pour Amazon SNS

Scénario : le compte A envoie des données d'un message MQTT vers une rubrique Amazon SNS du compte B.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action relative à la règle : sns:Publish
2222-2222-2222	Compte B	ARN de rubrique Amazon SNS : <i>arn:aws:sns:region:2222-2222-2222:ExampleTopic</i>

Tâches du compte A

Remarques

Pour exécuter les commandes suivantes, votre utilisateur IAM doit être autorisé à iot:CreateTopicRule utiliser l'ARN de la règle en tant que ressource et à iam:PassRole avec une ressource en tant qu'ARN de rôle.

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte A.
2. Créez un rôle IAM qui fait confiance au moteur de AWS IoT règles et qui associe une politique autorisant l'accès à la rubrique Amazon SNS du compte B. Pour des exemples de commandes et de documents de politique, voir [Octroi AWS IoT de l'accès requis](#).
3. Pour créer une règle associée à une rubrique, exécutez la [create-topic-rulecommande](#).

```
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 la `iot/test` rubrique dans la rubrique Amazon SNS spécifiée. L'instruction SQL filtre les messages et le rôle ARN accorde AWS IoT les autorisations nécessaires pour envoyer le message à la rubrique Amazon SNS.

```
{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "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 une AWS IoT règle, consultez la section [Actions relatives aux AWS IoT règles - Amazon SNS](#).

Tâches du compte B

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte B.
2. Pour autoriser le compte A à accéder à la ressource thématique Amazon SNS, exécutez la commande [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 intercompte pour Amazon S3

Scénario : le compte A envoie des données d'un message MQTT vers un compartiment Amazon S3 du compte B.

Compte AWS	Compte désigné comme	Description
<i>1111-1111-1111</i>	Compte A	Action relative à la règle : <code>s3:PutObject</code>
<i>2222-2222-2222</i>	Compte B	ARN de compartiment Amazon S3 : <i>arn:aws:s3:::ExampleBucket</i>

Tâches du compte A

Remarque

Pour exécuter les commandes suivantes, votre utilisateur IAM doit être autorisé à `iot:CreateTopicRule` utiliser la règle ARN en tant que ressource et à `iam:PassRole` agir avec une ressource en tant qu'ARN de rôle.

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte A.

2. Créez un rôle IAM qui fait confiance au moteur de AWS IoT règles et associe une politique autorisant l'accès au compartiment Amazon S3 du compte B. Pour des exemples de commandes et de documents de politique, voir [Octroi AWS IoT de l'accès requis](#).
3. Pour créer une règle associée à votre compartiment S3 cible, exécutez la [create-topic-rulecommande](#).

```
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 la `iot/test` rubrique dans le compartiment Amazon S3 spécifié. L'instruction SQL filtre les messages et le rôle ARN accorde AWS IoT les autorisations nécessaires pour ajouter le message au compartiment Amazon S3.

```
{  
    "sql": "SELECT * FROM 'iot/test'",  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23",  
    "actions": [  
        {  
            "s3": {  
                "bucketName": "ExampleBucket",  
                "key": "${topic()}/${timestamp()}",  
                "roleArn": "arn:aws:iam::1111-1111-1111:role/my-iot-role"  
            }  
        }  
    ]  
}
```

Pour de plus amples informations sur la définition d'une action Amazon S3 dans une AWS IoT règle, veuillez consulter [Actions de AWS IoT règles - Amazon S3](#).

Tâches du compte B

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte B.
2. Créez une politique de compartiment qui fait confiance au principal du compte A.

Voici un exemple de fichier de charge utile qui définit une politique de compartiment qui fait confiance au principal 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 plus d'informations, consultez des [exemples de politique de compartiment](#).

3. Pour associer la politique du bucket au bucket spécifié, exécutez la [put-bucket-policycommande](#).

```
aws s3api put-bucket-policy --bucket ExampleBucket --policy file://./my-bucket-policy.json
```

4. Pour que l'accès multicompte fonctionne, assurez-vous que les paramètres de blocage de tous les accès publics sont corrects. Pour plus d'informations, consultez [Bonnes pratiques de sécurité pour Amazon S3](#).

Configuration intercompte pour AWS Lambda

Scénario : Le compte A invoque une AWS Lambda fonction du compte B et transmet un message MQTT.

Compte AWS	Compte désigné comme	Description
1111-1111-1111	Compte A	Action relative à la règle : lambda:InvokeFunction
2222-2222-2222	Compte B	ARN de fonction Lambda : <i>arn:aws:lambda:region:2222-2222-2222:function:example-function</i>

Tâches du compte A

Remarques

Pour exécuter les commandes suivantes, votre utilisateur IAM doit être autorisé à iot:CreateTopicRule utiliser l'ARN de la règle en tant que ressource et à iam:PassRole agir avec la ressource en tant qu'ARN de rôle.

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte A.
2. Exécutez la [create-topic-rule commande](#) pour créer une règle qui définit l'accès multicompte 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 la iot/test rubrique dans la fonction Lambda spécifiée. L'instruction SQL filtre les messages et le rôle ARN AWS IoT autorise la transmission des données à la fonction Lambda.

```
{
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "lambda": {
        "functionArn": "arn:aws:lambda:region:2222-2222-2222:function:example-function"
      }
    }
  ]
}
```

Pour plus d'informations sur la façon de définir une AWS Lambda action dans une AWS IoT règle, consultez la section [Actions de la AWS IoT règle - Lambda](#).

Tâches du compte B

1. [Configurez AWS CLI](#) à l'aide de l'utilisateur IAM du compte B.
2. Exécutez la [commande add-permission de Lambda](#) pour autoriser les AWS IoT règles à activer la fonction Lambda. Pour exécuter la commande suivante, votre utilisateur IAM doit être autorisé à lambda:AddPermission agir.

```
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 autorise AWS IoT (représenté par `parrot.amazonaws.com`) à appeler la fonction Lambda.

--source-arn

Ce champ confirme que cette fonction Lambda est AWS IoT déclenchée uniquement `arn:aws:iot:region:1111-1111-1111:rule/example-rule` et qu'aucune autre règle du même compte ou d'un compte différent ne peut activer cette fonction Lambda.

--compte-source

Ce champ confirme que AWS IoT cette fonction Lambda est activée uniquement pour le `1111-1111-1111` compte.

Remarques

Si le message d'erreur « La règle est introuvable » s'affiche dans la console de votre AWS Lambda fonction sous Configuration, ignorez le message d'erreur et testez 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 activées, transmettant le résultat de l'instruction de requête.

Si un problème survient lors de l'activation d'une action, le moteur de règles active une action d'erreur, si une 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.
- Une erreur utilisateur entraîne le dépassement du débit provisionné de DynamoDB.

Note

La gestion des erreurs abordée dans cette rubrique concerne les [actions relatives aux règles \(p. 531\)](#). Pour résoudre les problèmes SQL, y compris les fonctions externes, vous pouvez configurer la AWS IoT journalisation. Pour plus d'informations, veuillez consulter [???](#) (p. 467).

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 présente 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: yMah1cwPhqTH267QLPhTKeVPKJB8B05ndBHz0mWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y=). Message arrived on: error/action, Action: s3, Bucket: us-east-1-s3-verify-user, Key: \"aaa\". Value of x-amz-id-2: yMah1cwPhqTH267QLPhTKeVPKJB8B05ndBHz0mWtxLTM6uAvwYYuqieAKyb6qRPTxP1tHXCoR4Y="  
        }  
    ]  
}
```

ruleName

Nom de la règle qui a déclenché l'action d'erreur.

topic

Rubrique dans laquelle le message d'origine a été reçu.

cloudwatchTraceld

Identité unique faisant référence aux journaux d'erreurs dans CloudWatch.

clientId

ID client de l'émetteur du message.

base64 OriginalPayload

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 les données des messages 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",  
            "bucketName" : "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 en vue de superviser la réussite ou l'échec des règles, consultez [Métriques et dimensions AWS IoT \(p. 478\)](#).

Réduire les coûts de messagerie grâce à Basic Ingest

[Vous pouvez utiliser Basic Ingest pour envoyer en toute sécurité les données des appareils aux personnes Services AWS prises en charge parActions de règle AWS IoT \(p. 531\), sans encourir de frais de messagerie.](#) Basic Ingest optimise le flux de données en supprimant le courtier de messages de publication/d'abonnement du chemin d'ingestion.

Basic Ingest peut envoyer des messages depuis vos appareils ou applications. Les noms des sujets des messages commencent par `$aws/rules/rule_name` « pour leurs trois premiers niveaux », où `rule_name` est le nom de la AWS IoT règle que vous souhaitez invoquer.

Vous pouvez utiliser une règle existante avec Basic Ingest en ajoutant le préfixe Basic Ingest (`$aws/rules/rule_name`) à la rubrique du message que vous utiliserez pour invoquer la règle. Par exemple, si vous avez nommé une règle `BuildingManager` qui est invoquée par des messages dont le sujet est tel que `Buildings/Building5/Floor2/Room201/Lights` (`"sql": "SELECT * FROM 'Buildings/#'"`), vous pouvez invoquer la même règle avec Basic Ingest en envoyant un message contenant un sujet `$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`

Remarque :

- Vos appareils et vos règles ne peuvent pas s'abonner aux rubriques réservées de Basic Ingest. Pour plus d'informations, veuillez consulter [Rubriques réservées \(p. 117\)](#).

- 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. Veillez toutefois à publier vos messages sur des sujets autres que ceux de Basic Ingest.

Utilisation de Basic Ingest

Avant d'utiliser Basic Ingest, vérifiez que votre appareil ou votre application utilise une [politique assortie \(p. 357\)](#) d'autorisations de publication. `$aws/rules/*` Vous pouvez également spécifier l'autorisation pour des règles individuelles `$aws/rules/rule_name/*` dans la politique. 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'y a aucune différence d'implémentation ou de gestion des erreurs entre les règles invoquées depuis Basic Ingest et celles invoquées via des abonnements à des courtiers de messages.

Vous pouvez créer des règles permettant à vos appareils d'interagir 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\(Decimal\) \(p. 673\)](#).
- Si vous définissez une règle qui est invoquée uniquement avec Basic Ingest, la `FROM` clause est facultative dans le `sql` champ de `rule` définition. Elle est toujours requise si la règle est également invoquée par d'autres messages qui doivent être envoyés via le courtier de messages (par exemple, parce que ces autres messages doivent être distribués à plusieurs abonnés). Pour plus d'informations, veuillez consulter [Référence SQL AWS IoT \(p. 618\)](#).
- Les trois premiers niveaux du sujet Basic Ingest (`$aws/rules/rule_name`) ne sont pas pris en compte dans la limite de 8 segments ni dans la limite de 256 caractères au total pour un sujet. Dans le cas contraire, les mêmes restrictions s'appliquent que celles décrites dans la section [AWS IoT Limites](#).
- Si vous recevez un message contenant une rubrique Basic Ingest qui spécifie une règle inactive ou une règle qui n'existe pas, un journal des erreurs est créé dans un CloudWatch journal Amazon pour vous aider à déboguer. Pour plus d'informations, veuillez consulter [Entrées de journal du moteur de règles \(p. 501\)](#). 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 \(p. 479\)](#).
- 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 relatives aux règles ont été effectuées avec succès. Vous pouvez configurer une action d'erreur pour gérer les erreurs lorsqu'une action est exécutée. Pour plus d'informations, veuillez consulter [Gestion des erreurs \(action d'erreur\) \(p. 615\)](#).

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 ces informations. Les messages à utiliser sont identifiés par le [filtre de sujet \(p. 116\)](#) spécifié dans la clause `FROM`.

La clause `SELECT` prend en charge [Types de données \(p. 622\)](#) [Opérateurs \(p. 626\)](#) [Fonctions \(p. 632\)](#) [Littéraux \(p. 678\)](#), [Instructions Case \(p. 679\)](#), [Extensions](#)

[JSON \(p. 680\)](#), [Modèles de substitution \(p. 681\)](#), [Requêtes d'objets imbriqués \(p. 683\)](#), et [Charges utiles binaires \(p. 684\)](#).

FROM

Le [filtre de sujet \(p. 116\)](#) de message MQTT qui identifie les messages dont on veut extraire des données. La règle est activée pour chaque message envoyé à une rubrique MQTT qui correspond au filtre de rubrique spécifié ici. Obligatoire pour les règles activées par les messages qui transitent par l'intermédiaire du courtier de messages. Facultatif pour les règles qui ne sont activées qu'à l'aide [de la fonctionnalité Basic Ingest \(p. 617\)](#).

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 \(p. 622\)](#), [Opérateurs \(p. 626\)](#), [Fonctions \(p. 632\)](#), [Littéraux \(p. 678\)](#), [Instructions Case \(p. 679\)](#), [Extensions JSON \(p. 680\)](#), [Modèles de substitution \(p. 681\)](#) et [Requêtes d'objets imbriqués \(p. 683\)](#).

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 \(p. 531\)](#).

Note

Les commentaires ne sont actuellement pas pris en charge dans la syntaxe AWS IoT SQL. Les noms d'attributs contenant des espaces ne peuvent pas être utilisés comme noms de champs dans l'instruction SQL. Bien que la charge utile entrante puisse comporter des noms d'attributs contenant des espaces, ces noms ne peuvent pas être utilisés dans l'instruction SQL. Ils seront toutefois transmis à la charge utile sortante si vous utilisez une spécification de nom de champ générique (*).

Clause SELECT

La clause SELECT AWS IoT est essentiellement la même que la clause SELECT SQL ANSI, avec certaines différences mineures.

La clause SELECT prend en charge [Types de données \(p. 622\)](#) [Opérateurs \(p. 626\)](#) [Fonctions \(p. 632\)](#) [Littéraux \(p. 678\)](#), [Instructions Case \(p. 679\)](#), [Extensions JSON \(p. 680\)](#), [Modèles de substitution \(p. 681\)](#), [Requêtes d'objets imbriqués \(p. 683\)](#), et [Charges utiles binaires \(p. 684\)](#).

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. Par exemple :

```
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. Par exemple :

```
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. Par exemple :

```
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. Par exemple :

```
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. Par exemple :

```
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. Par exemple :

```
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 chiffres ou le trait d'union (moins), voir [Extensions JSON \(p. 680\)](#)

Vous pouvez utiliser des fonctions (voir [Fonctions \(p. 632\)](#)) pour transformer la charge utile entrante. Vous pouvez utiliser des parenthèses pour le regroupement. Par exemple :

```
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 une [rubrique \(p. 115\)](#) ou à un [filtre de rubrique \(p. 116\)](#). Placez le sujet ou le filtre de sujet entre guillemets simples (''). La règle est déclenchée pour chaque message envoyé à une rubrique MQTT qui correspond au filtre de rubrique défini ici. Vous pouvez vous abonner à un groupe de sujets similaires à l'aide d'un filtre de sujets.

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 pas 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 quel sujet commençant par. Elle est donc exécutée trois fois 'topic', en envoyant des charges utiles sortantes de (pour le sujet/sous-sujet), {t: 50} (pour le sujet/sous-sujet-2) et {t: 60} (pour le sujet/sous-sujet-3/détails) vers ses actions. {t: 70} Il n'y a pas d'abonnement 'topic-2/subtopic-x', donc la règle n'est pas déclenchée pour le {temperature: 80} message.

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' et 'topic/subtopic-2', mais pas 'topic/subtopic-3/details' (elle comporte plus de niveaux que le filtre de sujet) ou 'topic-2/subtopic-x' (elle ne commence pas partopic).

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 \(p. 622\)](#), [Opérateurs \(p. 626\)](#), [Fonctions \(p. 632\)](#), [Littéraux \(p. 678\)](#), [Instructions Case \(p. 679\)](#), [Extensions JSON \(p. 680\)](#), [Modèles de substitution \(p. 681\)](#) et [Requêtes d'objets imbriqués \(p. 683\)](#).

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 la 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.)

Exemple avec une charge utile non JSON :

Charge utile entrante non JSON publiée sur le `sujet/sous-sujet` : '80'

SQL : `SELECT decode(encode(*, 'base64'), 'base64') AS value FROM 'topic/subtopic' WHERE decode(encode(*, 'base64'), 'base64') > 50`

Dans ce cas, la règle sera déclenchée et les actions spécifiées par la règle seront exécutées. La charge utile sortante sera transformée par la clause SELECT en tant que charge utile {"value": 80} JSON.

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	DiscretInt. 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.

Type	Signification
	<p>Note</p> <p>Certaines fonctions renvoient <code>Decimal</code> des valeurs avec une double précision au lieu d'une précision à 34 chiffres. Avec SQL V2 (23/03/2016), les valeurs numériques qui sont des nombres entiers, par exemple <code>10.0</code>, sont traitées comme une <code>Int</code> valeur (10) au lieu de la <code>Decimal</code> valeur attendue (.). <code>10.0</code> Pour traiter de manière fiable les valeurs numériques de nombres entiers en tant que <code>Decimal</code> valeurs, utilisez SQL V1 (08/10/2015) pour l'instruction de requête de règle.</p>
<code>Boolean</code>	True ou False.
<code>String</code>	Une chaîne UTF-8.
<code>Array</code>	Une série de valeurs qui ne sont pas nécessairement du même type.
<code>Object</code>	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.
<code>Null</code>	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 <code>Null</code> dans votre instruction SQL. Par exemple : "SELECT NULL AS n FROM 'topic/subtopic'"
<code>Undefined</code>	<p>Ce n'est pas une valeur. Non représenté dans JSON, sauf en omettant la valeur. Par exemple, dans l'objet <code>{"foo": null}</code>, la clé « <code>foo</code> » renvoie <code>NULL</code>, mais la clé « <code>bar</code> » renvoie <code>Undefined</code>. En interne, le langage SQL traite <code>Undefined</code> comme une valeur, mais il ne peut pas être représenté dans JSON, donc quand il est sérialisé au format JSON, les résultats sont <code>Undefined</code>.</p> <pre>{"foo":null, "bar":undefined}</pre> <p>est sérialisé au format JSON comme suit :</p> <pre>{"foo":null}</pre> <p>De même, <code>Undefined</code> 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 <code>Undefined</code>.</p>

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 utiliser explicitement 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.
Tableau	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 utiliser explicitement 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

Type d'argument	Résultat
	valeur String en Decimal, puis de tronquer les chiffres après la virgule de la valeur Decimal pour obtenir une valeur Int.
Tableau	Undefined.
Objet	Undefined.
Null	Null.
Non défini	Undefined.

En valeur booléenne

Type d'argument	Résultat
Int	Undefined. (Vous pouvez utiliser cette cast fonction de manière explicite pour transformer 0 = False, any_nonzero_value = True.)
Decimal	Undefined. (Vous pouvez utiliser explicitement 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.
Tableau	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.
Tableau	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.

Type d'argument	Résultat
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 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.

Opérateurs

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 <i>boolean AND boolean</i> .
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 opérandes est vrai. 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

Opérande gauche	Opérande droit	Sortie
		et traitées normalement en tant que valeurs <i>boolean</i> OR <i>boolean</i> .
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 la chaîne est « true » ou « false » (sans distinction entre majuscules et minuscules), elle est convertie en 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 de gauche est supérieur à l'opérande de droite. 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. True si l'opérande de gauche est supérieur ou égal à l'opérande de droite. 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 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 de gauche est inférieur à l'opérande de droite. 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. True si l'opérande de gauche est inférieur ou égal à l'opérande de droite. 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.

Opérande gauche	Opérande droit	Sortie
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.
Tableau	Tableau	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 apparié	Type non apparié	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.

Opérande gauche	Opérande droit	Sortie
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.
Tableau	Tableau	Vrai si les éléments de chaque opérande sont égaux et sont dans le même ordre. Sinon, la valeur renvoyée est Faux.
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 apparié	Type non apparié	Faux.

+ opérateur

Le signe « + » est un opérateur surchargé. Il peut être utilisé pour l'addition ou la concaténation de chaînes.

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.

Opérande gauche	Opérande droit	Sortie
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.
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 est renvoyée, Undefined.
Autre valeur	Autre valeur	Undefined.

Fonctions

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 pas être convertie, le résultat est Undefined.
Tableau	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 : $\text{acos}(0) = 1,5707963267948966$

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

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 : $\text{asin}(0) = 0,0$

Type d'argument	Résultat
<code>Int</code>	<code>Decimal</code> (avec double précision), le sinus inverse de l'argument. Des résultats imaginaires sont retournés sous la forme <code>Undefined</code> .
<code>Decimal</code>	<code>Decimal</code> (avec double précision), le sinus inverse de l'argument. Des résultats imaginaires sont retournés sous la forme <code>Undefined</code> .
<code>Boolean</code>	<code>Undefined</code> .

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

atan(Decimal)

Renvoie la tangente inverse d'un nombre en radians. Les arguments Decimal sont arrondis pour une meilleure prévision avant l'application de la fonction. 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.
Tableau	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 double et le point (x, y) spécifié)
Int/Decimal/String	Int/Decimal/String	Decimal, la chaîne ne peut pas être Undefined.
Autre valeur	Autre valeur	Undefined.

aws_lambda(functionArn, inputJson)

Appelle la fonction Lambda spécifiée en la transmettant `inputJson` à la fonction Lambda et renvoie le JSON généré par la fonction Lambda.

Arguments

Argument	Description
<code>functionArn</code>	ARN de la fonction Lambda à appeler. La fonction Lambda doit renvoyer des données JSON.
<code>inputJson</code>	Entrée JSON transmise à la fonction Lambda. Pour transmettre des requêtes et des littéraux d'objets imbriqués, vous devez utiliser SQL version 2016-03-23.

Vous devez accorder AWS IoT lambda:InvokeFunction les autorisations nécessaires 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 politique de ressources de la fonction.

`--region`

Le Région AWS de votre compte.

`--principal`

Mandataire qui obtient l'autorisation. Cela devrait `iot.amazonaws.com` permettre d'appeler AWS IoT une fonction Lambda.

`--source-arn`

ARN de la règle. Vous pouvez utiliser la `get-topic-rule` AWS CLI commande pour obtenir l'ARN d'une règle.

--source-account

L'Compte AWSendroit où la règle est définie.

--statement-id

Identifiant unique de l'instruction.

--action

Action Lambda que vous souhaitez autoriser dans cette instruction. Pour autoriser AWS IoT l'appel d'une fonction Lambda, spécifiez. lambda:InvokeFunction

Important

Si vous ajoutez une autorisation pour un AWS IoT principal sans fournir le source-arn ousource-account, toute personne Compte AWS qui crée une règle avec votre action Lambda peut déclencher des règles à partir desquelles votre fonction Lambda sera invoquée. AWS IoT Pour plus d'informations, consultez Modèle d'[autorisation Lambda](#).

Soit une charge utile de message JSON comme suit :

```
{  
    "attribute1": 21,  
    "attribute2": "value"  
}
```

La aws_lambda fonction 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 la charge utile complète du message MQTT, vous pouvez spécifier la charge utile JSON à l'aide de « * », comme dans l'exemple suivant.

```
SELECT  
aws_lambda("arn:aws:lambda:us-east-1:account_id:function:lambda_function", *) as output  
FROM 'topic-filter'
```

payload.inner.elementsélectionne les données des messages publiés sur le sujet « sujet/sous-sujet ».

some.valuesélectionne 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 AND des deux arguments.
Int/Decimal	Int/Decimal	Int, une opération AND des deux arguments. Tous les nombres doivent être convertis en Int. Si la valeur Int inférieure à 0 ou supérieure à 4294967295, le résultat est Undefined.
Int/Decimal/String	Int/Decimal/String	Int, une opération AND des deux arguments. Toutes les valeurs sont converties en Int. Les décimales et arrondies au chiffre le plus proche. Si la conversion échoue, le résultat est 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 OR des deux arguments.
Int/Decimal	Int/Decimal	Int, une opération OR des deux arguments. Tous les nombres doivent être convertis en Int. Si la valeur Int inférieure à 0 ou supérieure à 4294967295, le résultat est Undefined.
Int/Decimal/String	Int/Decimal/String	Int, une opération OR des deux arguments. Toutes les valeurs sont converties en Int. Les décimales et arrondies au chiffre le plus proche. Si la conversion échoue, le résultat est 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 XOR des deux arguments.

Type d'argument	Type d'argument	Résultat
Int/Decimal	Int/Decimal	Int, une opération XOR entre les deux arguments. Les nombres sont convertis en Int inférieure la plus proche.
Int/Decimal/String	Int/Decimal/String	Int, un XOR au niveau de l'argument. Les chaînes sont converties en valeurs Int inférieures. Si une conversion échoue, le résultat est Undefined.
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.
Texte	Il convertit une valeur en String.

Mot clé	Résultat
Ntext	Il convertit une valeur en String.
varchar	Il convertit une valeur en String.
Int	Il convertit une valeur en Int.
Entier	Il convertit une valeur en Int.
Double	Convertit la valeur en Decimal (avec une 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	<p>La valeur source.</p> <p>Note</p> <p>Avec SQL V2 (23/03/2016), les valeurs numériques qui sont des nombres entiers, par exemple 10.0, renvoient une Int valeur (10) au lieu de la Decimal valeur attendue (.). 10.0 Pour convertir de manière fiable des valeurs numériques de nombres entiers en Decimal valeurs, utilisez SQL V1 (08/10/2015) pour l'instruction de requête de règle.</p>
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.
Tableau	Undefined.
Objet	Undefined.
Null	Undefined.
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.
Tableau	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.
Tableau	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.
Decimal	Une chaîne représentant la valeur Decimal, probablement en notation scientifique.
Boolean	« true » ou « false », tout en minuscules.

Type d'argument	Résultat
String	« true »=True et « false »=False (insensible à la casse). Autres valeurs de chaînes = Undefined.
Tableau	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 Decimal et arrondie à la valeur la plus proche Int. Si la chaîne ne peut pas être convertie en une valeur Decimal, le résultat est Undefined.
Autre valeur	Undefined.

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 <code>Undefined</code> .
Decimal	Le caractère correspondant à la valeur ASCII spécifiée. L'argument <code>Decimal</code> est arrondi à la valeur <code>Int</code> inférieure la plus proche. Si l'argument n'est pas une valeur ASCII valide, le résultat est <code>Undefined</code> .
Boolean	<code>Undefined</code> .
String	Si la valeur <code>String</code> peut être convertie en valeur <code>Decimal</code> , elle est arrondie à la valeur <code>Int</code> inférieure la plus proche. Si l'argument n'est pas une valeur ASCII valide, le résultat est <code>Undefined</code> .
Tableau	<code>Undefined</code> .
Objet	<code>Undefined</code> .
Null	<code>Undefined</code> .
Autre valeur	<code>Undefined</code> .

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. Les arguments sont convertis en chaînes à l'aide des conversions standard répertoriées précédemment.

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.
Tableau	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.
Tableau	Undefined.
Objet	Undefined.
Null	Undefined.
Non défini	Undefined.

décoder (valeur, DecodingScheme)

Utilisez la decode fonction pour décoder une valeur codée. Si la chaîne décodée est un document JSON, un objet adressable est renvoyé. Sinon, la chaîne décodée est renvoyée sous la forme d'une chaîne. La fonction renvoie la valeur NULL si la chaîne ne peut pas être décodée. Cette fonction prend en charge le décodage des chaînes codées en base64 et le format de message Protocol Buffer (protobuf).

Pris en charge par SQL 2016-03-23 et versions ultérieures.

value

Une valeur de chaîne ou l'une des expressions valides, telles que définies dans [Référence SQL AWS IoT \(p. 618\)](#), qui renvoient une chaîne.

Schéma de décodage

Chaîne littérale représentant le schéma utilisé pour décoder la valeur. Actuellement, seuls 'base64' et 'proto' sont pris en charge.

Décoder des chaînes codées en Base64

Dans cet exemple, la charge utile du message inclut une valeur codée.

```
{
    encoded_temp: "eyAidGVtcGVyYXR1cmUiOiAzMyB9Cg=="
}
```

La decode fonction de cette instruction SQL décode la valeur de la charge utile du message.

```
SELECT decode(encoded_temp,"base64").temperature AS temp from 'topic/subtopic'
```

Le décodage de la `encoded_temp` valeur produit le document JSON valide suivant, qui permet à l'instruction SELECT de lire la valeur de température.

```
{ "temperature": 33 }
```

Le résultat de l'instruction SELECT dans cet exemple est affiché ici.

```
{ "temp": 33 }
```

Si la valeur décodée n'était pas un document JSON valide, la valeur décodée serait renvoyée sous forme de chaîne.

Décodage de la charge utile du message protobuf

Vous pouvez utiliser la fonction SQL de décodage pour configurer une règle capable de décoder la charge utile de votre message protobuf. Pour plus d'informations, consultez [Décoder les charges utiles des messages protobuf. \(p. 685\)](#)

La signature de la fonction se présente comme suit :

```
decode(<ENCODED DATA>, 'proto', '<S3 BUCKET NAME>', '<S3 OBJECT KEY>', '<PROTO NAME>',  
'<MESSAGE TYPE>')
```

ENCODED DATA

Spécifie les données codées en protobuf à décoder. Si l'intégralité du message envoyé à la règle est constituée de données codées en protobuf, vous pouvez référencer la charge utile binaire brute entrante à l'aide de. * Dans le cas contraire, ce champ doit être une chaîne JSON codée en base 64 et une référence à cette chaîne peut être transmise directement.

1) Pour décoder un protobuf binaire brut de la charge utile entrante :

```
decode(*, 'proto', ...)
```

2) Pour décoder un message codé en protobuf représenté par une chaîne codée en base64 « a.b » :

```
decode(a.b, 'proto', ...)
```

proto

Spécifie les données à décoder dans un format de message protobuf. Si vous spécifiez base64 plutôt que proto, cette fonction décodera les chaînes codées en base64 au format JSON.

S3 BUCKET NAME

Le nom du compartiment Amazon S3 dans lequel vous avez chargé votre `FileDescriptorSet` fichier.

S3 OBJECT KEY

Clé d'objet qui spécifie le `FileDescriptorSet` fichier dans le compartiment Amazon S3.

PROTO NAME

Le nom du .proto fichier (à l'exclusion de l'extension) à partir duquel le `FileDescriptorSet` fichier a été généré.

MESSAGE TYPE

Le nom de la structure de message protobuf du `FileDescriptorSet` fichier, à laquelle les données à décoder doivent être conformes.

Un exemple d'expression SQL utilisant la fonction SQL de décodage peut ressembler à ce qui suit :

```
SELECT VALUE decode(*, 'proto', 's3-bucket', 'messageformat.desc', 'myproto',
'messagetype') FROM 'some/topic'
```

- *

Représente une charge utile binaire entrante, conforme au type de message protobuf appelé.
mymessagetype

- messageformat.desc

Le `FileDescriptorSet` fichier stocké dans un compartiment Amazon S3 nommé `s3-bucket`.

- myproto

Le `.proto` fichier d'origine utilisé pour générer le `FileDescriptorSet` fichier nommé `myproto.proto`.

- messagetype

Le type de message appelé `messagetype` (ainsi que toutes les dépendances importées) tel que défini dans `myproto.proto`.

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.

`value`

Une des expressions valides, telles que définies dans la [Référence SQL AWS IoT \(p. 618\)](#).

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
<code>String</code>	<code>String</code>	Vrai si le premier argument est une chaîne et que la dernière partie de cette chaîne correspond à la valeur de l'argument 2. Sinon, la valeur <code>Undefined</code> .
Autre valeur	Autre valeur	Les deux arguments doivent être de types compatibles pour pouvoir être comparés. Les règles de conversion sont appliquées pour convertir les types si nécessaire. La comparaison se termine dans le sens où l'opération réussit si les deux valeurs sont égales.

Type d'argument 1	Type d'argument 2	Résultat
		renvoyée est Faux. Si Undefined, le résultat

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 Decimal et arrondie à la valeur inférieure la plus procheInt. 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)= « un »
```

Type d'argument 1	Type d'argument 2	Résultat
Tableau	Tout type (converti valeur Int)	L'élément à l'index de la valeur fourni par le deuxième argument. Si la conversion échoue, le résultat est en dehors des limites de l'array (array.length), le résultat sera Undefined.
Chaîne	Tout type (converti valeur Int)	Le caractère est à l'index de la valeur fournie par le deuxième argument. Si la conversion échoue, le résultat est en dehors des limites de la chaîne (string.length), le résultat sera Undefined.
Objet	String (aucune conversion appliquée)	La valeur stockée dans l'objet correspondant à la clé de l'argument.
Autre valeur	N'importe quelle valeur	Undefined.

get_dynamodb (TableName,,, partitionKeyName roleArn) partitionKeyValue sortKeyName sortKeyValue

Récupère les données à partir 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`

Le nom de la table DynamoDB à interroger.

`partitionKeyName`

Nom de la clé de partition. Pour plus d'informations, consultez [DynamoDB Keys](#).

`partitionKeyValue`

Valeur de la clé de partition utilisée pour identifier un enregistrement. Pour plus d'informations, consultez [DynamoDB Keys](#).

`sortKeyName`

(Facultatif) Le nom de la clé de tri. Ce paramètre est requis uniquement si la table DynamoDB interrogée utilise une clé composée. Pour plus d'informations, consultez [DynamoDB Keys](#).

`sortKeyValue`

(Facultatif) La valeur de la clé de tri. Ce paramètre est requis uniquement si la table DynamoDB interrogée utilise une clé composée. Pour plus d'informations, consultez [DynamoDB Keys](#).

roleArn

ARN d'un rôle IAM qui accorde l'accès à la table DynamoDB. Le moteur de règles endosse 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 politique 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"  
        }  
    ]  
}
```

À titre d'exemple d'utilisation `get_dynamodb()`, disons que vous disposez d'une table DynamoDB qui contient l'identifiant de l'appareil et les informations de localisation de tous vos appareils connectés. AWS 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_mqtt_property (nom)

Fait référence à l'un des en-têtes MQTT5 suivants :`contentType`,`payLoadFormatIndicator`,`responseTopic` et`correlationData`. Cette fonction prend l'une des chaînes littérales suivantes comme argument :`content_type`,`format_indicator`,`response_topic`, et`correlation_data`. Pour plus d'informations, consultez le tableau d'arguments de fonction suivant.

Type de contenu

Chaîne : chaîne codée en UTF-8 qui décrit le contenu du message de publication.
`payLoadFormatIndicator`

Chaîne : valeur de chaîne Enum qui indique si la charge utile est formatée en UTF-8. Les valeurs valides sont `UNSPECIFIED_BYTES` et `UTF8_DATA`.

Réponse au sujet

Chaîne : chaîne codée en UTF-8 utilisée comme nom de rubrique pour un message de réponse. La rubrique de réponse permet de décrire la rubrique dans laquelle le récepteur doit effectuer la publication dans le cadre du flux demande-réponse. La rubrique ne doit pas contenir de caractères génériques.

Données de corrélation

Chaîne : données binaires codées en base64 utilisées par l'expéditeur du message de demande pour identifier la demande à laquelle correspond le message de réponse au moment de sa réception.

Le tableau suivant présente les arguments de fonction acceptables et les types de retour associés pour la `get_mqtt_property` fonction :

Arguments de la fonction

SQL	Type de données renvoyé (le cas échéant)	Type de données renvoyé (s'il n'est pas présent)
<code>get_mqtt_property("format_indicator")</code> (UNSPECIFIED_BYTES ou UTF8_DATA)	Chaîne (UNSPECIFIED_BYTES)	
<code>get_mqtt_property("content_type")</code>	Non défini	
<code>get_mqtt_property("response_topic")</code>	Non défini	
<code>get_mqtt_property("correlation_data")</code> Chaîne codée en base64	Non défini	
<code>get_mqtt_property("some_invisible_mime")</code>	Non défini	

L'exemple de règles SQL suivant fait référence à l'un des en-têtes MQTT5 suivants :`contentType`, `payloadFormatIndicator`, `responseTopic` et `correlationData`

```
SELECT *, get_mqtt_property('content_type') as contentType,
       get_mqtt_property('format_indicator') as payloadFormatIndicator,
       get_mqtt_property('response_topic') as responseTopic,
       get_mqtt_property('correlation_data') as correlationData
FROM 'some/topic'
```

get_secret (secretId, SecretType, clé, roleArn)

Récupère la valeur du `SecretBinary` champ crypté `SecretString` ou de la version actuelle d'un secret dans [AWS Secrets Manager](#). Pour plus d'informations sur la création et la gestion de secrets [CreateSecret](#), voir [UpdateSecret](#), et [PutSecretValue](#).

`get_secret()` accepte les paramètres suivants :

secretId

Chaîne : Amazon Resource Name (ARN) ou nom convivial du secret à récupérer.

Type de secret

String : type secret. Valeurs valides : `SecretString` | `SecretBinary`.

`SecretString`

- Pour les secrets que vous créez sous forme d'objets JSON à l'aide des API AWS CLI, de ou de la AWS Secrets Manager console :
 - Si vous spécifiez une valeur pour le `key` paramètre, cette fonction renvoie la valeur de la clé spécifiée.
 - Si vous ne spécifiez pas de valeur pour le `key` paramètre, cette fonction renvoie l'objet JMS dans son intégralité.
- Pour les secrets que vous créez en tant qu'objets non JSON à l'aide des API ou des AWS CLI :

- Si vous spécifiez une valeur pour le key paramètre, cette fonction échoue avec une exception.
- Si vous ne spécifiez pas de valeur pour le key paramètre, cette fonction renvoie le contenu du secret.

SecretBinary

- Si vous spécifiez une valeur pour le key paramètre, cette fonction échoue avec une exception.
- Si vous ne spécifiez pas de valeur pour le key paramètre, cette fonction renvoie la valeur secrète sous la forme d'une chaîne UTF-8 codée en base64.

key

(Facultatif) Chaîne : nom de clé contenu dans 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 récupérer l'intégralité de l'objet JSON.

Si vous spécifiez une valeur pour ce paramètre et que le secret ne contient aucun objet JSON dans son SecretString champ, cette fonction échoue à une exception près.

roleArn

Chaîne : rôle ARN avec secretsmanager:GetSecretValue autorisations.

Note

Cette fonction renvoie toujours la version actuelle du secret (la version avec la AWSCURRENT balise). Le moteur de AWS IoT règles met en cache chaque secret pendant 15 minutes. Par conséquent, le moteur de règles peut mettre 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 avec AWS Secrets Manager, cette fonction peut renvoyer la version précédente.

Cette fonction n'est pas mesurée, mais des AWS Secrets Manager frais s'appliquent. En raison du mécanisme de mise en cache secret, le moteur de règles appelle AWS Secrets Manager parfois.

Le moteur de règles étant un service entièrement distribué, il est possible que vous receviez plusieurs appels d'API Secrets Manager provenant du moteur de règles pendant la fenêtre de mise en cache de 15 minutes.

Exemples :

Vous pouvez utiliser la get_secret fonction dans un en-tête d'authentification dans le cadre d'une action de règle HTTPS, comme dans l'exemple d'authentification par clé d'API suivant.

```
"API_KEY": "${get_secret('API_KEY', 'SecretString', 'API_KEY_VALUE',  
'arn:aws:iam::12345678910:role/getsecret')}"
```

Pour plus d'informations sur l'action de la règle HTTPS, consultez[the section called "HTTP" \(p. 551\)](#).

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 l'objet dont vous souhaitez récupérer le shadow.

shadowName

(Facultatif) Chaîne : nom de l'ombre. Ce paramètre est requis uniquement quand vous référez des shadows nommés.

roleArn

Chaîne : un ARN de rôle avec une autorisation `iot:GetThingShadow`.

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'
```

get_user_properties () userPropertyKey

Fait référence aux propriétés utilisateur, qui sont l'un des types d'en-têtes de propriétés pris en charge dans MQTT5.

Propriété de l'utilisateur

Chaîne : une propriété utilisateur est une paire clé-valeur. Cette fonction prend la clé comme argument et renvoie un tableau de toutes les valeurs correspondant à la clé associée.

Arguments de fonction

Pour les propriétés utilisateur suivantes dans les en-têtes des messages :

Clé	Valeur
une clé	une certaine valeur
une clé différente	une valeur différente
une clé	valeur avec clé dupliquée

Le tableau suivant présente le comportement SQL attendu :

SQL	Type de données renvoyé	Valeur des données renvoyées
<code>get_user_properties (« une clé »)</code>	Tableau de chaînes	<code>['some value', 'value with duplicate key']</code>
<code>get_user_properties (« autre clé »)</code>	Tableau de chaînes	<code>['a different value']</code>

SQL	Type de données renvoyé	Valeur des données renvoyées
get_user_properties ()	Tableau de paires clé-valeur Objects	[{"some key": "some value"}, {"other key": "a different value"}, {"some key": "value with duplicate key"}]
get_user_properties (« clé inexistante »)	Non défini	

L'exemple de règles SQL suivant fait référence aux propriétés utilisateur (un type d'en-tête de propriété MQTT5) dans la charge utile :

```
SELECT *, get_user_properties('user defined property key') as userProperty
FROM 'some/topic'
```

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`

In(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.
Tableau	Undefined.
Objet	Undefined.
Null	Undefined.
Non défini	Undefined.

log(Decimal)

Renvoie le logarithme 10 de base de l'argument Les arguments Decimal sont arrondis pour une meilleure prévision avant l'application de la fonction. 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.
Tableau	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
<code>String</code>	<code>Int</code>	<code>String</code> , l'argument <code>String</code> suivi d'un nombre d'espaces
<code>String</code>	<code>Decimal</code>	L'argument <code>Decimal</code> arrondi à la valeur la plus proche, et l'argument <code>String</code> complété à gauche par le nombre d'espaces
<code>String</code>	<code>String</code>	Le deuxième argument <code>String</code> qui est arrondie à la valeur la plus proche, et l'argument <code>String</code> suivi d'espaces spécifié. Si l'argument <code>String</code> n'est pas converti en une valeur
Autre valeur	<code>Int/Decimal/String</code>	La première valeur convertie à l'aide des conversions spécifiées et appliquée sur cette valeur. Si la valeur n'est pas convertie, le résultat est Undefined.
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	StringReprésentation du booléen (« vrai » ou « faux ») avec tous les espaces blancs de début supprimés.
String	L'argument avec tous les espaces de début supprimés.
Tableau	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, roleArn, enregistrement)

Utilisez la machinelearning_predict fonction pour établir des prévisions à l'aide des données d'un message MQTT basé sur un SageMaker modèle Amazon. 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

Rôle IAM doté d'une politique et d'machinelearning:GetMLModelAutorisations machinelearning:Predict et permettant d'accéder au modèle par rapport auquel la prédiction est exécutée.

record

Les données à transmettre dans l'API de prévision SageMaker. 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 SageMaker pour faire des prévisions. La valeur doit être SGD.

predictedScores

Contient le score de classification brut correspondant à chaque étiquette.

predictedValue

La valeur prévue par SageMaker.

mod(Decimal, Decimal)

Renvoie le reste résultant de la division du premier argument par le deuxième argument. Équivalent à [remainder\(Decimal, Decimal\) \(p. 664\)](#). Vous pouvez également utiliser « % » comme opérateur infixé 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 deux derniers arguments doivent être de type Int. Si vous voulez exécuter une division réelle, utilisez la fonction division (/).
Int/Decimal	Int/Decimal	Decimal, le premier argument doit être de type Int ou Decimal et le deuxième argument doit être de type Decimal. Pour lesquels vous voulez exécuter une division réelle, utilisez la fonction division (/).
String/Int/Decimal	String/Int/Decimal	Si toutes les chaînes sont converties en nombres, le résultat est le premier argument. Sinon la valeur null.
Autre valeur	Autre valeur	Undefined.

nombril (,) 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 argument.
Null	N'importe quelle valeur	Le deuxième argument.
Decimal (NaN)	N'importe quelle valeur	Le deuxième argument.

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 (Chaîne, Longue [, Chaîne])

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, reportez-vous à la section. [time_to_epoch \(Chaîne, Chaîne\) \(p. 672\)](#)

La `parse_time` fonction attend les arguments suivants :

pattern

(Chaîne) Un modèle de date/heure qui suit les formats [Joda-Time](#).

timestamp

(Long) Heure à formater en millisecondes depuis l'époque Unix. Voir la fonction [timestamp\(\) \(p. 673\)](#).

timezone

(Chaîne) Fuseau horaire de la date/heure formatée. 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:role/S3Role"
                }
            }
        ]
    }
}
```

```

        "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 la valeur Int.
Int/Decimal/String	Int/Decimal/String	Une valeur Decimal argument renvoyé à la valeur String. Toutes les chaînes sont converties en valeur String échouant le résultat est Undefined.
Autre valeur	Autre valeur	Undefined.

principal()

Renvoie le principal que l'appareil utilise pour l'authentification, en fonction de la manière dont le message déclencheur a été publié. Le tableau suivant décrit le mandataire renvoyé pour chaque méthode et protocole de publication.

Comment le message est publié	Protocole	Type d'informations de principal
Client MQTT	MQTT	Certificat d'appareil X.509
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.509
SDK pour les appareils AWS IoT	MQTT terminé WebSocket	Utilisateur ou rôle IAM

Les exemples suivants montrent les différents types de valeurs `principal()` pouvant être renvoyées :

- Empreinte du certificat X.509 :
ba67293af50bf2506f5f93469686da660c7c844e7b3950bfb16813e0d31e9373
- ID du rôle IAM et nom de session : ABCD1EFG3HIJK2LMN0P5:my-session-name
- Renvoie un ID utilisateur : ABCD1EFG3HIJK2LMN0P5

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 String représentation du booléen (« vrai » ou « faux »).
String	La valeur String.
Tableau	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 String représentation du booléen (« vrai » ou « faux »).
String	La valeur source.
Tableau	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.

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 String représentation du booléen (« vrai » ou « faux »).
String	L'argument String.
Tableau	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`.

remainder(Decimal, Decimal)

Renvoie le reste résultant de la division du premier argument par le deuxième argument. Équivalent à [`mod\(Decimal, Decimal\) \(p. 658\)`](#). Vous pouvez également utiliser « % » comme opérateur infixé 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
<code>Int</code>	<code>Int</code>	<code>Int</code> , les premiers et derniers éléments de la chaîne que vous voulez exécuter.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Decimal</code> , le premier et derniers éléments de la chaîne pour lesquels vous voulez exécuter Modulo.
<code>String/Int/Decimal</code>	<code>String/Int/Decimal</code>	Si toutes les chaînes sont des nombres, le résultat est le premier argument. Sinon la valeur <code>Undefined</code> .
Autre valeur	Autre valeur	<code>Undefined</code> .

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
<code>Int</code>	La représentation <code>String</code> de la valeur <code>Int</code> .
<code>Decimal</code>	La représentation <code>String</code> de la valeur <code>Decimal</code> .
<code>Boolean</code>	La <code>String</code> représentation du booléen (« vrai » ou « faux »).
<code>String</code>	La valeur source.
<code>Tableau</code>	La représentation <code>String</code> de la valeur <code>Array</code> (à l'aide des règles de conversion standard).
<code>Objet</code>	La représentation <code>String</code> de l'objet (à l'aide des règles de conversion standard).
<code>Null</code>	<code>Undefined</code> .

Type d'argument	Résultat
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 d'espaces égal à la valeur

Type d'argument 1	Type d'argument 2	Résultat
		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 String représentation du booléen (« vrai » ou « faux »).
Tableau	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 est renvoyée. Lorsque le signe de l'argument est négatif, la valeur -1 est 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.
Tableau	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.
Tableau	Undefined.
Objet	Undefined.
Null	Undefined.
Non défini	Undefined.

sous-chaîne (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 est 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.
Tableau	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 commence par la deuxième chaîne
Autre valeur	Autre valeur	Les deux arguments doivent être de type chaîne. Si l'une des deux chaînes n'est pas de type chaîne, ou si l'une des deux chaînes est Null ou Undefined, le résultat est Undefined.

tan(Decimal)

Renvoie la tangente d'un nombre en radians. Les valeurs Decimal sont arrondies 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.
Tableau	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.
Tableau	Undefined.
Objet	Undefined.
Null	Undefined.
Non défini	Undefined.

time_to_epoch (Chaîne, Chaîne)

Utilisez la `time_to_epoch` fonction pour convertir une chaîne d'horodatage en millisecondes à l'époque Unix. Pris en charge par SQL 2016-03-23 et versions ultérieures. Pour convertir des millisecondes

en une chaîne d'horodatage formatée, reportez-vous à la section [parse_time \(Chaîne, Longue \[, Chaîne\]\) \(p. 659\)](#).

La `time_to_epoch` fonction attend les arguments suivants :

`timestamp`

(Chaîne) La 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`

(Chaîne) Un modèle de date/heure qui suit les formats d'heure [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 \(p. 617\)](#) 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"
```

transformation (chaîne, objet, tableau)

Renvoie un tableau d'objets contenant le résultat de la transformation spécifiée du Object paramètre sur le 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 manière dont ils créent les paramètres Result à partir Object des Array paramètres et.

Objet

Un objet qui contient les attributs à appliquer à chaque élément duArray.

Tableau

Tableau d'objets auxquels les attributs de Object sont appliqués.

Chaque objet de ce tableau correspond à un objet de 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 tels que déterminés par le mode de transformation spécifié dansString.

String paramètre	Object paramètre	Array paramètre	Résultat
enrichArray	Objet	Tableau d'objets	Tableau d'objets dans lequel chaque objet contient les attributs d'un élément à partir du Array paramètre et les attributs du Object paramètre.
Toute autre valeur	N'importe quelle valeur	N'importe quelle valeur	Non défini

Note

Le tableau renvoyé par cette fonction est limité à 128 KiB.

Exemple 1 de fonction de transformation

Cet exemple montre comment la transform() fonction produit un tableau unique d'objets à partir d'un objet de données et d'un tableau.

Dans cet exemple, le message suivant est publié dans la rubrique A/B MQTT.

```
{
  "attributes": {
    "data1": 1,
    "data2": 2
  },
  "values": [
    {
      "value": 3
    }
  ]
}
```

```

        "a": 3
    },
    {
        "b": 4
    },
    {
        "c": 5
    }
]
}

```

Cette instruction SQL pour une action sur une règle de rubrique utilise la transform() fonction avec une String valeur de enrichArray. Dans cet exemple, Object il s'agit de la attributes propriété de la charge utile du message et Array du values tableau, qui contient trois objets.

```
select value transform("enrichArray", attributes, values) from 'A/B'
```

À la réception de la charge utile du message, l'instruction SQL génère la réponse suivante.

```

[
{
    "a": 3,
    "data1": 1,
    "data2": 2
},
{
    "b": 4,
    "data1": 1,
    "data2": 2
},
{
    "c": 5,
    "data1": 1,
    "data2": 2
}
]

```

Exemple 2 de fonction de transformation

Cet exemple montre comment la transform() fonction peut utiliser des valeurs littérales pour inclure et renommer des attributs individuels à partir de la charge utile du message.

Dans cet exemple, le message suivant est publié dans la rubrique A/B MQTT. Il s'agit du même message que celui utilisé dans [the section called “Exemple 1 de fonction de transformation” \(p. 674\)](#).

```

{
    "attributes": {
        "data1": 1,
        "data2": 2
    },
    "values": [
        {
            "a": 3
        },
        {
            "b": 4
        },
        {
            "c": 5
        }
    ]
}

```

}

Cette instruction SQL pour une action sur une règle de rubrique utilise la transform() fonction avec une String valeur de `enrichArray`. La transform() fonction Object in possède un seul attribut nommé `key` avec la valeur de `attributes.data1` dans la charge utile du message. Il s'agit d'un tableau qui contient les trois mêmes objets que ceux utilisés dans l'exemple précédent.

```
select value transform("enrichArray", {"key": attributes.data1}, values) from 'A/B'
```

À la réception de la charge utile du message, cette instruction SQL génère la réponse suivante. Remarquez comment la `data1` propriété est nommée `key` dans la réponse.

```
[  
  {  
    "a": 3,  
    "key": 1  
  },  
  {  
    "b": 4,  
    "key": 1  
  },  
  {  
    "c": 5,  
    "key": 1  
  }]
```

Exemple 3 de fonction de transformation

Cet exemple montre comment la transform() fonction peut être utilisée dans des clauses SELECT imbriquées pour sélectionner plusieurs attributs et créer de nouveaux objets à traiter ultérieurement.

Dans cet exemple, le message suivant est publié dans la rubrique A/B MQTT.

```
{  
  "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 fonction de transformation Object pour cette opération est l'objet renvoyé par l'instruction SELECT, qui contient les deux éléments `a` et `b` de l'objet `data2` du message. Le `Array` paramètre comprend les deux objets du tableau `c` du message d'origine.

```
select value transform('enrichArray', (select a, b from data2), (select value c from data2)) from 'A/B'
```

Avec le message précédent, l'instruction SQL aboutit 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 renvoyé dans cette réponse peut être utilisé avec des actions de règles de rubrique compatiblesbatchMode.

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.
Tableau	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.

Type d'argument	Résultat
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 deuxième s'il ne s'agit pas d'un inférieure la plus prochaine entière.
Int/Decimal/String	Int/Decimal	Le premier argument décrite par le deuxième s'il ne s'agit pas d'un inférieure la plus prochaine entière convertie, le résultat est un Decimal.
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"]`

Littéraux

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 {}). Par exemple :

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). Par exemple :

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 branches, comme une instruction switch.

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 mise en correspondance pour en vérifier l'égalité par rapport à la **t[i]** valeur de chaque WHEN clause. Si une correspondance est trouvée, l'**r[i]** expression correspondante devient le résultat de l'CASEinstruction. Les WHEN clauses sont évaluées dans l'ordre suivant lequel, s'il existe plusieurs clauses correspondantes, le résultat de la première clause correspondante devient le résultat de l'CASEinstruction. S'il n'y a aucune correspondance, **r[e]** c'est le résultat de la ELSE clause. S'il n'y a pas de correspondance ni de ELSE clause, le résultat est Undefined.

CASEles déclarations nécessitent au moins une WHEN clause. Une ELSE clause est facultative.

Par exemple :

Charge utile entrante publiée sur le sujet 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 résultante serait :

```
{  
    "instructions": "caution"  
}
```

Note

Si tel **v**est le casUndefined, le résultat de l'exposé de cas estUndefined.

Extensions JSON

Vous pouvez utiliser les extensions suivantes de la syntaxe ANSI SQL pour faciliter le travail avec des objets JSON imbriqués.

«. » Opérateur

Cet opérateur accède aux membres dans les objets et fonctions JSON intégrés de manière identique à ANSI SQL et JavaScript Par exemple :

```
SELECT foo.bar AS bar.baz FROM 'topic/subtopic'
```

sélectionne la valeur de la `bar` propriété de l'`foo`objet à partir de la charge utile du message suivant envoyé à la `topic/subtopic` rubrique.

```
{  
    "foo": {  
        "bar": "RED",  
        "bar1": "GREEN",  
        "bar2": "BLUE"  
    }  
}
```

Si le nom d'une propriété JSON inclut un trait d'union ou des caractères numériques, la notation « point » ne fonctionnera pas. Vous devez plutôt utiliser la [fonction get \(p. 647\)](#) pour extraire la valeur de la propriété.

Dans cet exemple, le message suivant est envoyé à la `iot/rules` rubrique.

```
{  
    "mydata": {  
        "item2": {  
            "0": {  
                "my-key": "myValue"  
            }  
        }  
    }  
}
```

Normalement, la valeur de `my-key` doit être identifiée comme dans cette requête.

```
SELECT * from iot/rules WHERE mydata.item2.0.my-key= "myValue"
```

Toutefois, étant donné que le nom de la propriété my-key contient un trait d'union et item2 un caractère numérique, la [fonction get \(p. 647\)](#) doit être utilisée comme l'indique 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. Par exemple :

```
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 \(p. 531\)](#). 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

Dans la mesure où une expression d'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 uniquement référencer les informations présentes dans la charge utile, les [fonctions \(p. 632\)](#) et [les opérateurs \(p. 626\)](#) d'origine.

Pour plus d'informations concernant les expressions prises en charge, consultez la page [Référence SQL AWS IoT \(p. 618\)](#).

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 \(p. 533\)](#)
- [Alarmes CloudWatch \(p. 541\)](#)
- [Journaux CloudWatch \(p. 542\)](#)
- [Métriques CloudWatch \(p. 544\)](#)
- [DynamoDB \(p. 546\)](#)
- [DynamoDBv2 \(p. 548\)](#)
- [Elasticsearch \(p. 549\)](#)
- [HTTP \(p. 551\)](#)
- [IoT Analytics \(p. 578\)](#)
- [AWS IoT Events \(p. 580\)](#)
- [AWS IoT SiteWise \(p. 582\)](#)
- [Kinesis Data Streams \(p. 587\)](#)
- [Kinesis Data Firehose \(p. 586\)](#)
- [Lambda \(p. 589\)](#)
- [Emplacement \(p. 591\)](#)
- [OpenSearch \(p. 594\)](#)
- [Republish \(p. 595\)](#)
- [S3 \(p. 597\)](#)
- [SNS \(p. 600\)](#)
- [SQS \(p. 601\)](#)
- [Step Functions \(p. 603\)](#)
- [Timestream \(p. 604\)](#)

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 aplatis 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

Pour traiter la charge utile de votre message comme des données binaires brutes (plutôt que comme un objet JSON), vous pouvez utiliser l'opérateur* pour y faire référence dans une clause SELECT.

Dans cette rubrique :

- [Exemples de charge utile binaire \(p. 684\)](#)
- [Décoder les charges utiles des messages Protobuf \(p. 685\)](#)

Exemples de charge utile binaire

Lorsque vous utilisez* pour désigner la charge utile du message sous forme de données binaires brutes, vous pouvez ajouter des données à la règle. Si vous avez une charge utile vide ou JSON, des données peuvent être ajoutées à la charge utile résultante à l'aide de la règle. Vous trouverez ci-dessous des exemples de SELECT clauses prises en charge.

- Vous pouvez utiliser les SELECT clauses suivantes avec uniquement un* pour les charges utiles binaires.
 - ```
SELECT * FROM 'topic/subtopic'
```
  - ```
SELECT * FROM 'topic/subtopic' WHERE timestamp() % 12 = 0
```
- Vous pouvez également ajouter des données et utiliser les SELECT clauses suivantes.
 - ```
SELECT *, principal() as principal, timestamp() as time FROM 'topic/subtopic'
```
  - ```
SELECT encode(*, 'base64') AS data, timestamp() AS ts FROM 'topic/subtopic'
```
- Vous pouvez également utiliser ces SELECT clauses avec des charges utiles binaires.
 - Ce qui suit fait référence device_type à la clause WHERE.

```
SELECT * FROM 'topic/subtopic' WHERE device_type = 'thermostat'
```

- Les éléments suivants sont également pris en charge.

```
{  
    "sql": "SELECT * FROM 'topic/subtopic'",  
    "actions": [  
        {  
            "republish": {  
                "topic": "device/${device_id}"  
            }  
        }  
    ]  
}
```

}

Les actions de règles suivantes ne prennent pas en charge les charges utiles binaires. Vous devez donc les décoder.

- Certaines actions de règles ne prennent pas en charge l'entrée de charge utile binaire, comme une [action Lambda](#). Vous devez donc décoder des charges utiles binaires. L'action de la règle Lambda peut recevoir des données binaires, si elles sont codées en base64 et qu'elles figurent dans une charge utile JSON. Pour ce faire, vous pouvez modifier la règle comme suit.

```
SELECT encode(*, 'base64') AS data FROM 'my_topic'
```

- L'instruction SQL ne prend pas en charge les chaînes en entrée. Pour convertir une entrée de chaîne en JSON, vous pouvez exécuter la commande suivante.

```
SELECT decode(encode(*, 'base64'), 'base64') AS payload FROM 'topic'
```

Décoder les charges utiles des messages Protobuf

[Protocol Buffers \(protobuf\)](#) est un format de données open source utilisé pour sérialiser des données structurées sous une forme binaire compacte. Il est utilisé pour transmettre des données sur des réseaux ou les stocker dans des fichiers. Protobuf vous permet d'envoyer des données sous forme de paquets de petite taille et à un rythme plus rapide que les autres formats de messagerie. AWS IoT Core Les règles prennent en charge le protocole protobuf en fournissant la fonction SQL [decode \(value, decodingScheme\) \(p. 644\)](#), qui vous permet de décoder les charges utiles des messages codés en protobuf au format JSON et de les acheminer vers des services en aval. Cette section détaille le step-by-step processus de configuration du décodage protobuf dans Rules. AWS IoT Core

Dans cette section :

- [Prérequis \(p. 685\)](#)
- [Création de fichiers descripteurs \(p. 685\)](#)
- [Charger les fichiers descripteurs dans le compartiment S3 \(p. 686\)](#)
- [Configurer le décodage protobuf dans les règles \(p. 687\)](#)
- [Limites \(p. 688\)](#)
- [Bonnes pratiques \(p. 688\)](#)

Prérequis

- Compréhension de base des [Protocol Buffers \(protobuf\)](#)
- Les [.protofichiers](#) qui définissent les types de messages et les dépendances associées
- Installation du [compilateur Protobuf \(protoc\)](#) sur votre système

Création de fichiers descripteurs

Si vous avez déjà vos fichiers descripteurs, vous pouvez ignorer cette étape. Un fichier descripteur (.desc) est une version compilée d'un .proto fichier, qui est un fichier texte qui définit les structures de données et les types de messages à utiliser dans une sérialisation protobuf. Pour générer un fichier descripteur, vous devez définir un .proto fichier et utiliser le compilateur de [protocoles pour](#) le compiler.

1. Créez des .proto fichiers qui définissent les types de messages. Par exemple, .proto le fichier peut ressembler à ce qui suit :

```
syntax = "proto3";

message Person {
    optional string name = 1;
    optional int32 id = 2;
    optional string email = 3;
}
```

Dans ce .proto fichier d'exemple, vous utilisez la syntaxe proto3 et vous définissez le type de Person message. La définition du Person message spécifie trois champs (nom, identifiant et e-mail). Pour plus d'informations sur les formats de .proto fichiers de message, consultez le [Guide linguistique \(proto3\)](#).

2. Utilisez le compilateur [de protocoles](#) pour compiler les .proto fichiers et générer un fichier descripteur. Voici un exemple de commande pour créer un fichier descripteur (.desc) :

```
protoc --descriptor_set_out=<FILENAME>.desc \
--proto_path=<PATH_TO_IMPORTS_DIRECTORY> \
--include_imports \
<PROTO_FILENAME>.proto
```

Cet exemple de commande génère un fichier descripteur<FILENAME>.desc, que AWS IoT Core Rules peut utiliser pour décoder les charges utiles protobuf conformes à la structure de données définie dans. <PROTO_FILENAME>.proto

- **--descriptor_set_out**

Spécifie le nom du fichier descripteur (<FILENAME>.desc) qui doit être généré.

- **--proto_path**

Spécifie les emplacements de tous .proto les fichiers importés auxquels le fichier en cours de compilation fait référence. Vous pouvez spécifier l'indicateur plusieurs fois si vous avez plusieurs .proto fichiers importés avec des emplacements différents.

- **--include_imports**

Spécifie que tous .proto les fichiers importés doivent également être compilés et inclus dans le fichier <FILENAME>.desc descripteur.

- **<PROTO_FILENAME>.proto**

Spécifie le nom du .proto fichier que vous souhaitez compiler.

Pour plus d'informations sur la référence de protocole, consultez Référence de [l'API](#).

Charger les fichiers descripteurs dans le compartiment S3

Après avoir créé vos fichiers descripteurs<FILENAME>.desc, chargez-les dans un compartiment Amazon S3 <FILENAME>.desc à l'aide de l'AWSAPI, du AWS SDK ou du AWS Management Console

Considérations importantes

- Assurez-vous de charger les fichiers descripteurs dans un compartiment Amazon S3 de votre ordinateur Compte AWS dans Région AWS lequel vous avez l'intention de configurer vos règles.
- Assurez-vous que vous autorisez AWS IoT Core l'accès pour lire le FileDescriptorSet depuis S3. Si le chiffrement côté serveur (SSE-S3) est désactivé pour votre compartiment S3 ou si votre compartiment S3 est chiffré à l'aide des clés gérées par Amazon S3 (SSE-S3), aucune configuration

de stratégie supplémentaire n'est requise. Cela peut être accompli à l'aide de l'exemple de politique de compartiment :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "s3:Get*",
      "Resource": "arn:aws:s3:::<BUCKET NAME>/<FILENAME>.desc"
    }
  ]
}
```

- Si votre compartiment S3 est chiffré au moyen d'une AWS Key Management Service clé (SSE-KMS), assurez-vous d'accorder AWS IoT Core l'autorisation d'utiliser la clé pour accéder à votre compartiment S3. Pour ce faire, ajoutez cette instruction à votre politique clé :

```
{
  "Sid": "Statement1",
  "Effect": "Allow",
  "Principal": {
    "Service": "iot.amazonaws.com"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

Configurer le décodage protobuf dans les règles

Après avoir chargé les fichiers descripteurs dans votre compartiment Amazon S3, configurez une règle capable de décoder le format de charge utile de votre message protobuf à l'aide de la fonction SQL [decode \(value, decodingScheme \(p. 644\)\)](#). Vous trouverez une signature de fonction détaillée et un exemple dans la fonction SQL [decode \(value, decodingScheme\) \(p. 644\)](#) de la référence SQL AWS IoT.

Voici un exemple d'expression SQL utilisant la fonction [decode \(value, decodingScheme \(p. 644\)\)](#) :

```
SELECT VALUE decode(*, 'proto', '<BUCKET NAME>', '<FILENAME>.desc', '<PROTO_FILENAME>',
  '<PROTO_MESSAGE_TYPE>') FROM '<MY_TOPIC>'
```

Dans cet exemple d'expression :

- Vous utilisez la fonction SQL [decode \(value, decodingScheme\) \(p. 644\)](#) pour décoder la charge utile du message binaire référencée par. * Il peut s'agir d'une charge utile binaire codée en protobuf ou d'une chaîne JSON qui représente une charge utile protobuf codée en base64.
- La charge utile du message fournie est codée à l'aide du type de Person message défini dansPROTO_FILENAME.proto.
- Le compartiment Amazon S3 nommé BUCKET_NAME contient le fichier FILENAME.desc généré à partir dePROTO_FILENAME.proto.

Une fois la configuration terminée, publiez un message AWS IoT Core sur le sujet auquel la règle est abonnée.

Limites

AWS IoT Core Les règles prennent en charge protobuf avec les limitations suivantes :

- Le décodage des charges utiles des messages protobuf dans les [modèles de substitution](#) n'est pas pris en charge.
- Lors du décodage des charges utiles des messages protobuf, vous pouvez utiliser la [fonction SQL de décodage \(p. 644\)](#) au sein d'une même expression SQL jusqu'à deux fois.
- La taille maximale de la charge utile entrante est de 128 KiB (1 Ko = 1024 octets), la taille maximale de la charge utile sortante est de 128 KiB et la taille maximale d'un FileDescriptorSet objet stocké dans un compartiment Amazon S3 est de 32 KiB.
- Les compartiments Amazon S3 chiffrés avec le chiffrement SSE-C ne sont pas pris en charge.

Bonnes pratiques

Voici quelques bonnes pratiques et conseils de dépannage.

- Sauvegardez vos fichiers proto dans le compartiment Amazon S3.

Il est recommandé de sauvegarder vos fichiers proto en cas de problème. Par exemple, si vous modifiez de manière incorrecte les fichiers proto sans sauvegarde lors de l'exécution du protocole, cela peut entraîner des problèmes dans votre pile de production. Il existe plusieurs méthodes pour sauvegarder vos fichiers dans un compartiment Amazon S3. Par exemple, vous pouvez [utiliser le contrôle de version dans les compartiments S3](#). Pour plus d'informations sur la façon de sauvegarder des fichiers dans des compartiments Amazon S3, consultez le manuel [Amazon S3 Developer Guide](#).

- Configurez la AWS IoT journalisation pour afficher les entrées du journal.

Il est recommandé de configurer la AWS IoT journalisation afin de pouvoir consulter AWS IoT les journaux de votre compte CloudWatch. Lorsque la requête SQL d'une règle appelle une fonction externe, AWS IoT Core Rules génère une entrée de journal avec un eventType deFunctionExecution, qui contient le champ Motif qui vous aidera à résoudre les défaillances. Les erreurs possibles incluent un objet Amazon S3 introuvable ou un descripteur de fichier protobuf non valide. Pour plus d'informations sur la façon de configurer la AWS IoT journalisation et de consulter les entrées du journal, voir [Configurer la AWS IoT journalisation](#) et [les entrées du journal du moteur de règles](#).

- Effectuez la mise à jour à l'FileDescriptorSet aide d'une nouvelle clé d'objet et mettez à jour la clé d'objet dans votre règle.

Vous pouvez effectuer la mise à jour FileDescriptorSet en chargeant un fichier descripteur mis à jour dans votre compartiment Amazon S3. Vos mises à jour FileDescriptorSet peuvent prendre jusqu'à 15 minutes pour être prises en compte. Pour éviter ce retard, il est recommandé de télécharger votre mise à jour à FileDescriptorSet l'aide d'une nouvelle clé d'objet et de mettre à jour la clé d'objet dans votre règle.

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 \(p. 527\)](#). 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 :

- 2016-03-23— La version SQL construite le 23/03/2016 (recommandée).
- 2015-10-08— Version SQL d'origine construite le 08/10/2015.
- beta— La version bêta la plus récente de SQL. Cette version peut apporter des modifications majeures à 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, veuillez consulter [Requêtes d'objets imbriqués \(p. 683\)](#).
- 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 \(p. 646\)](#).

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.

AWS IoTLes objets ne possèdent aucune ombre nommée tant qu'ils ne sont pas créés de manière explicite ; toutefois, une ombre classique anonyme est créée pour un objet lors de sa création. 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 ombres à l'aide de MQTT et des rubriques MQTT réservées \(p. 128\)](#), [du protocole HTTP à l'aide de l'API REST Device Shadow \(p. 716\)](#) et [du AWS CLI for AWS IoT](#). Les ombres étant stockées AWS dans le cloud, elles peuvent collecter et signaler les données d'état de l'appareil à partir d'applications et d'autres services cloud, que l'appareil soit connecté ou non.

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.

Si vos appareils sont fréquemment hors ligne et que vous souhaitez les configurer pour recevoir des messages delta après leur reconnexion, vous pouvez utiliser la fonctionnalité de session persistante. Pour plus d'informations sur la période d'expiration des sessions [persistantes, voir Période d'expiration des sessions persistantes.](#)

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 \(p. 128\)](#).

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. Pour en savoir plus sur les stratégies à utiliser [avec les stratégies pour AWS IoT](#).

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.

L'indexation de la flotte prend en charge les ombres anonymes et les ombres nommées différemment. Pour de plus amples informations, veuillez consulter [Gestion de l'indexation de la flotte \(p. 930\)](#).

Accès aux shadows

Chaque shadow a une [rubrique MQTT \(p. 128\)](#) réservée et une [URL HTTP \(p. 716\)](#) qui prend en charge les actions get, update et delete sur le shadow.

Les shadows utilisent des [documents de shadow JSON \(p. 729\)](#) 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 \(p. 694\)](#) et [Utilisation des shadows dans les applications et les services \(p. 697\)](#).

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. Par exemple :

- Les appareils 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",
                "qos": 1
            }
        }
    ]
}
```

```
        "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 \(p. 524\)](#).

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 \(p. 698\)](#).

Important

Étant donné que MQTT utilise un modèle de communication publication/abonnement, vous devez vous abonner aux rubriques de réponse avant de publier une rubrique de demande. Si ce n'est pas le cas, il se peut que vous ne receviez pas de réponse à la demande de publication.

Si vous utilisez un [Kit SDK des appareils AWS IoT \(p. 1494\)](#) 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 dans laquelle le terme **ShadowTopicPrefix** peut faire référence à une ombre nommée ou non, 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.

ShadowTopicPrefix valeur	Type de shadow
\$aws/things/ <i>thingName</i> /shadow	Shadow non nommé (classique)
\$aws/things/ <i>thingName</i> /shadow/ <i>name</i> / <i>shadowName</i>	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 \(p. 128\)](#) 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
<i>ShadowTopicPrefix/</i> <i>delete/accepted</i>	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.
<i>ShadowTopicPrefix/</i> <i>delete/rejected</i>	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.
<i>ShadowTopicPrefix/get/</i> <i>accepted</i>	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.
<i>ShadowTopicPrefix/get/</i> <i>rejected</i>	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.
<i>ShadowTopicPrefix/</i> <i>update/accepted</i>	La demande update a été acceptée par AWS IoT, et le corps du message contient le document shadow actuel.	Confirmer que les données mises à jour dans le corps du message correspondent à l'état de l'appareil.
<i>ShadowTopicPrefix/</i> <i>update/rejected</i>	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.
<i>ShadowTopicPrefix/</i> <i>update/delta</i>	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.

Sujet	Signification	Action qu'un appareil doit effectuer lors de la réception de cette rubrique
<i>ShadowTopicPrefix/</i> 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 publant 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 publant un message /update avec l'état actuel de l'appareil.

Par exemple, supposons que vous ayez un objet My_IoT_Thing qui ne comporte des variantes. Si vous publiez maintenant une /get requête sur le sujet réservé \$aws/things/My_IoT_Thing/shadow/get, cela renvoie une erreur sur le \$aws/things/My_IoT_Thing/shadow/get/rejected sujet car le sujet ne comporte aucune ombre. Pour résoudre cette erreur, commencez par publier un /update message en utilisant la \$aws/things/My_IoT_Thing/shadow/update rubrique correspondant à l'état actuel de l'appareil, par exemple la charge utile suivante.

```
{
  "state": {
    "reported": {
      "welcome": "aws-iot",
      "color": "yellow"
    }
  }
}
```

Une ombre classique est désormais créée pour l'objet et le message est publié dans le \$aws/things/My_IoT_Thing/shadow/update/accepted sujet. Si vous publiez sur la rubrique \$aws/things/My_IoT_Thing/shadow/get, elle renvoie une réponse à la \$aws/things/My_IoT_Thing/shadow/get/accepted rubrique avec l'état de l'appareil.

Pour les ombres nommées, vous devez d'abord créer l'ombre nommée ou publier une mise à jour avec le nom de l'ombre avant d'utiliser la requête get. Par exemple, pour créer une ombre nommée namedShadow1, publiez d'abord les informations sur l'état de l'appareil dans la rubrique \$aws/things/My_IoT_Thing/shadow/namedShadow1/update. Pour récupérer les informations d'état, utilisez la /get requête pour l'ombre nommée \$aws/things/My_IoT_Thing/shadow/namedShadow1/get.

Traitement des messages lorsque l'appareil est connecté à AWS IoT

Lorsqu'un appareil est connecté AWS IoT, il peut recevoir des messages /update/delta et doit maintenir l'état de l'appareil en fonction de l'évolution de ses ombres 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, 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.	<i>ShadowTopicPrefix</i> /update	Un document shadow avec la propriété <code>reported</code> .
L'appareil peut ne pas être synchronisé avec le shadow.	<i>ShadowTopicPrefix</i> /get	(empty)
Une action sur l'appareil indique qu'une ombre ne sera plus supportée par l'appareil, par exemple lors de son retrait ou de son remplacement.	<i>ShadowTopicPrefix</i> /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 \(p. 694\)](#), 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 \(p. 694\)](#).

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/things/thingName/shadow?name=shadowName
```

et pour un shadow non nommé :

```
https://endpoint/things/thingName/shadow
```

où :

point de terminaison

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

- Créez une règle qui republie le message MQTT LWT sur la rubrique réservée. L'exemple suivant est une règle qui écoute les messages relatifs au my/things/myLightBulb/update sujet et les republie sur. \$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"
        }
      }
    ]
  }
}
```

- 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"
    }
  }
}
```

- 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"
    }
  }
}
```

- 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 en détail, vous aurez besoin d'un Compte AWS et d'un système sur lequel vous pouvez exécuter le 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. 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 dans [Configurez 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 nommé et son ombre nommée `SimShadow1`. `mySimulatedThing`

Création d'un objet et de sa politique en matière d'IoT

Pour créer un objet, dans la AWS IoTconsole :

1. Choisissez Gérer, puis Objets.
2. Cliquez sur le bouton Créer si des éléments sont répertoriés, sinon cliquez sur Enregistrer un seul objet> pour créer un seul AWS IoT objet.
3. Entrez le nom `mySimulatedThing`, laissez les autres paramètres par défaut, puis cliquez sur Suivant.

4. Utilisez la création de certificats en un clic pour générer les certificats qui authentifieront la connexion de l'appareil à AWS IoT. Cliquez sur Activer pour activer le certificat.
5. Vous pouvez joindre la politique My_IoT_Policy qui autoriserait l'appareil à publier et à s'abonner aux sujets réservés du MQTT. Pour obtenir des instructions plus détaillées sur la création d'un AWS IoT objet et sur la façon de créer cette politique, consultez [Création d'un objet \(p. 42\)](#).

Crée une ombre nommée pour l'objet objet

Vous pouvez créer une ombre nommée pour un objet en publant une demande de mise à jour pour la rubrique \$aws/things/mySimulatedThing/shadow/name/simShadow1/update, comme décrit ci-dessous.

Vous pouvez également créer une ombre nommée :

1. Dans la AWS IoTconsole, choisissez votre objet dans la liste des objets affichés, puis choisissez Ombres.
2. Choisissez Ajouter une ombre, entrez le nom simShadow1, puis choisissez Créer pour ajouter l'ombre nommée.

Abonnez-vous et publiez sur des sujets MQTT réservés

Dans la console, abonnez-vous aux sujets parallèles réservés au MQTT. 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. Dans le client MQTT, choisissez S'abonner à une rubrique.
2. Entrez le getupdate, et les delete sujets auxquels vous souhaitez vous abonner. Copiez un sujet à la fois dans la liste suivante, collez-le dans le champ Filtre des sujets, puis cliquez sur S'abonner. Vous devriez voir les rubriques apparaître sous Abonnements.
 - \$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 le client 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 ne prend en charge qu'une seule ombre, l'ombre qui soutient un objet nommé SimShadow1. mySimulatedThing

Pour obtenir l'état actuel du shadow à partir du client MQTT

1. Dans le client MQTT, choisissez Publier dans une rubrique.

2. Sous Publier, entrez le sujet suivant et supprimez tout contenu de la fenêtre du corps du message située en dessous de l'endroit où vous avez saisi le sujet à consulter. Vous pouvez ensuite choisir Publier dans le sujet pour publier la demande. \$aws/things/mySimulatedThing/shadow/name/simShadow1/get.

Si vous n'avez pas créé l'ombre nommée simShadow1, vous recevez un message dans la \$aws/things/mySimulatedThing/shadow/name/simShadow1/get/rejected rubrique et l'code est 404, comme dans cet exemple indiquant que l'ombre n'a pas été créée. Nous allons donc la créer ensuite.

```
{  
  "code": 404,  
  "message": "No shadow exists with name: 'simShadow1'"  
}
```

Pour créer un shadow avec l'état actuel de l'appareil

1. Dans le client MQTT, choisissez Publier dans une rubrique et saisissez 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. Choisissez Publier pour publier la demande.

```
{  
  "state": {  
    "reported": {  
      "ID": "SmartLamp21",  
      "ColorRGB": [  
        128,  
        128,  
        128  
      ]  
    }  
  },  
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"  
}
```

Si vous recevez un message dans le sujet :

- \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/accepted: Cela signifie que l'ombre a été créée et que le corps du message contient le document miroir actuel.
- \$aws/things/mySimulatedThing/shadow/name/simShadow1/update/rejected: vérifiez l'erreur dans le corps du message.
- \$aws/things/mySimulatedThing/shadow/name/simShadow1/get/accepted: L'ombre existe déjà et le corps du message possède l'état d'ombre actuel, 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  
      ]  
    }  
  },  
  "clientToken": "426bfd96-e720-46d3-95cd-014e3ef12bb6"  
}
```

```
        ],
    },
    "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 plateformes Windows, vous pouvez utiliser à la con place 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": [  
                255,  
                255,  
                0  
            ]  
        }  
    }  
}
```

```
{  
    "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/ /shadow/name/simshadow1/update/delta mySimulatedThing 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 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 du corps du message, dans le champ de rubrique situé au-dessus de la fenêtre du corps du message, entrez le sujet de l'ombre suivi de l'/updateaction : \$aws/things/*mySimulatedThing*/shadow/name/*simShadow1*/update et dans le corps du message, entrez ce document parallèle mis à jour, qui décrit l'état actuel de l'appareil. Cliquez sur Publier pour publier l'état mis à jour de l'appareil.

```
{  
  "state": {  
    "reported": {  
      "ColorRGB": [255,255,0]  
    }  
  },  
  "clientToken": "a4dc2227-9213-4c6a-a6a5-053304f60258"  
}
```

Si le message a bien été reçu par AWS IoT, vous devriez voir apparaître une nouvelle réponse dans le journal des messages \$aws/things/ *mySimulatedThing* /shadow/name/*simshadow1*/update/accepted du client MQTT avec l'état actuel de l'ombre, 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 plateformes Windows, vous pouvez utiliser à la place 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

Testez l'interaction entre AWS 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 l'ombre de l'appareil et son contenu.

Vous ne pouvez pas restaurer un document miroir d'appareil supprimé, mais vous pouvez créer un nouveau document miroir d'appareil avec le nom du document miroir d'appareil supprimé. Si vous créez un document miroir d'appareil portant le même nom que celui qui a été supprimé au cours des dernières 48 heures, le numéro de version du nouveau document miroir de l'appareil suivra celui du document supprimé. Si un document miroir de l'appareil a été supprimé pendant plus de 48 heures, le numéro de version d'un nouveau document miroir de l'appareil portant le même nom sera 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 \(p. 721\)](#) et [API REST Device Shadow \(p. 716\)](#).

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 \(p. 718\)](#) ou en publiant dans la rubrique [/update \(p. 723\)](#). 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 \(p. 730\)](#) et le message `/update/delta` contient un document shadow [/documents d'état de la réponse delta \(p. 731\)](#).

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 \(p. 733\)](#) 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 \(p. 718\)](#) avec un document d'état [Document d'état de demande \(p. 729\)](#) 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 \(p. 730\)](#) 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 \(p. 731\)](#) 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 \(p. 733\)](#) 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 publant 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 \(p. 730\)](#).
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 \(p. 733\)](#) 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 \(p. 718\)](#) avec un document d'état [Document d'état de demande \(p. 729\)](#) 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 \(p. 730\)](#) 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 [Document d'état de la réponse delta \(p. 731\)](#) 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 \(p. 733\)](#) 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.

Par exemple :

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": 10  
}
```

```
        "colors": [
            "BLUE"
        ]
    },
    "version": 11
}
```

Récupération d'un document Shadow

Vous pouvez récupérer un document shadow à l'aide de l'API [GetThingShadow \(p. 717\)](#) ou en vous abonnant et en publant dans la rubrique [/get \(p. 722\)](#). 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 \(p. 730\)](#) 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 \(p. 733\)](#) 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 \(p. 717\)](#) 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 \(p. 730\)](#) 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 \(p. 733\)](#) 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 \(p. 719\)](#) ou publiez dans la rubrique [/delete \(p. 727\)](#).

Note

La suppression d'une ombre ne réinitialise pas immédiatement son numéro de version à zéro. Il sera remis à zéro après 48 heures.

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 \(p. 714\)](#) 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 \(p. 730\)](#) 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 \(p. 733\)](#) 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 \(p. 718\)](#) avec un [Document d'état de demande \(p. 729\)](#) 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 \(p. 730\)](#) 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 \(p. 733\)](#) comme corps de message de réponse.

Suppression d'un shadow

Vous trouverez ci-après quelques considérations relatives à la suppression de l'ombre d'un appareil.

- 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 immédiatement son numéro de version à zéro. Il sera remis à zéro après 48 heures.

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 \(p. 730\)](#) 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 \(p. 733\)](#) 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 \(p. 719\)](#) 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 \(p. 730\)](#) et un document shadow [/document d'état de la réponse accepté \(p. 730\)](#) 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 \(p. 733\)](#) 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://account-specific-prefix-ats.iot.region.amazonaws.com/things/thingName/shadow
```

Le point de terminaison est spécifique à votre Compte AWS. Pour trouver votre point de terminaison, vous pouvez :

- Utilisez la commande [describe-endpoint](#) depuis le AWS CLI
- Utilisez les paramètres de AWS IoT la console. Dans Paramètres, le point de terminaison est répertorié sous Point de terminaison personnalisé
- Utilisez la page des détails de la AWS IoT console. Dans la console :
 1. Ouvrez Gérer et sous Gérer, choisissez Objets.
 2. Dans la liste des éléments, choisissez l'élément pour lequel vous souhaitez obtenir l'URI du point de terminaison.
 3. Choisissez l'onglet Device Shadows et choisissez votre ombre. Vous pouvez consulter l'URI du point de terminaison dans la section URL de Device Shadow de la page de détails de Device Shadow.

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 des appareils \(p. 89\)](#).

Note

Pour utiliser les API, vous devez l'utiliser `iotdevicegateway` comme nom de service pour l'authentification. Pour plus d'informations, consultez la section [IoT DataPlane](#).

Actions d'API

- [GetThingShadow \(p. 717\)](#)
- [UpdateThingShadow \(p. 718\)](#)
- [DeleteThingShadow \(p. 719\)](#)
- [ListNamedShadowsForThing \(p. 720\)](#)

Vous pouvez également utiliser l'API pour créer une ombre nommée en la fournissant dans le `name=shadowName` cadre du paramètre de requête de l'API.

GetThingShadow

Obtient le shadow de l'objet spécifié.

Le document d'état de réponse comprend le delta entre les états `desired` et `reported`.

Requête

La demande comprend les en-têtes HTTP standard, plus l'URI suivante :

`HTTP GET https://endpoint/things/thingName/shadow?name=shadowName`
Request body: (none)

Le paramètre de requête `name` n'est pas requis pour les shadows non nommés (classiques).

Réponse

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 \(p. 730\)](#).

Autorisation

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 : la version 4 de signature avec des 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.

Requête

La demande comprend les en-têtes HTTP standard, plus l'URI et le corps suivants :

```
HTTP POST https://endpoint/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 \(p. 729\)](#).

Réponse

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 \(p. 730\)](#).

Autorisation

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 : la

version 4 de signature avec des 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.

Requête

La demande comprend les en-têtes HTTP standard, plus l'URI suivante :

```
HTTP DELETE https://endpoint/things/thingName/shadow?name=shadowName  
Request body: (none)
```

Le paramètre de requête name n'est pas requis pour les shadows non nommés (classiques).

Réponse

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.

Autorisation

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 : la version 4 de signature avec des 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.

Requête

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.

Réponse

En cas de réussite, la réponse inclut les en-têtes HTTP standard ainsi que le code de réponse suivant et un [Document de réponse de liste de noms de shadows \(p. 733\)](#).

Note

Le shadow non nommé (classique) n'apparaît pas dans cette liste. La réponse est une liste vide si vous n'avez qu'une ombre classique ou si celle thingName que vous avez spécifiée n'existe pas.

```
HTTP 200  
Response body: Shadow name list document
```

Autorisation

Pour répertorier l'ombre d'un appareil, vous devez définir une politique qui autorise l'appelant à effectuer l'iot:ListNamedShadowsForThingaction. Le service Device Shadow accepte deux formes d'authentification : la version 4 de signature avec des 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 \(p. 709\)](#).

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.

ShadowTopicPrefix valeur	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 \(p. 722\)](#)
- [/get/accepted \(p. 722\)](#)
- [/get/rejected \(p. 723\)](#)
- [/update \(p. 723\)](#)
- [/update/delta \(p. 724\)](#)
- [/update/accepted \(p. 725\)](#)
- [/update/documents \(p. 726\)](#)
- [/update/rejected \(p. 726\)](#)
- [/delete \(p. 727\)](#)
- [/delete/accepted \(p. 728\)](#)
- [/delete/rejected \(p. 728\)](#)

/get

Publier un message vide dans cette rubrique pour obtenir le shadow d'appareil :

ShadowTopicPrefix/get

AWS IoT répond en publant dans [/get/accepted \(p. 722\)](#) ou [/get/rejected \(p. 723\)](#).

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, veuillez consulter [Documents d'état de la réponse \(p. 730\)](#).

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, veuillez consulter [Document de réponse d'erreur \(p. 733\)](#).

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 \(p. 729\)](#).

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 publant dans [/update/accepted \(p. 725\)](#) ou [/update/rejected \(p. 726\)](#).

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 \(p. 731\)](#).

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 \(p. 730\)](#).

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"  
      ]  
    }  
  ]  
}
```

```
    "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 \(p. 731\)](#).

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 \(p. 733\)](#).

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"  
            ]  
        }  
    ]  
}
```

/delete

Pour supprimer un shadow d'appareil, publiez un message vide dans la rubrique delete :

ShadowTopicPrefix/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 \(p. 728\)](#) ou [/delete/rejected \(p. 728\)](#).

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": "TopicArn"  
        }  
    ]  
}
```

```
    "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 \(p. 733\)](#).

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.

Table des matières

- [Exemples de documents shadow \(p. 729\)](#)
- [Propriétés du document \(p. 734\)](#)
- [État Delta \(p. 734\)](#)
- [Documents shadow de gestion des versions \(p. 736\)](#)
- [Jetons clients dans les documents shadow \(p. 736\)](#)
- [Propriétés de document shadow vides \(p. 736\)](#)
- [Valeurs de tableau dans les documents shadow \(p. 737\)](#)

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 \(p. 716\)](#) ou des [messages de publication/abonnement MQTT \(p. 721\)](#).

Exemples

- [Document d'état de demande \(p. 729\)](#)
- [Documents d'état de la réponse \(p. 730\)](#)
- [Document de réponse d'erreur \(p. 733\)](#)
- [Document de réponse de liste de noms de shadows \(p. 733\)](#)

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 et les valeurs d'état dont la mise à jour est demandée sur l'appareil.
- **reported**— Les propriétés d'état et les valeurs signalées par l'appareil.
- **clientToken**— S'il est utilisé, vous pouvez faire correspondre la demande et la réponse correspondante à l'aide du 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
}
```

```
    "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
    },
    "attribute2timestamp
    },
    ...
    "attributeNtimestamp
    }
}
},
"version": version-1
},
"current": {
    "state": {
        "desired": {
            "attribute1integer2,
            "attribute2attributeNboolean2
        },
        "reported": {
            "attribute1integer2,
            "attribute2attributeNboolean2
        }
    },
    "metadata": {
        "desired": {
            "attribute1timestamp
            },
            "attribute2timestamp
            },
            ...
            "attributeNtimestamp
            }
        },
        "reported": {
            "attribute1timestamp
            },
            "attribute2timestamp
            },
            ...
            "attributeNtimestamp
            }
        }
    },
    "version": version
},
"timestamp": timestamp,
"clientToken": "token"
}

```

Propriétés du document d'état de réponse

- **previous**— Après une mise à jour réussie, contient le contenu `state` de l'objet avant la mise à jour.
- **current**— Après une mise à jour réussie, contient le nom `state` de l'objet après la mise à jour.

- **state**
 - **reported**— Présent uniquement si un objet contient des données dans la `reported` section et ne contient que des champs figurant dans le document d'état de la demande.
 - **desired**— Présent uniquement si un appareil a signalé des données dans la `desired` section et contient uniquement les champs figurant dans le document d'état de la demande.
 - **delta**— Présent uniquement si les `desired` données diffèrent des `reported` données actuelles de l'ombre.
- **metadata**— Contient les horodatages de chaque attribut dans les `reported` sections `desired` et afin que vous puissiez déterminer à quel moment l'état a été mis à jour.
- **timestamp**— La date et l'heure de l'époque à laquelle la réponse a été générée AWS IoT.
- **clientToken**— Présent uniquement si un jeton client a été utilisé lors de la publication d'un code JSON valide dans la `/update` rubrique.
- **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**— Un code de réponse HTTP qui indique le type d'erreur.
- **message**— Un message texte qui fournit des informations supplémentaires.
- **timestamp**— La date et l'heure auxquelles la réponse a été générée AWS 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, veuillez consulter [Messages d'erreur de Device Shadow \(p. 738\)](#).

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**— Le tableau des noms des ombres.
- **nextToken**— La valeur du jeton à utiliser dans les demandes paginées pour accéder à la page suivante de la séquence. Cette propriété n'est pas présente quand il ne reste plus de noms de shadows à renvoyer.

- **timestamp**— La date et l'heure auxquelles la réponse a été générée AWS 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 IoT Limites 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, veuillez consulter [Exemples de documents shadow \(p. 729\)](#).

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**. Par exemple :

```
{  
  "state": {  
    "desired": {  
      "color": "RED",  
      "lastReported": "2018-01-01T12:00:00Z"  
    },  
    "reported": {  
      "color": "GREEN",  
      "lastReported": "2018-01-01T12:00:00Z"  
    }  
  }  
}
```

```

        "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 uniquement émis en réponse à une demande de publication sur l'un des \$aws sujets réservés. 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)	<ul style="list-style-type: none"> 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 <p>Note</p> <p>Un jeton client d'une longueur supérieure à 64 octets génère cette réponse.</p> <ul style="list-style-type: none"> Le code JSON contient trop de niveaux d'imbrication ; le maximum est 6 L'état contient un nœud non valide
401 (Accès non autorisé)	<ul style="list-style-type: none"> Non autorisé
403 (Accès interdit)	<ul style="list-style-type: none"> Accès interdit
404 (Introuvable)	<ul style="list-style-type: none"> Objet non trouvé Aucun shadow n'existe avec le nom : <i>shadowName</i>
409 (Conflit)	<ul style="list-style-type: none"> Conflit de versions
413 (Charge utile trop importante)	<ul style="list-style-type: none"> La charge utile dépasse la taille maximale autorisée
415 (Type de support non pris en charge)	<ul style="list-style-type: none"> Codage documenté non pris en charge ; l'encodage pris en charge est UTF-8
429 (Nombre de requêtes trop élevé)	<ul style="list-style-type: none"> Le service Device Shadow génère ce message d'erreur lorsqu'il y a plus de 10 demandes en cours sur une seule connexion. Une demande en cours est une demande en cours qui a été lancée mais qui n'est pas encore terminée.
500 (Erreur interne du serveur)	<ul style="list-style-type: none"> Échec du service interne

Tâches

Utilisez AWS IoT Jobs pour définir un ensemble d'opérations à distance qui peuvent être envoyées et exécutées sur un ou plusieurs appareils connectés AWS IoT. Par exemple, vous pouvez définir une tâche qui demande à un ensemble d'appareils de télécharger et d'installer des applications, d'exécuter des mises à jour du microprogramme, de redémarrer, de faire pivoter des certificats ou d'effectuer des opérations de dépannage à distance.

Accès aux AWS IoT offres d'emploi

Vous pouvez commencer à utiliser AWS IoT Jobs à l'aide de la console ou de l'AWS IoT Core API.

Utilisation de la console

Connectez-vous au AWS Management Console, puis accédez à la AWS IoT console. Dans le volet de navigation, choisissez Gérer, puis Tâches. Vous pouvez créer et gérer des tâches à partir de cette section. Si vous souhaitez créer et gérer des modèles de Job, dans le volet de navigation, sélectionnez Modèles de tâches. Pour plus d'informations, veuillez consulter [Créez et gérez des tâches à l'aide du AWS Management Console \(p. 749\)](#).

Utilisation de l'API ou de la CLI

Vous pouvez commencer en utilisant les opérations de l'AWS IoT Core API. Pour de plus amples informations, veuillez consulter [Référence d'API AWS IoT](#). L'AWS IoT Core API sur laquelle les AWS IoT jobs sont créés est prise en charge par le AWS SDK. Pour plus d'informations, veuillez consulter Kits [AWSSDK et boîtes à outils](#).

Vous pouvez utiliser les commandes AWS CLI pour exécuter afin de créer et de gérer des tâches et des modèles de tâches. Pour plus d'informations, veuillez consulter la [référence de l'AWS IoT Interface](#) de ligne de commande.

AWS IoT Emplois Régions et points de terminaison

AWS IoT Jobs prend en charge les points de terminaison de l'API du plan de contrôle et du plan de données spécifiques à votre Région AWS. Les points de terminaison de l'API du plan de données sont spécifiques à votre Compte AWS et Région AWS. Pour plus d'informations sur les points de terminaison des AWS IoT tâches, voir [AWS IoT Device Management- Points de terminaison des données des tâches](#) dans la Référence AWS générale.

Qu'est-ce que AWS IoT Jobs ?

Utilisez AWS IoT Jobs pour définir un ensemble d'opérations à distance qui peuvent être envoyées et exécutées sur un ou plusieurs appareils connectés AWS IoT.

Pour créer des tâches, commencez par définir un document de travail qui contient une liste d'instructions décrivant les opérations que l'appareil doit effectuer à distance. Pour effectuer ces opérations, spécifiez une liste de cibles, qui sont des objets individuels, [des groupes \(p. 294\)](#) d'objets ou les deux. Le document de travail et les objectifs constituent ensemble un déploiement.

Chaque déploiement peut comporter des configurations supplémentaires :

- Déploiement : cette configuration définit le nombre d'appareils qui reçoivent le document de travail chaque minute.
- Abandonner : si un certain nombre d'appareils ne reçoivent pas la notification de tâche, utilisez cette configuration pour annuler la tâche. Cela évite d'envoyer une mauvaise mise à jour à l'ensemble d'un parc.
- Délai d'expiration : si aucune réponse n'est reçue de la part de vos cibles de travail dans un certain délai, la tâche peut échouer. Vous pouvez suivre la tâche en cours d'exécution sur ces appareils.
- Réessayer : si un appareil signale une panne ou si une tâche expire, vous pouvez utiliser AWS IoT Jobs pour renvoyer automatiquement le document de travail à l'appareil.
- Planification : Cette configuration vous permet de planifier une tâche pour une date et une heure future. Il vous permet également de créer des fenêtres de maintenance récurrentes qui mettent à jour les appareils pendant des périodes prédéfinies à faible trafic.

AWS IoT Jobs envoie un message pour informer les cibles qu'une tâche est disponible. La cible démarre l'exécution de la tâche par le téléchargement du document de tâche, l'exécution des opérations spécifiées et le rapport de son avancement à AWS IoT. Vous pouvez suivre la progression d'une tâche pour une cible spécifique ou pour toutes les cibles en exécutant les commandes fournies par AWS IoT Jobs. Lorsqu'une tâche démarre, elle a le statut En cours. Les appareils signalent ensuite les mises à jour incrémentielles tout en affichant cet état jusqu'à ce que la tâche réussisse, échoue ou expire.

Les rubriques suivantes décrivent certains concepts clés des tâches ainsi que le cycle de vie des tâches et de leur exécution.

Rubriques

- [Concepts clés relatifs aux tâches \(p. 740\)](#)
- [Tâches et états d'exécution des tâches \(p. 743\)](#)

Concepts clés relatifs aux tâches

Les concepts suivants fournissent des détails sur les AWS IoT tâches et sur la manière de créer et de déployer des tâches pour exécuter des opérations à distance sur vos appareils.

Concepts de base

Vous trouverez ci-dessous les concepts de base que vous devez connaître lorsque vous utilisez AWS IoT Jobs.

Job

Une tâche est une opération à distance qui est envoyée et exécutée sur un ou plusieurs appareils connectés AWS IoT. Par exemple, vous pouvez définir une tâche qui demande à un ensemble d'appareils de télécharger et d'installer une application ou d'exécuter des mises à jour du microprogramme, de redémarrer, de faire pivoter des certificats ou d'effectuer des opérations de dépannage à distance.

Document de Job

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 Job sont des documents JSON codés en UTF-8 et contiennent les informations dont vos appareils ont besoin pour effectuer une tâche. Un document de travail contient une ou plusieurs URL sur lesquelles l'appareil peut télécharger une mise à jour ou d'autres données. Le

document de tâche peut être stockée dans un compartiment Amazon S3 ou inclus en ligne avec la commande qui crée la tâche.

Tip

Pour des exemples de documents de travail, consultez l'exemple [jobs-agent.js](#) dans le AWS IoT SDK pour JavaScript.

Cible

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 \(p. 294\)](#), ou les deux. Le service AWS IoT Jobs envoie un message pour informer chaque cible qu'une tâche est disponible.

Déploiement

Une fois que vous avez créé un travail en fournissant le document de travail et en spécifiant votre liste de cibles, le document de travail est ensuite déployé sur les machines cibles distantes pour lesquelles vous souhaitez effectuer la mise à jour. Pour les tâches de capture instantanée, la tâche se terminera après le déploiement sur les machines cibles. Pour les tâches continues, une tâche est déployée sur un groupe d'appareils au fur et à mesure qu'ils sont ajoutés aux groupes.

Exécution du Job

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 AWS IoT Jobs fournit des commandes permettant de suivre la progression de l'exécution d'une tâche sur une cible et la progression d'une tâche sur toutes les cibles.

Concepts types d'Job

Les concepts suivants peuvent vous aider à mieux comprendre les différents types de tâches que vous pouvez créer avec AWS IoT Jobs.

Tâche d'instantané

Par défaut, une tâche est envoyée à toutes les cibles que vous spécifiez lorsque vous créez la tâche. Une fois que ces cibles ont terminé le travail (ou signalent qu'elles ne sont pas en mesure de le faire), le travail est terminé.

Travail continu

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 des appareils au fur et à mesure qu'ils sont ajoutés à un groupe. Vous pouvez rendre une tâche continue en définissant un paramètre facultatif lors de la création de la tâche.

Note

Lorsque vous ciblez votre parc IoT à l'aide de groupes d'objets dynamiques, nous vous recommandons d'utiliser des tâches continues plutôt que des tâches instantanées. En utilisant des tâches continues, les appareils qui rejoignent le groupe reçoivent l'exécution de la tâche même après sa création.

URL présignalées

Pour un accès sécurisé et limité dans le temps aux données qui ne figurent pas dans le document de travail, vous pouvez utiliser des URL Amazon S3 présignées. Placez vos données dans un compartiment Amazon S3 et ajoutez un lien réservé aux données dans le document de travail. Lorsque AWS IoT Jobs reçoit une demande concernant le document de travail, il analyse le document en recherchant les liens d'espace réservé, puis remplace les liens par des URL Amazon S3 présignées.

Le lien de l'espace réservé est au format suivant :

```
${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 présignées ne fonctionnent que si le propriétaire de la ressource possède une licence ICP (fournisseur de contenu Internet). Pour plus d'informations, consultez [Amazon Simple Storage Service](#) dans la documentation [Getting Started with AWS Services in China](#).

Concepts de Job

Les concepts suivants peuvent vous aider à comprendre comment configurer des tâches.

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. Vous pouvez créer une configuration de déploiement en utilisant un taux de déploiement statique ou un taux de déploiement exponentiel. Pour spécifier le nombre maximum de cibles de tâches à informer par minute, utilisez un taux de déploiement statique.

Pour obtenir des exemples de définition des taux de déploiement et pour plus d'informations sur la configuration des déploiements de tâches, consultez [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)

Planification

La planification des Job vous permet de planifier le calendrier de déploiement d'un document de travail sur tous les appareils du groupe cible pour les tâches continues et instantanées. En outre, vous pouvez créer une fenêtre de maintenance facultative contenant les dates et heures spécifiques auxquelles une tâche déployera le document de travail sur tous les appareils du groupe cible. Une fenêtre de maintenance est une instance récurrente avec une fréquence de dates et d'heures quotidiennes, hebdomadaires, mensuelles ou personnalisées sélectionnées lors de la création de la tâche initiale ou du modèle de tâche. Seules des tâches continues peuvent être planifiées pour effectuer un déploiement pendant une fenêtre de maintenance.

La planification des tâches est spécifique à votre travail. Les exécutions de Job individuelles ne peuvent pas être planifiées. Pour plus d'informations, veuillez consulter [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#).

Avorter

Vous pouvez créer un ensemble de conditions pour annuler les déploiements lorsque les critères que vous avez spécifiés sont remplis. Pour plus d'informations, veuillez consulter [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#).

Délais d'expiration

Les délais d'expiration d'une tâche vous avertissent chaque fois qu'un déploiement de tâches reste bloqué dans l'IN_PROGRESS état pendant une période étonnamment longue. Il existe deux types de minuteurs : minuteurs d'avancement et minuteurs d'étape. Une fois la tâche terminée IN_PROGRESS, vous pouvez surveiller et suivre la progression du déploiement de votre tâche.

Les configurations de déploiement et d'abandon sont spécifiques à votre tâche, tandis que la configuration du délai d'expiration est spécifique au déploiement d'une tâche. Pour plus d'informations, veuillez consulter [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#).

Réessais

Les nouvelles tentatives de Job permettent de réessayer l'exécution d'une tâche lorsqu'une tâche échoue, expire ou les deux. Vous pouvez avoir jusqu'à 10 tentatives pour exécuter la tâche. Vous pouvez surveiller et suivre la progression de votre nouvelle tentative et déterminer si l'exécution de la tâche a réussi.

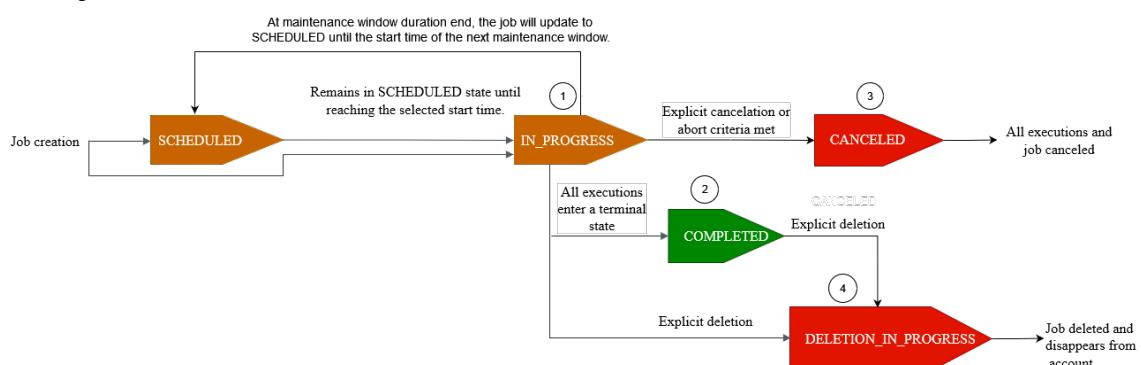
Les configurations de déploiement et d'abandon sont spécifiques à votre tâche, tandis que les configurations de délai d'expiration et de nouvelle tentative sont spécifiques à l'exécution d'une tâche. Pour plus d'informations, veuillez consulter [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#).

Tâches et états d'exécution des tâches

Les sections suivantes décrivent le cycle de vie d'une AWS IoT tâche et le cycle de vie de son exécution.

États de l'Job

Le diagramme suivant illustre les différents états d'une AWS IoT tâche.



Une tâche que vous créez à l'aide de AWS IoT Jobs peut avoir l'un des états suivants :

- PRÉVU

Lors de la création initiale de la tâche ou du modèle de tâche à l'aide de la AWS IoT console, [CreateJob](#)de [CreateJobTemplate](#)l'API ou de l'API, vous pouvez sélectionner la configuration de planification facultative SchedulingConfig dans la AWS IoT console ou dans l'[CreateJob](#)API ou l'[CreateJobTemplate](#)API. Lorsque vous démarrez une tâche planifiée contenant un, et endBehavoir spécifique startTimeendTime, le statut de la tâche est mis à jour enSCHEIVED. Lorsque la tâche atteint la fenêtre de maintenance que vous avez sélectionnée startTime ou la fenêtre startTime de maintenance suivante (si vous avez sélectionné le déploiement de la tâche pendant une fenêtre de maintenance), l'état sera mis à jour SCHEIVED à partir de IN_PROGRESS et le déploiement du document de travail sur tous les appareils du groupe cible commencera.

- IN_PROGRESS

Lorsque vous créez une tâche à l'aide de la AWS IoT console ou de l'[CreateJob](#)API, le statut de la tâche est mis à jour àIN_PROGRESS. Lors de la création d'une tâche, AWS IoT Jobs commence à déployer des exécutions de tâches sur les appareils de votre groupe cible. Une fois toutes les exécutions de tâches effectuées, AWS IoT Jobs attend que les appareils terminent l'action à distance.

Pour plus d'informations sur la concurrence et les limites qui s'appliquent aux tâches en cours, consultez [Limites des tâches \(p. 845\)](#)

Note

Lorsqu'une IN_PROGRESS tâche atteint la fin de la fenêtre de maintenance en cours, le déploiement du document de travail s'arrête. La tâche sera mise à jour SCHEDULED jusqu'à startTime la prochaine fenêtre de maintenance.

- TERMINÉ

Une tâche continue est gérée de l'une des manières suivantes :

- Dans le cas d'une tâche continue sans la configuration de planification facultative sélectionnée, elle est toujours en cours et continue de s'exécuter pour tous les nouveaux appareils ajoutés au groupe cible. Il n'atteindra jamais l'état de COMPLETED.
- Pour une tâche continue avec la configuration de planification facultative sélectionnée, ce qui suit est vrai :
 - Si un endTime a été fourni, une tâche continue atteindra le COMPLETED statut lorsqu'elle sera endTime terminée et que toutes les exécutions de tâches auront atteint l'état de terminal.
 - Si aucun élément n'endTime a été fourni dans la configuration de planification facultative, la tâche continue continuera à exécuter le déploiement du document de travail.

Pour une tâche de capture instantanée, l'état de la tâche change COMPLETED lorsque toutes ses exécutions entrent dans un état terminal SUCCEEDED, tel que FAILEDTIMED_OUT, REMOVED, ou CANCELED.

- ANNULÉE

Lorsque vous annulez une tâche à l'aide de la AWS IoT console, de l'[CancelJob API](#) ou du [Configuration de l'abandon d'une Job \(p. 785\)](#), le statut de la tâche passe à CANCELED. Lors de l'annulation d'une tâche, AWS IoT Jobs commence à annuler les exécutions de tâches créées précédemment.

Pour plus d'informations sur la simultanéité et les limites applicables aux tâches annulées, consultez [Limites des tâches \(p. 845\)](#).

- SUPPRESSION_EN COURS

Lorsque vous supprimez une tâche à l'aide de la AWS IoT console ou de l'[DeleteJob API](#), le statut de la tâche devient DELETION_IN_PROGRESS. Lors de la suppression d'une tâche, AWS IoT Jobs commence à supprimer les exécutions de tâches créées précédemment. Une fois que toutes les exécutions de tâches ont été supprimées, la tâche disparaît de votre AWS compte.

États d'exécution de tâche

Le tableau suivant indique les différents états d'exécution d'une AWS IoT tâche et indique si le changement d'état est initié par l'appareil ou par les AWS IoT tâches.

État et source de l'exécution des Job

État d'exécution de tâche	Initié par l'appareil ?	Initié par AWS IoT Jobs ?	État du terminal ?	Peut être réessayé ?
QUEUED	Non	Oui	Non	Ne s'applique pas
IN_PROGRESS	Oui	Non	Non	Ne s'applique pas
SUCCEEDED	Oui	Non	Oui	Ne s'applique pas
FAILED	Oui	Non	Oui	Oui

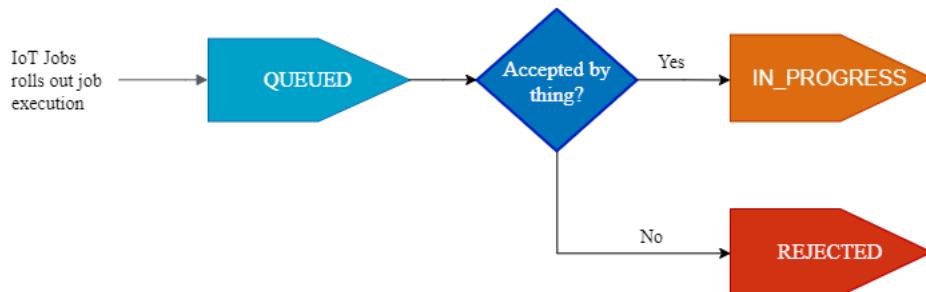
État d'exécution de tâche	Initié par l'appareil ?	Initié par AWS IoT Jobs ?	État du terminal ?	Peut être réessayé ?
TIMED_OUT	Non	Oui	Oui	Oui
REJECTED	Oui	Non	Oui	Non
REMOVED	Non	Oui	Oui	Non
CANCELED	Non	Oui	Oui	Non

La section suivante décrit plus en détail les états d'exécution d'une tâche déployée lorsque vous créez une tâche avec AWS IoT Jobs.

- **QUEUED**

Lorsque AWS IoT Jobs déploie une exécution de tâche pour une machine cible, l'état d'exécution de la tâche est défini sur **QUEUED**. L'exécution de la tâche reste en l'**QUEUED** état jusqu'à ce que :

- Votre appareil reçoit l'exécution de la tâche, appelle les opérations de l'API Jobs et indique l'état sous **IN_PROGRESS** la forme.
- Vous annulez la tâche ou son exécution, ou lorsque les critères d'abandon que vous avez spécifiés sont remplis et que le statut passe à **CANCELED**.
- Votre appareil est retiré du groupe cible et son état passe à **REMOVED**.



- **IN_PROGRESS**

Si votre appareil IoT s'abonne à l'option réservée [Rubriques de tâche \(p. 124\)](#) \$notify et qu'\$notify-next il appelle l'`IStartPendingJobExecutionAPI` ou l'API dont le `UpdateJobExecution` statut est **IN_PROGRESS**, AWS IoT Jobs définira le statut d'exécution de la tâche sur **IN_PROGRESS**.

L'`UpdateJobExecutionAPI` peut être invoquée plusieurs fois avec un statut de **IN_PROGRESS**. Vous pouvez spécifier des détails supplémentaires sur les étapes d'exécution à l'aide de `l'statusDetails` objet.

Note

Si vous créez plusieurs tâches pour chaque appareil, les AWS IoT tâches et le protocole MQTT ne garantissent pas l'ordre de livraison.

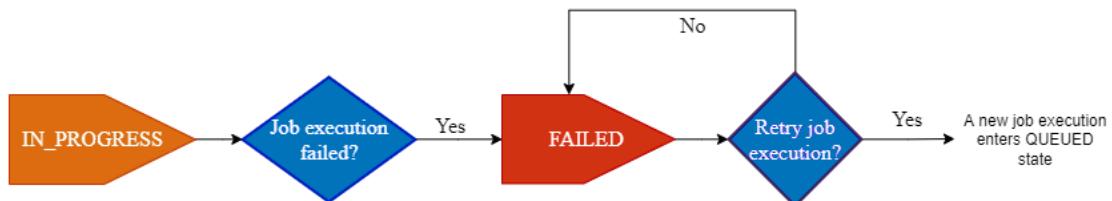
- **RÉUSSI**

Lorsque votre appareil termine avec succès l'opération à distance, il doit appeler l'`UpdateJobExecutionAPI` avec un état de **SUCCEEDED** pour indiquer que l'exécution de la tâche a réussi. AWS IoT Les tâches sont ensuite mises à jour et renvoient l'état d'exécution des tâches sous la forme **SUCCEEDED**.



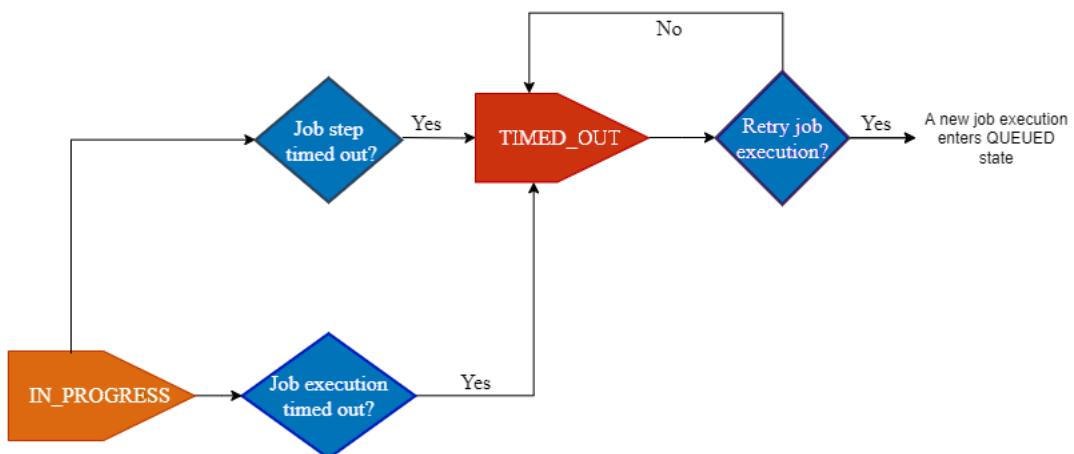
- ÉCHEC

Lorsque votre appareil ne parvient pas à terminer l'opération à distance, il doit appeler l'UpdateJobExecutionAPI avec un état de Failed pour indiquer que l'exécution de la tâche a échoué. AWS IoT Les tâches sont ensuite mises à jour et renvoient l'état d'exécution des tâches sous la formeFailed. Vous pouvez réessayer d'exécuter cette tâche pour l'appareil à l'aide du[Configuration de nouvelle tentative d'exécution de Job \(p. 787\)](#).



- TIMED_OUT

Lorsque votre appareil ne parvient pas à terminer une étape de tâche alors que son état est le casIN_PROGRESS, ou lorsqu'il ne parvient pas à terminer l'opération à distance dans le délai imparti, AWS IoT Jobs définit l'état d'exécution de la tâche sur. TIMED_OUT Vous disposez également d'un chronomètre pour chaque étape d'une tâche en cours, qui s'applique uniquement à l'exécution de la tâche. La durée du temporisateur en cours est spécifiée à l'aide de la inProgressTimeoutInMinutes propriété du[Configuration du délai d'attente d'exécution de Job \(p. 785\)](#). Vous pouvez réessayer d'exécuter cette tâche pour l'appareil à l'aide du[Configuration de nouvelle tentative d'exécution de Job \(p. 787\)](#).



- REJETÉS

Lorsque votre appareil reçoit une demande non valide ou incompatible, il doit appeler l'UpdateJobExecutionAPI avec un statut deREJECTED. AWS IoT Les tâches sont ensuite mises à jour et renvoient l'état d'exécution des tâches sous la formeREJECTED.

- ENLEVÉ

Lorsque votre appareil n'est plus une cible valide pour l'exécution de la tâche, par exemple lorsqu'il est détaché d'un groupe d'objets dynamiques, AWS IoT Jobs définit l'état d'exécution de la tâche surREMOVED. Vous pouvez rattacher l'objet à votre groupe cible et redémarrer l'exécution du travail pour l'appareil.

- ANNULÉE

Lorsque vous annulez une tâche ou annulez l'exécution d'une tâche à l'aide de la console CancelJob ou de l'CancelJobExecutionAPI, ou lorsque les critères d'abandon spécifiés à l'aide du [Configuration de l'abandon d'une Job \(p. 785\)](#) sont remplis, AWS IoT Jobs annule la tâche et définit l'état d'exécution de la tâche sur. CANCELED

Gestion des tâches

Utilisez des tâches pour informer les appareils de la mise à jour d'un logiciel ou d'un microprogramme. Vous pouvez utiliser la [AWS IoT Console Gestion des Job et opérations de l'API de contrôle \(p. 813\)](#), les kits SDK ou les [AWS Command Line Interface AWSkits SDK](#) pour créer et gérer des tâches.

Signature de code pour les tâches

Lorsque vous envoyez du code à des appareils, pour que les appareils puissent détecter si le code a été modifié pendant le transport, nous vous recommandons de signer le fichier de code à l'aide du AWS CLI. Pour obtenir des instructions, voir [Création et gestion de tâches à l'aide du AWS CLI \(p. 751\)](#).

Pour plus d'informations, consultez À [quoi sert la signature de code AWS IoT ?](#).

Document de Job

Avant de créer une tâche, vous devez créer un document de tâche. Si vous utilisez la signature de code pour AWS IoT, vous devez télécharger votre document de tâche dans un compartiment Amazon S3 versionné. Pour plus d'informations sur la création d'un compartiment Amazon S3 et le chargement de fichiers dans ce compartiment, veuillez consulter [Mise en route sur Amazon Simple Storage Service](#) dans le Guide de démarrage Amazon S3.

Tip

Pour des exemples de documents de travail, consultez l'exemple [jobs-agent.js](#) dans le AWS IoT SDK pour JavaScript.

URL présignées

Votre document de travail peut contenir une URL Amazon S3 présignée pointant vers votre fichier de code (ou un autre fichier). Les URL Amazon S3 présignées ne sont valides que pour une durée limitée et sont générées lorsqu'un appareil demande un document de travail. Étant donné que l'URL présignée n'est pas créée lorsque vous créez le document de travail, utilisez plutôt une URL de remplacement dans votre document de travail. Une URL d'espace réservé se présente comme suit :

```
 ${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/<bucket>/<code file>}
```

où :

- **compartiment** est le compartiment Amazon S3 qui contient le fichier de code.
- **le fichier de code** est 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.

Rôle IAM pour accorder l'autorisation de télécharger des fichiers depuis S3

Lorsque vous créez une tâche utilisant des URL Amazon S3 présignées, vous devez fournir un rôle IAM. Le rôle doit accorder l'autorisation de télécharger des fichiers depuis le compartiment Amazon S3 où sont stockées les données ou les mises à jour. 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, reportez-vous à la section [CreateJob](#).

Accordez à AWS IoT Jobs l'autorisation d'assumer votre rôle

1. Accédez au [hub Rôles de la console IAM](#) et choisissez votre rôle.
2. Dans l'onglet Relations de confiance, choisissez Modifier la relation de confiance et remplacez le document de politique par le code JSON suivant. Choisissez Update Trust Policy (Mettre à jour la politique d'approbation).

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

3. Pour éviter tout problème de confusion chez les adjoints, ajoutez les clés contextuelles de la condition globale [aws:SourceArn](#) et [aws:SourceAccount](#) à la politique.

Important

Vous aws:SourceArn devez respecter le format :arn:aws:iot:*region:account-id*:*. Assurez-vous que *la région correspond à votre AWS IoT région* et que l'identifiant de *compte correspond à l'identifiant* de votre compte client. Pour plus d'informations, consultez [Prévention du problème de l'adjoint confus entre services \(p. 368\)](#).

```
{  
    "Effect": "Allow",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "Service":  
                    "iot.amazonaws.com"  
            },  
            "Action": "sts:AssumeRole",  
            "Condition": {  
                "StringEquals": {  
                    "aws:SourceAccount": "123456789012"  
                },  
                "ArnLike": {  
                    "aws:SourceArn": "arn:aws:iot:*:123456789012:job/*"  
                }  
            }  
        }  
    ]  
}
```

```
        }
    }
}
```

4. Si votre tâche utilise un document de travail qui est un objet Amazon S3, choisissez Autorisations et utilisez le code JSON suivant. Cela ajoute une politique qui autorise le téléchargement de fichiers depuis votre compartiment Amazon S3 :

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::your_S3_bucket/*"
    }
  ]
}
```

Rubriques

- [Créez et gérez des tâches à l'aide du AWS Management Console \(p. 749\)](#)
- [Créez et gérez des tâches à l'aide du AWS CLI \(p. 751\)](#)

Créez et gérez des tâches à l'aide du AWS Management Console

Pour créer une tâche

1. Connectez-vous à la AWS IoT console AWS Management Console et connectez-vous à celle-ci.
2. Dans le volet de navigation de gauche, dans la section Gérer, choisissez Actions à distance, puis Jobs.
3. Sur la page Tâches de la boîte de dialogue Tâches, choisissez Créer une tâche.
4. Selon l'appareil que vous utilisez, vous pouvez créer une tâche personnalisée, une tâche de mise à jour de FreeRTOS OTA ou une AWS IoT Greengrass tâche. Pour cet exemple, choisissez Create a custom job. Choisissez Suivant.
5. Sur la page Propriétés personnalisées de la Job, dans la boîte de dialogue Propriétés de la tâche, entrez vos informations pour les champs suivants :
 - Nom : Entrez un nom de travail alphanumérique unique.
 - Description : facultatif : entrez une description facultative de votre Job.
 - Étiquettes - facultatives :

Note

Nous vous recommandons de ne pas utiliser d'informations personnelles identifiables dans vos ID de poste et votre description de poste.

Choisissez Suivant.

6. Sur la page Configuration du fichier, dans la boîte de dialogue Cibles des Job, sélectionnez les objets ou les groupes d'objets que vous souhaitez exécuter cette tâche.

Dans la boîte de dialogue Document de Job, sélectionnez l'une des options suivantes :

- Depuis le fichier : fichier de tâche JSON que vous avez chargé précédemment dans un compartiment Amazon S3
- Signature de code

Dans le document de travail situé dans votre URL Amazon S3, \${aws:iot:code-sign-signature:s3://region.bucket/code-file@code-file-version-id} est obligatoire comme espace réservé jusqu'à ce qu'il soit remplacé par le chemin du fichier de code signé à l'aide de votre profil de signature de code. Le nouveau fichier de code signé apparaîtra initialement dans un SignedImages dossier de votre compartiment source Amazon S3. Un nouveau document de travail contenant un Codesigned_ préfixe sera créé avec le chemin du fichier de code signé remplaçant l'espace réservé au signe de code et placé dans votre URL Amazon S3 pour créer une nouvelle tâche.

- URL des ressources à signer au préalable

Dans la liste déroulante des rôles de présignature, choisissez le rôle IAM que vous avez créé dans les URL [présignées](#). L'utilisation \${aws:iot:s3-presigned-url} de la présignature d'URL pour des objets situés dans Amazon S3 constitue une bonne pratique de sécurité pour les appareils qui téléchargent des objets depuis Amazon S3.

Si vous souhaitez utiliser des URL présignées comme espace réservé à la signature de code, utilisez l'exemple de modèle suivant :

```
${aws:iot:s3-presigned-url:${aws:iot:code-sign-signature:<S3 URL>}}
```

- À partir du modèle : modèle de travail contenant un document de travail et des configurations de travail. Le modèle de travail peut être un modèle de travail personnalisé que vous avez créé ou un modèle AWS géré.

Si vous créez une tâche pour effectuer des actions à distance fréquemment utilisées, telles que le redémarrage de votre appareil, vous pouvez utiliser un modèle AWS géré. Ces modèles ont déjà été préconfigurés pour être utilisés. Pour plus d'informations, consultez [Création d'un modèle de travail personnalisé \(p. 773\)](#) et [Création de modèles de tâches personnalisés à partir de modèles gérés \(p. 769\)](#).

7. Sur la page Configuration des Job, dans la boîte de dialogue Configuration des Job, sélectionnez l'un des types de tâches suivants :
 - Tâche de capture instantanée : une tâche de capture instantanée est terminée lorsqu'elle a terminé son exécution sur les appareils et les groupes cibles.
 - Tâche continue : une tâche continue s'applique à des groupes d'objets et s'exécute sur n'importe quel appareil que vous ajoutez ultérieurement à un groupe cible spécifié.
8. Dans la boîte de dialogue Configurations supplémentaires - facultative, passez en revue les configurations de Job facultatives suivantes et effectuez vos sélections en conséquence :
 - Configuration du déploiement
 - Configuration de la planification
 - Configuration du délai d'exécution des Job
 - Configuration des nouvelles tentatives d'exécution des Job : nouveau
 - Abandonner la configuration

Reportez-vous aux sections suivantes pour plus d'informations sur les configurations des Job :

- [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
- [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Passez en revue toutes vos sélections d'emplois, puis choisissez Soumettre pour créer votre travail.

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. Pour gérer les tâches, accédez au [hub de Job de la console](#).

Créez et gérez des tâches à l'aide du AWS CLI

Cette section décrit comment créer et gérer des tâches.

Création de tâches

Pour créer une AWS IoT tâche, utilisez la CreateJob commande. 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 une AWS IoT tâche, vous avez besoin d'un document de travail qui peut être inclus dans le corps de la demande ou sous forme de 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 rôle IAM Amazon Resource Name (ARN) autorisé à télécharger le fichier et autorisant le service AWS IoT Jobs à assumer ce rôle.

Signature de code avec des tâches

Si vous utilisez la signature de code pour AWS IoT, vous devez démarrer une tâche de signature de code et inclure le résultat dans votre document de travail. Cela remplacera l'espace réservé à la signature par code dans votre document de travail, qui est obligatoire jusqu'à ce qu'il soit remplacé par le chemin du fichier de code signé à l'aide de votre profil de signature de code. L'espace réservé pour la signature de code se présente comme suit :

```
${aws:iot:code-sign-signature:s3://region.bucket/code-file@code-file-version-id}
```

Utilisez la [start-signing-job](#) commande pour créer une tâche de signature de code. start-signing-job renvoie un identifiant de tâche. Pour obtenir l'emplacement Amazon S3 où la signature est stockée, utilisez la describe-signing-job commande. Vous pouvez ensuite télécharger la signature depuis Amazon S3. Pour plus d'informations sur les tâches de signature de code, consultez [Signature de code pour AWS IoT](#).

Votre document de travail doit contenir un espace réservé d'URL présigné pour votre fichier de code et la sortie de signature JSON placée dans un compartiment Amazon S3 à l'aide de la start-signing-job commande suivante :

```
{  
    "presign": "${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/bucket/image}"  
}
```

Créer une tâche avec un document de tâche

La commande suivante montre comment créer une tâche à l'aide d'un document de travail (job-document.json) stocké dans un compartiment Amazon S3 (jobBucket) et d'un rôle autorisé à télécharger des fichiers depuis Amazon S3 (S3). DownloadRole

```
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 \  
    --role ARN
```

```
--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 d'exécution de la tâche n'est pas défini sur un autre état du terminal avant l'expiration du délai, il est défini sur `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 tâche d'exécution reste dans l'`IN_PROGRESS` état plus longtemps que cet intervalle, elle échoue et passe à l'`TIMED_OUT` état terminal. AWS IoT publie également une notification MQTT.

Pour plus d'informations sur la création de configurations pour les déploiements et les abandons de tâches, voir [Configuration du déploiement et de l'abandon des tâches](#).

Note

Les documents de Job spécifiés sous forme de fichiers Amazon S3 sont récupérés au moment où vous créez le travail. Si vous modifiez le contenu du fichier Amazon S3 que vous avez utilisé comme source de votre document de travail après avoir créé le document de travail, le contenu envoyé aux cibles de travail ne change pas.

Mettre à jour une tâche

Pour mettre à jour une tâche, utilisez la `UpdateJob` commande. 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](#).

Annuler une tâche

Pour annuler une tâche, utilisez la `CancelJob` commande. 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 tâches en cours `QUEUED`. AWS IoT conserve toutes les exécutions de tâches dans un état de

terminal intactes car l'appareil a déjà terminé 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 Job qui sont dans un IN_PROGRESS état sont annulées, mais uniquement si vous spécifiez le --force paramètre facultatif. Les exécutions de Job dans un état terminal ne sont pas annulées.

Warning

L'annulation d'une tâche en cours d'exécution (en définissant le --force paramètre) annule toutes les exécutions de tâches en cours et empêche l'appareil qui exécute la tâche de mettre à jour l'IN_PROGRESS état 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 parce qu'elle répond aux critères définis par un AbortConfig objet, le service ajoute des valeurs renseignées automatiquement pour les reasonCode champs comment et. Vous pouvez créer vos propres valeurs pour reasonCode lorsque la tâche d'annulation est orientée utilisateurs.

Annulation d'une exécution de tâche

Pour annuler l'exécution d'une tâche sur un périphérique, utilisez la CancelJobExecution commande. Cela annule l'exécution d'une tâche en cours d'QUEUED état. Si vous souhaitez annuler l'exécution d'une tâche en cours, vous devez utiliser le --force paramètre.

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.

L'exécution d'une tâche qui est dans un QUEUED état est annulée. L'exécution d'une tâche qui est dans un IN_PROGRESS état est annulée, mais uniquement si vous spécifiez le --force paramètre facultatif. Les exécutions de Job dans un état terminal ne peuvent pas être annulées.

Warning

Lorsque vous annulez l'exécution d'une tâche qui est dans l'IN_PROGRESS état, l'appareil ne peut pas mettre à jour l'état d'exécution de la tâche. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Si l'exécution de la tâche est dans un état terminal, ou si l'exécution de la tâche est dans un IN_PROGRESS état et que le --force paramètre n'est pas défini sur true, cette commande provoque un `InvalidStateException`.

Le statut d'une exécution de tâche annulée est cohérent à terme. La modification du statut de l'exécution d'une tâche CANCELED peut prendre un certain temps, en fonction de divers facteurs.

Suppression d'une tâche

Pour supprimer une tâche et ses exécutions de tâche, utilisez la `DeleteJob` commande. Par défaut, vous ne pouvez supprimer qu'une tâche dont l'état est terminal (SUCCEEDED ou CANCELED). Dans le cas contraire, une exception se produit. Vous ne pouvez toutefois supprimer une tâche dans IN_PROGRESS cet état que si le force paramètre est défini sur 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 en cours d'exécution, IN_PROGRESS l'appareil qui déploie la tâche ne peut pas accéder aux informations de la tâche ni mettre à jour l'état d'exécution de la tâche. Faites preuve de prudence et assurez-vous que chaque appareil déployant une tâche supprimée peut revenir à un état correct.

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. Une erreur se produit si vous tentez de supprimer ou d'annuler une tâche dont le statut est déjà défini DELETION_IN_PROGRESS.

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

Pour récupérer un document de travail pour un travail, utilisez la `GetJobDocument` commande. Un document de tâche est une description des opérations distantes à exécuter par les appareils.

Pour obtenir un document de tâche, exécutez la commande suivante :

```
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/  
firmWareUpdate-01\",\\n\t\"data\":\"\${aws:iot:s3-presigned-url:https://s3.amazonaws.com/job-  
test-bucket/datafile}\\n\"}  
}
```

Note

Lorsque vous utilisez cette commande pour récupérer un document de travail, les URL de remplacement ne sont pas remplacées par des URL Amazon S3 présignées. Lorsqu'un appareil appelle l'opération d'[GetPendingJobExecutions](#) API, les URL de substitution sont remplacées par des URL Amazon S3 présignées dans le document de travail.

Affichage des tâches

Pour obtenir la liste de toutes les tâches de votre Compte AWS, utilisez la ListJobs commande. 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

Pour obtenir le statut d'une tâche, exécutez la DescribeJob commande. La commande suivante explique comment décrire une tâche :

```
$ aws iot describe-job --job-id 010
```

La commande renvoie le statut de la tâche spécifiée. Par exemple :

```
{  
    "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,  
            "lastUpdatedAt": 1486687079.743  
        }  
    }  
}
```

```
"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": "PENDING",  
                "progress": 0,  
                "lastUpdatedAt": 1486593195.006,  
                "createdAt": 1486593195.006,  
                "targetSelection": "SNAPSHOT",  
                "rolloutConfiguration": {  
                    "exponentialRate": {  
                        "baseRatePerMinute": 10,  
                        "incrementFactor": 2,  
                        "rateIncreaseCriteria": {  
                            "numberOfNotifiedThings": 10,  
                            "numberOfSucceededThings": 10  
                        }  
                    },  
                    "maximumPerMinute": 10  
                }  
            }  
        }  
    ]  
}
```

```
        "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](#). Par exemple :

```
{  
    "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 de la tâche doit être `QUEUED` ou être 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 l'exécution d'une tâche dont l'état est `IN_PROGRESS`, l'appareil qui exécute la tâche ne peut pas accéder aux informations de la tâche ni mettre à jour l'état

d'exécution de la tâche. Soyez vigilant et vérifiez que l'appareil est en mesure de reprendre un état valide.

Modèles de tâche

Utilisez des modèles de tâches pour préconfigurer des tâches que vous pouvez déployer sur plusieurs ensembles de machines cibles. Pour déployer des actions à distance fréquemment effectuées sur vos appareils, comme le redémarrage ou l'installation d'une application, vous pouvez utiliser des modèles pour définir des configurations standard. Pour effectuer des opérations telles que le déploiement de correctifs de sécurité et de corrections de bogues, vous pouvez créer des modèles à partir de tâches existantes.

Lors de la création d'un modèle de tâche, spécifiez les configurations et ressources supplémentaires suivantes.

- Propriétés Job
- Documents de Job et objectifs
- Critères de déploiement, de planification et d'annulation
- Critères de délai d'expiration et de nouvelle tentative

Modèles personnalisés et AWS gérés

Selon l'action à distance que vous souhaitez effectuer, vous pouvez créer un modèle de tâche personnalisé ou utiliser un modèle AWS géré. Utilisez des modèles de tâches personnalisés pour fournir votre propre document de travail personnalisé et créer des tâches réutilisables à déployer sur vos appareils. AWS les modèles gérés sont des modèles de tâches fournis par AWS IoT Jobs pour les actions fréquemment effectuées. Ces modèles comportent un document de travail prédéfini pour certaines actions à distance afin que vous n'ayez pas à créer votre propre document de travail. Les modèles gérés vous aident à créer des tâches réutilisables pour un lancement plus rapide sur vos appareils.

Rubriques

- [Utilisez des modèles AWS gérés pour déployer des opérations à distance courantes \(p. 759\)](#)
- [Créez des modèles de travail personnalisés \(p. 773\)](#)

Utilisez des modèles AWS gérés pour déployer des opérations à distance courantes

AWS les modèles gérés sont des modèles de tâches fournis par AWS. Ils sont utilisés pour les actions à distance fréquemment effectuées, telles que le redémarrage, le téléchargement d'un fichier ou l'installation d'une application sur vos appareils. Ces modèles comportent un document de travail prédéfini pour chaque action à distance afin que vous n'ayez pas à créer votre propre document de travail.

Vous pouvez choisir parmi un ensemble de configurations prédéfinies et créer des tâches à l'aide de ces modèles sans écrire de code supplémentaire. À l'aide de modèles gérés, vous pouvez consulter le document de travail déployé sur vos flottes. Vous pouvez créer une tâche à l'aide de ces modèles et créer un modèle de tâche personnalisé que vous pouvez réutiliser pour vos opérations à distance.

Que contiennent les modèles gérés ?

Chaque modèle AWS géré contient :

- L'environnement dans lequel exécuter les commandes du document de travail.
- Un document de travail qui spécifie le nom de l'opération et ses paramètres. Par exemple, si vous utilisez un modèle de fichier de téléchargement, le nom de l'opération est Télécharger le fichier et les paramètres peuvent être les suivants :
 - URL du fichier que vous souhaitez télécharger sur votre appareil. Il peut s'agir d'une ressource Internet ou d'une URL Amazon Simple Storage Service (Amazon S3), publiques ou pré-signées.
 - Un chemin de fichier local sur l'appareil pour stocker le fichier téléchargé.

Pour plus d'informations sur les documents de tâche et leurs paramètres, consultez[Modèles gérés d'actions à distance et de documents de travail \(p. 761\)](#).

Note

Le nombre total de modèles de substitution dans un document de tâche doit être inférieur ou égal à dix.

Prérequis

Pour que vos appareils puissent exécuter les actions à distance spécifiées dans le modèle de document de travail géré, vous devez :

- Installez le logiciel spécifique sur votre appareil

Utilisez le logiciel et les gestionnaires de tâches de votre appareil, ou le client de l'AWS IoT appareil. En fonction de votre analyse de rentabilisation, vous pouvez également les exécuter tous les deux de manière à ce qu'ils exécutent des fonctions différentes.

- Utilisez votre propre appareil, le logiciel et les gestionnaires de tâches

Vous pouvez écrire votre propre code pour les appareils en utilisant le gestionnaire Kit SDK des appareils AWS IoT et sa bibliothèque de gestionnaires qui prennent en charge les opérations à distance. Pour déployer et exécuter des tâches, vérifiez que les bibliothèques d'agents de périphérique ont été correctement installées et qu'elles s'exécutent sur les appareils.

Vous pouvez également choisir d'utiliser vos propres principaux qui prennent en charge les opérations à distance. Pour plus d'informations, consultez la section [Exemples de gestionnaires de tâches](#) dans le GitHub référentiel AWS IoT Device Client.

- Utiliser le client de AWS IoT l'appareil

Vous pouvez également installer et exécuter le AWS IoT Device Client sur vos appareils car il prend en charge par défaut l'utilisation de tous les modèles gérés directement depuis la console.

Le Device Client est un logiciel open source écrit en C++ que vous pouvez compiler et installer sur vos appareils IoT intégrés basés sur Linux. Le Device Client possède un client de base et des fonctionnalités discrètes côté client. Le client de base établit la connectivité AWS IoT via le protocole MQTT et peut se connecter aux différentes fonctionnalités côté client.

Pour effectuer des opérations à distance sur vos appareils, utilisez la fonction Tâches côté client du Device Client. Cette fonctionnalité contient un analyseur destiné à recevoir le document de travail et des gestionnaires de tâches qui implémentent les actions à distance spécifiées dans le document de travail. Pour plus d'informations sur le Device Client et ses fonctionnalités, voir [AWS IoT Device Client](#).

Lors de l'exécution sur des appareils, le client de périphérique reçoit le document de travail et possède une implémentation spécifique à la plate-forme qu'il utilise pour exécuter des commandes dans le document. Pour plus d'informations sur la configuration de l'appareil client et l'utilisation de la fonction Job, consultez les [AWS IoT didacticiels](#).

- Utiliser un environnement pris en charge

Pour chaque modèle géré, vous trouverez des informations sur l'environnement que vous pouvez utiliser pour exécuter les actions à distance. Nous vous recommandons d'utiliser le modèle avec un environnement Linux compatible, comme indiqué dans le modèle. Utilisez le AWS IoT Device Client pour exécuter les actions à distance du modèle géré, car il prend en charge les microprocesseurs et les environnements Linux courants, tels que Debian et Ubuntu.

Modèles gérés d'actions à distance et de documents de travail

La section suivante répertorie les différents modèles AWS gérés pour les AWS IoT tâches et décrit les actions à distance qui peuvent être effectuées sur les appareils. La section suivante contient des informations sur le document de travail et une description des paramètres du document de travail pour chaque action à distance. Le logiciel côté appareil utilise le nom du modèle et les paramètres pour effectuer l'action à distance.

AWS les modèles gérés acceptent des paramètres d'entrée pour lesquels vous spécifiez une valeur lors de la création d'une tâche à l'aide du modèle. Tous les modèles gérés ont deux paramètres d'entrée facultatifs en commun : `runAsUser` et `pathToHandler`. À l'exception du AWS-Reboot modèle, les modèles nécessitent des paramètres d'entrée supplémentaires pour lesquels vous devez spécifier une valeur lors de la création d'une tâche à l'aide du modèle. Ces paramètres requis varient en fonction du modèle que vous choisissez. Par exemple, si vous choisissez le AWS-Download-File modèle, vous devez spécifier une liste de packages à installer et une URL à partir de laquelle télécharger les fichiers.

Spécifiez une valeur pour les paramètres d'entrée lorsque vous utilisez la AWS IoT console ou le AWS Command Line Interface (AWS CLI) pour créer une tâche utilisant un modèle géré. Lorsque vous utilisez l'interface de ligne de commande, fournissez ces valeurs à l'aide de l'`document-parameters` objet. Pour plus d'informations, consultez [documentParameters](#).

Note

À utiliser `document-parameters` uniquement lors de la création de tâches à partir de modèles AWS gérés. Ce paramètre ne peut pas être utilisé avec des modèles de tâches personnalisés ni pour créer des tâches à partir de ceux-ci.

Vous trouverez ci-dessous une description des paramètres d'entrée facultatifs courants. Vous trouverez une description des autres paramètres d'entrée requis par chaque modèle géré dans la section suivante.

`runAsUser`

Ce paramètre indique s'il faut exécuter le gestionnaire de tâches en tant qu'autre utilisateur. Si cela n'est pas spécifié lors de la création de la tâche, le gestionnaire de tâches est exécuté sous le même nom d'utilisateur que le client de l'appareil. Lorsque vous exécutez le gestionnaire de tâches en tant qu'autre utilisateur, spécifiez une valeur de chaîne ne dépassant pas 256 caractères.

`pathToHandler`

Le chemin d'accès au gestionnaire de tâches qui s'exécute sur l'appareil. S'il n'est pas spécifié lors de la création de la tâche, le client de l'appareil utilise le répertoire de travail actuel.

Vous trouverez ci-dessous les différentes actions à distance, leurs documents de travail et les paramètres qu'elles acceptent. Tous ces modèles prennent en charge l'environnement Linux pour exécuter l'opération à distance sur l'appareil.

AWS—Télécharger—Fichier

Nom du modèle

AWS-Download-File

Description du modèle

Modèle géré fourni par AWS pour le téléchargement d'un fichier.

Paramètres d'entrée

Ce modèle comporte les paramètres requis suivants. Vous pouvez également spécifier les paramètres facultatifs `runAsUser` et `pathToHandler`.

downloadUrl

URL à partir de laquelle télécharger le fichier. Il peut s'agir d'une ressource Internet, d'un objet d'Amazon S3 accessible au public ou d'un objet d'Amazon S3 accessible uniquement par votre appareil à l'aide d'une URL présignée. Pour plus d'informations sur l'utilisation des URL présignées et l'octroi d'autorisations, consultez. [URL présignées \(p. 747\)](#)

filePath

Un chemin de fichier local qui indique l'emplacement du fichier téléchargé sur l'appareil.

Comportement du dispositif

L'appareil télécharge le fichier à partir de l'emplacement spécifié, vérifie que le téléchargement est terminé et le stocke localement.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script shell que le gestionnaire de tâches doit exécuter pour télécharger le fichier. `download-file.sh` Il affiche également les paramètres requis `downloadUrl` et `filePath`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Download-File",  
        "type": "runHandler",  
        "input": {  
          "handler": "download-file.sh",  
          "args": [  
            "${aws:iot:parameter:downloadUrl}",  
            "${aws:iot:parameter:filePath}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Installation—Application

Nom du modèle

AWS-Install-Application

Description du modèle

Modèle géré fourni par AWS pour l'installation d'une ou de plusieurs applications.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant,`packages`. Vous pouvez également préciser les paramètres suivants `runAsUser` et `pathToHandler`.

`packages`

Liste, séparées par des espaces, d'une ou plusieurs applications à installer.

Comportement du dispositif

L'appareil installe les applications comme indiqué dans le document de travail.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script shell que le gestionnaire de tâches doit exécuter pour télécharger le fichier `install-packages.sh`. Il affiche également le paramètre `requisitePackages`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Install-Application",  
        "type": "runHandler",  
        "input": {  
          "handler": "install-packages.sh",  
          "args": [  
            "${aws:iot:parameter:packages}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Redémarrer

Nom du modèle

AWS-Reboot

Description du modèle

Un modèle géré fourni par AWS pour redémarrer votre appareil.

Paramètres d'entrée

Ce modèle n'a aucun paramètre requis. Vous pouvez spécifier les paramètres facultatifs `runAsUser` et `pathToHandler`.

Comportement du dispositif

L'appareil redémarre correctement.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script shell que le gestionnaire de tâches doit exécuter pour redémarrer l'appareil. `reboot.sh`

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Reboot",  
        "type": "runHandler",  
        "input": {  
          "handler": "reboot.sh",  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Supprimer—Application

Nom du modèle

AWS-Remove-Application

Description du modèle

Modèle géré fourni par AWS pour la désinstallation d'une ou de plusieurs applications.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant,`packages`. Vous pouvez également préciser les paramètres suivants `runAsUser` et `pathToHandler`.

`packages`

Liste, séparées par des espaces, d'une ou plusieurs applications à désinstaller.

Comportement du dispositif

L'appareil désinstalle les applications comme indiqué dans le document de travail.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script shell que le gestionnaire de tâches doit exécuter pour télécharger le fichier. `remove-packages.sh` Il affiche également le paramètre `requispackages`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Remove-Application",  
        "type": "runHandler",  
        "input": {  
          "handler": "remove-packages.sh",  
          "args": [  
            "${aws:iot:parameter:packages}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

```
    ]  
}
```

AWS—Redémarrer — Application

Nom du modèle

AWS-Restart-Application

Description du modèle

Modèle géré fourni par AWS pour arrêter et redémarrer un ou plusieurs services.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant, `services`. Vous pouvez également préciser les paramètres suivants `runAsUser` et `pathToHandler`.

Services

Liste, séparées par des espaces, d'une ou plusieurs applications à redémarrer.

Comportement du dispositif

Les applications spécifiées sont arrêtées puis redémarrées sur l'appareil.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script `shellrestart-services.sh`, que le gestionnaire de tâches doit exécuter pour redémarrer les services système. Il affiche également le paramètre `requiservices`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Restart-Application",  
        "type": "runHandler",  
        "input": {  
          "handler": "restart-services.sh",  
          "args": [  
            "${aws:iot:parameter:services}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Démarrer—Application

Nom du modèle

AWS-Start-Application

Description du modèle

Modèle géré fourni par AWS pour démarrer un ou plusieurs services.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant, `services`. Vous pouvez également préciser les paramètres suivants `runAsUser` et `pathToHandler`.

`services`

Liste, séparées par des espaces, d'une ou plusieurs applications à démarrer.

Comportement du dispositif

Les applications spécifiées commencent à s'exécuter sur l'appareil.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script `shellstart-services.sh`, que le gestionnaire de tâches doit exécuter pour démarrer les services système. Il affiche également le paramètre `requiservices`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Start-Application",  
        "type": "runHandler",  
        "input": {  
          "handler": "start-services.sh",  
          "args": [  
            "${aws:iot:parameter:services}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Arrêter l'application

Nom du modèle

[AWS-Stop-Application](#)

Description du modèle

Modèle géré fourni par AWS pour arrêter un ou plusieurs services.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant, `services`. Vous pouvez également préciser les paramètres suivants `runAsUser` et `pathToHandler`.

`services`

Liste, séparées par des espaces, d'une ou plusieurs applications à arrêter.

Comportement du dispositif

Les applications spécifiées cessent de s'exécuter sur l'appareil.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script `stop-services.sh`, que le gestionnaire de tâches doit exécuter pour arrêter les services système. Il affiche également le paramètre `requiservices`.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Stop-Application",  
        "type": "runHandler",  
        "input": {  
          "handler": "stop-services.sh",  
          "args": [  
            "${aws:iot:parameter:services}"  
          ],  
          "path": "${aws:iot:parameter:pathToHandler}"  
        },  
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
      }  
    }  
  ]  
}
```

AWS—Exécuter—Commande

Nom du modèle

AWS-Run-Command

Description du modèle

Un modèle géré fourni par AWS pour exécuter une commande shell.

Paramètres d'entrée

Ce modèle possède le paramètre obligatoire suivant, `command`. Vous pouvez également préciser le paramètre `requisrunAsUser`.

`command`

Chaîne de commande séparée par des virgules. Toute virgule contenue dans la commande elle-même doit être supprimée.

Comportement du dispositif

Le périphérique exécute la commande shell comme indiqué dans le document de travail.

Document de Job

Ce qui suit montre le document de travail et sa dernière version. Le modèle indique le chemin d'accès à la commande de tâche et à la commande que vous avez fournie et qui sera exécutée par le périphérique.

```
{  
  "version": "1.0",  
  "steps": [  
    {  
      "action": {  
        "name": "Run-Command",  
        "type": "runCommand",  
        "input": {  
          "command": "${aws:iot:parameter:command}"  
        },  
      }  
    }  
  ]  
}
```

```
        "runAsUser": "${aws:iot:parameter:runAsUser}"  
    }  
}  
]
```

Rubriques

- [Créez une tâche à partir de modèles AWS gérés à l'aide du AWS Management Console \(p. 768\)](#)
- [Créez une tâche à partir de modèles AWS gérés à l'aide du AWS CLI \(p. 770\)](#)

Créez une tâche à partir de modèles AWS gérés à l'aide du AWS Management Console

Utilisez le AWS Management Console pour obtenir des informations sur les modèles AWS gérés et créer une tâche à l'aide de ces modèles. Vous pouvez ensuite enregistrer la tâche que vous créez en tant que modèle personnalisé.

Obtenir des détails sur les modèles gérés

Vous pouvez obtenir des informations sur les différents modèles gérés pouvant être utilisés à partir de la AWS IoT console.

1. Pour voir vos modèles gérés disponibles, accédez au [hub de modèles de Job de la AWS IoT console](#) et choisissez l'onglet Modèles gérés.
2. Pour afficher les détails, choisissez un modèle géré.

La page de détails contient les informations suivantes :

- Nom, description et Amazon Resource Name (ARN) du modèle géré.
- Environnement dans lequel les opérations à distance peuvent être effectuées, tel que Linux.
- Le document de travail JSON qui spécifie le chemin d'accès au gestionnaire de tâches et les commandes à exécuter sur le périphérique. Par exemple, ce qui suit montre un exemple de document de travail pour le modèle de redémarrage d'AWS. Le modèle indique le chemin d'accès au gestionnaire de tâches et au script shell que le gestionnaire de tâches doit exécuter pour redémarrer l'appareil. `reboot.sh`

```
{  
    "version": "1.0",  
    "steps": [  
        {  
            "action": {  
                "name": "Reboot",  
                "type": "runHandler",  
                "input": {  
                    "handler": "reboot.sh",  
                    "path": "${aws:iot:parameter:pathToHandler}"  
                },  
                "runAsUser": "${aws:iot:parameter:runAsUser}"  
            }  
        ]  
    ]  
}
```

Pour plus d'informations sur le document de travail et ses paramètres pour les différentes actions à distance, reportez-vous à la section[Modèles gérés d'actions à distance et de documents de travail \(p. 761\)](#).

- La dernière version du document de travail.

Création d'une tâche à l'aide de modèles gérés

Vous pouvez utiliser la console AWS de gestion pour choisir un modèle AWS géré à utiliser pour créer une tâche. Cette section vous montre comment le faire.

Vous pouvez également démarrer le flux de travail de création de tâches, puis choisir le modèle AWS géré que vous souhaitez utiliser lors de la création de la tâche. Pour plus d'informations sur ce flux de travail, consultez [Créez et gérez des tâches à l'aide du AWS Management Console \(p. 749\)](#).

1. Choisissez votre modèle AWS géré

Accédez au [hub de modèles de Job de la AWS IoT console](#), choisissez l'onglet Modèles gérés, puis choisissez votre modèle.

2. Créez une tâche à l'aide de votre modèle géré

1. Sur la page de détails de votre modèle, choisissez Create job.

La console passe à l'étape Propriétés personnalisées des tâches du flux de travail Créer des tâches où la configuration de votre modèle a été ajoutée.

2. Entrez un nom de tâche alphanumérique unique, ainsi qu'une description et des balises facultatives, puis choisissez Suivant.

3. Choisissez les objets ou les groupes d'objets comme cibles de tâche que vous souhaitez exécuter dans cette tâche.

4. Dans la section Document de Job, votre modèle s'affiche avec ses paramètres de configuration et ses paramètres de saisie. Entrez des valeurs pour les paramètres d'entrée du modèle que vous avez choisi. Par exemple, si vous avez choisi le modèle AWS-Download-File :

- Pour DownloadURL, entrez l'URL du fichier à télécharger, par exemple :<https://example.com/index.html>.
- Pour FilePath, entrez le chemin sur l'appareil pour stocker le fichier téléchargé, par exemple : [path/to/file](#)

Vous pouvez également saisir des valeurs pour les paramètres `runAsUser` et `pathToHandler`. Pour plus d'informations sur les paramètres d'entrée de chaque modèle, consultez [Modèles gérés d'actions à distance et de documents de travail \(p. 761\)](#).

5. Sur la page Configuration de la Job, choisissez le type de tâche : tâche continue ou tâche instantanée. Une tâche de capture instantanée est terminée lorsqu'elle termine son exécution sur les équipements et les groupes cibles. Une tâche continue s'applique à des groupes d'objets et s'exécute sur n'importe quel appareil que vous ajoutez à un groupe cible spécifique.

6. Continuez à ajouter des configurations supplémentaires pour votre tâche, puis passez en revue et créez votre tâche. Pour plus d'informations sur les configurations supplémentaires, voir :

- [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
- [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Création de modèles de tâches personnalisés à partir de modèles gérés

Vous pouvez utiliser un modèle AWS géré et une tâche personnalisée comme point de départ pour créer votre propre modèle de tâche personnalisé. Pour créer un modèle de travail personnalisé, commencez par créer un travail à partir de votre modèle AWS géré, comme décrit dans la section précédente.

Vous pouvez ensuite enregistrer la tâche personnalisée en tant que modèle pour créer votre propre modèle de tâche personnalisé. Pour enregistrer en tant que modèle :

1. Accédez au [hub de Job de la AWS IoT console](#) et choisissez la tâche contenant votre modèle géré.

- Choisissez Enregistrer en tant que modèle de travail, puis créez votre modèle de travail personnalisé. Pour plus d'informations sur la création d'un modèle de tâche personnalisé, consultez [Créer un modèle de tâche existant \(p. 774\)](#).

Créez une tâche à partir de modèles AWS gérés à l'aide du AWS CLI

Utilisez le AWS CLI pour obtenir des informations sur les modèles AWS gérés et créer une tâche à l'aide de ces modèles. Vous pouvez ensuite enregistrer la tâche en tant que modèle, puis créer votre propre modèle personnalisé.

Répertorier les modèles gérés

La [list-managed-job-templates](#) AWS CLI commande répertorie tous les modèles de tâches de votre Compte AWS.

```
aws iot list-managed-job-templates
```

Par défaut, l'exécution de cette commande affiche tous les modèles AWS gérés disponibles et leurs détails.

```
{
    "managedJobTemplates": [
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Reboot:1.0",
            "templateName": "AWS-Reboot",
            "description": "A managed job template for rebooting the device.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        },
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Remove-Application:1.0",
            "templateName": "AWS-Remove-Application",
            "description": "A managed job template for uninstalling one or more applications.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        },
        {
            "templateArn": "arn:aws:iot:region::jobtemplate/AWS-Stop-Application:1.0",
            "templateName": "AWS-Stop-Application",
            "description": "A managed job template for stopping one or more system services.",
            "environments": [
                "LINUX"
            ],
            "templateVersion": "1.0"
        },
        ...
        {
            "templateArn": "arn:aws:iot:us-east-1::jobtemplate/AWS-Restart-
Application:1.0",
            "templateName": "AWS-Restart-Application",
        }
    ]
}
```

```
        "description": "A managed job template for restarting one or more system
services.",
        "environments": [
            "LINUX"
        ],
        "templateVersion": "1.0"
    ]
}
```

Pour plus d'informations, reportez-vous à la section [ListManagedJobTemplates](#).

Obtenir des détails sur un modèle géré

La [describe-managed-job-template](#) AWS CLI commande permet d'obtenir des détails sur un modèle de tâche spécifié. Spécifiez le nom du modèle de tâche et une version facultative du modèle. Si la version du modèle n'est pas spécifiée, la version par défaut prédéfinie est renvoyée. Voici un exemple d'exécution de la commande pour obtenir des détails sur le AWS-Download-File modèle.

```
aws iot describe-managed-job-template \
--template-name AWS-Download-File
```

La commande affiche les détails du modèle et l'ARN, son document de travail et le `documentParameters` paramètre, qui est une liste de paires clé-valeur de paramètres d'entrée du modèle. Pour plus d'informations sur les différents modèles et paramètres d'entrée, consultez [Modèles gérés d'actions à distance et de documents de travail \(p. 761\)](#).

Note

L'`documentParameters` objet renvoyé lorsque vous utilisez cette API ne doit être utilisé que lors de la création de tâches à partir de modèles AWS gérés. L'objet ne doit pas être utilisé pour des modèles de tâches personnalisés. Pour obtenir un exemple pratique illustrant l'utilisation de ce paramètre, consultez [Création d'une tâche à l'aide de modèles gérés \(p. 772\)](#).

```
{
    "templateName": "AWS-Download-File",
    "templateArn": "arn:aws:iot:region:jobtemplate/AWS-Download-File:1.0",
    "description": "A managed job template for downloading a file.",
    "templateVersion": "1.0",
    "environments": [
        "LINUX"
    ],
    "documentParameters": [
        {
            "key": "downloadUrl",
            "description": "URL of file to download.",
            "regex": "(.*?)",
            "example": "http://www.example.com/index.html",
            "optional": false
        },
        {
            "key": "filePath",
            "description": "Path on the device where downloaded file is written.",
            "regex": "(.*?)",
            "example": "/path/to/file",
            "optional": false
        },
        {
            "key": "runAsUser",
            "description": "Execute handler as another user. If not specified, then handler
is executed as the same user as device client."
        }
    ]
}
```

```

        "regex": "(.){0,256}",
        "example": "user1",
        "optional": true
    },
    {
        "key": "pathToHandler",
        "description": "Path to handler on the device. If not specified, then device client will use the current working directory.",
        "regex": "(.){0,4096}",
        "example": "/path/to/handler/script",
        "optional": true
    }
],
"document": "{\"version\":\"1.0\",\"steps\":[{\"action\":{\"name\":\"Download-File\",\"type\":\"runHandler\",\"input\":{\"handler\":\"download-file.sh\",\"args\":[$aws:iot:parameter:downloadUrl,$aws:iot:parameter:filePath],\"path\":\"$aws:iot:parameter:pathToHandler\"},\"runAsUser\":$aws:iot:parameter:runAsUser}}]}"
}

```

Pour plus d'informations, reportez-vous à la section [DescribeManagedJobTemplate](#).

Création d'une tâche à l'aide de modèles gérés

La [create-job](#) AWS CLI commande peut être utilisée pour créer une tâche à partir d'un modèle de tâche. Il cible un appareil nommé `thingOne` et spécifie l'Amazon Resource Name (ARN) du modèle géré à utiliser comme base de la tâche. Vous pouvez remplacer les configurations avancées, telles que les configurations de temporisation et d'annulation, en transmettant les paramètres associés de la `create-job` commande.

L'exemple montre comment créer une tâche qui utilise le `AWS-Download-File` modèle. Il montre également comment spécifier les paramètres d'entrée du modèle à l'aide du `document-parameters` paramètre.

Note

Utilisez l'`document-parameters` objet uniquement avec des modèles AWS gérés. Cet objet ne doit pas être utilisé avec des modèles de tâches personnalisés.

```
aws iot create-job \
--targets arn:aws:iot:region:account-id:thing/thingOne \
--job-id "new-managed-template-job" \
--job-template-arn arn:aws:iot:region::jobtemplate/AWS-Download-File:1.0 \
--document-parameters downloadUrl=https://example.com/index.html,filePath=path/to/file
```

où :

- **la région** est la Région AWS.
- **account-id** est le numéro unique Compte AWS.
- **thingOne** est le nom de l'objet IoT pour lequel le poste est destiné.
- **AWS-Download-File:1.0** est le nom du modèle géré.
- **https://example.com/index.html** est l'URL à partir de laquelle télécharger le fichier.
- **https://path/to/file/index** est le chemin de stockage du fichier téléchargé sur l'appareil.

Exécutez la commande suivante pour créer une tâche pour le modèle `AWS-Download-File`.

```
"jobArn": "arn:aws:iot:region:account-id:job/new-managed-template-job",  
"jobId": "new-managed-template-job",  
"description": "A managed job template for downloading a file."  
}
```

Création d'un modèle de travail personnalisé à partir de modèles gérés

1. Créez une tâche à l'aide d'un modèle géré comme indiqué dans la section précédente.
2. Créez un modèle de tâche personnalisé à l'aide de l'ARN de la tâche que vous avez créée. Pour plus d'informations, veuillez consulter [Créer un modèle de tâche existant \(p. 776\)](#).

Créez des modèles de travail personnalisés

Vous pouvez créer des modèles de tâches à l'aide de la AWS IoT console AWS CLI et. Vous pouvez également créer des tâches à partir de modèles de tâches en utilisant la AWS CLI AWS IoT console et les applications Web Fleet Hub for AWS IoT Device Management. Pour plus d'informations sur l'utilisation de modèles de tâches dans les applications Fleet Hub, consultez la section [Utilisation de modèles de tâches dans Fleet Hub pour la gestion des AWS IoT appareils](#).

Note

Le nombre total de modèles de substitution dans un document de tâche doit être inférieur ou égal à dix.

Rubriques

- [Créez des modèles de tâches personnalisés à l'aide du AWS Management Console \(p. 773\)](#)
- [Créez des modèles de tâches personnalisés à l'aide du AWS CLI \(p. 776\)](#)

Créez des modèles de tâches personnalisés à l'aide du AWS Management Console

Cette rubrique explique comment créer, supprimer et afficher les détails des modèles de tâches à l'aide de la AWS IoT console.

Création d'un modèle de travail personnalisé

Vous pouvez créer un modèle de travail personnalisé original ou créer un modèle de travail à partir d'un travail existant. Vous pouvez également créer un modèle de travail personnalisé à partir d'un travail existant créé à l'aide d'un modèle AWS géré. Pour plus d'informations, veuillez consulter [Création de modèles de tâches personnalisés à partir de modèles gérés \(p. 769\)](#).

Création d'un modèle de travail original

1. Commencez à créer votre modèle de travail
 1. Accédez au [hub de modèles de Job de la AWS IoT console](#) et choisissez l'onglet Modèles personnalisés.
 2. Choisissez Créer un modèle de tâche.

Note

Vous pouvez également accéder à la page des modèles de Job à partir de la page Services associés sous Fleet Hub.

2. Spécifier les propriétés du modèle de tâche

Sur la page Créer un modèle de tâche, entrez un identifiant alphanumérique pour le nom de votre tâche et une description alphanumérique afin de fournir des informations supplémentaires sur le modèle.

Note

Nous vous déconseillons d'utiliser des informations personnelles identifiables dans vos identifiants de poste ou vos descriptions de poste.

3. Fournir un document de travail

Fournissez un fichier de travail JSON stocké dans un compartiment S3 ou sous la forme d'un document de travail en ligne spécifié dans la tâche. Ce fichier de travail deviendra le document de travail lorsque vous créerez un travail à l'aide de ce modèle.

Si le fichier de travail est stocké dans un compartiment S3, entrez l'URL S3 ou choisissez Parcourir S3, puis accédez à votre document de travail et sélectionnez-le.

Note

Vous ne pouvez sélectionner que des compartiments S3 dans votre région actuelle.

4. Continuez à ajouter des configurations supplémentaires pour votre tâche, puis passez en revue et créez votre tâche. Pour plus d'informations sur les configurations suivantes :

- [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
- [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Créer un modèle de tâche existant

1. Choisissez votre travail

1. Accédez au [hub de Job de la AWS IoT console](#) et choisissez la tâche que vous souhaitez utiliser comme base pour votre modèle de tâche.
2. Choisissez Enregistrer en tant que modèle de tâche.

Note

Vous pouvez éventuellement choisir un autre document de travail ou modifier les configurations avancées par rapport à la tâche d'origine, puis choisir Créer un modèle de tâche. Votre nouveau modèle de Job apparaît sur la page Modèles de tâches.

2. Spécifier les propriétés du modèle de tâche

Sur la page Créer un modèle de tâche, entrez un identifiant alphanumérique pour le nom de votre tâche et une description alphanumérique afin de fournir des informations supplémentaires sur le modèle.

Note

Le document de travail est le fichier de travail que vous avez spécifié lors de la création du modèle. Si le document de travail est spécifié dans le travail plutôt que dans un emplacement S3, vous pouvez voir le document de travail sur la page de détails de ce travail.

3. Continuez à ajouter des configurations supplémentaires pour votre tâche, puis passez en revue et créez votre tâche. Pour plus d'informations sur les configurations supplémentaires, voir :

- [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
- [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Création d'une tâche à partir d'un modèle de tâche personnalisé

Vous pouvez créer un travail à partir d'un modèle de travail personnalisé en accédant à la page de détails de votre modèle de travail, comme décrit dans cette rubrique. Vous pouvez également créer une tâche ou choisir le modèle de tâche que vous souhaitez utiliser lors de l'exécution du flux de travail de création de tâches. Pour plus d'informations, veuillez consulter [Créez et gérez des tâches à l'aide du AWS Management Console \(p. 749\)](#).

Cette rubrique explique comment créer une tâche à partir de la page de détails d'un modèle de tâche personnalisé. Vous pouvez également créer une tâche à partir d'un modèle AWS géré. Pour plus d'informations, veuillez consulter [Création d'une tâche à l'aide de modèles gérés \(p. 769\)](#).

1. Choisissez votre modèle de tâche personnalisé

Accédez au [hub de modèles de Job de la AWS IoT console](#) et choisissez l'onglet Modèles personnalisés, puis choisissez votre modèle.

2. Créez une tâche à l'aide de votre modèle personnalisé

Pour créer une tâche :

1. Sur la page de détails de votre modèle, choisissez Create job.

La console passe à l'étape Propriétés personnalisées des tâches du flux de travail Créer des tâches où la configuration de votre modèle a été ajoutée.

2. Entrez un nom de tâche alphanumérique unique, ainsi qu'une description et des balises facultatives, puis choisissez Suivant.
3. Choisissez les objets ou les groupes d'objets comme cibles de tâche que vous souhaitez exécuter dans cette tâche.

Dans la section Document de Job, votre modèle s'affiche avec ses paramètres de configuration. Si vous souhaitez utiliser un autre document de travail, choisissez Parcourir et sélectionnez un bucket et un document différents. Choisissez Suivant.

4. Sur la page Configuration de la Job, choisissez le type de tâche : tâche continue ou tâche instantanée. Une tâche de capture instantanée est terminée lorsqu'elle termine son exécution sur les équipements et les groupes cibles. Une tâche continue s'applique à des groupes d'objets et s'exécute sur n'importe quel appareil que vous ajoutez à un groupe cible spécifique.
5. Continuez à ajouter des configurations supplémentaires pour votre tâche, puis passez en revue et créez votre tâche. Pour plus d'informations sur les configurations supplémentaires, voir :
 - [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
 - [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Note

Lorsqu'une tâche créée à partir d'un modèle de tâche met à jour les paramètres existants fournis par le modèle de tâche, ces paramètres mis à jour remplacent les paramètres existants fournis par le modèle de tâche pour cette tâche.

Vous pouvez également créer des tâches à partir de modèles de tâches avec les applications Web Fleet Hub. Pour plus d'informations sur la création de tâches dans Fleet Hub, consultez la section [Utilisation de modèles de tâches dans Fleet Hub pour la gestion des AWS IoT appareils](#).

Supprimer un modèle de tâche

Pour supprimer un modèle de Job, accédez d'abord au [hub Modèles de tâches de la AWS IoT console](#) et choisissez l'onglet Modèles personnalisés. Choisissez ensuite le modèle de travail que vous souhaitez supprimer, puis cliquez sur Suivant.

Note

Une suppression est définitive et le modèle de tâche n'apparaît plus dans l'onglet Modèles personnalisés.

Créez des modèles de tâches personnalisés à l'aide du AWS CLI

Cette rubrique explique comment créer, supprimer et récupérer des détails sur les modèles de tâches à l'aide du AWS CLI.

Créer un modèle de tâche à partir de zéro

La AWS CLI commande suivante montre comment créer une tâche à l'aide d'un document de travail (job-document.json) stocké dans un compartiment S3 et d'un rôle autorisé à télécharger des fichiers depuis Amazon S3 (S3). DownloadRole

```
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 `timeout-config` paramètre suivant indique le temps dont dispose chaque appareil pour terminer d'exécuter la tâche. Le minuteur est démarré quand l'état de l'exécution de la tâche a la valeur `IN_PROGRESS`. Si l'état d'exécution de la tâche n'est pas défini sur un autre état du terminal avant l'expiration du délai, il est défini sur `TIMED_OUT`.

Le chronomètre en cours ne peut pas être mis à jour et s'applique à tous les lancements de tâches pour la tâche. Chaque fois qu'un lancement de la `IN_PROGRESS` tâche demeure en cours plus longtemps que l'intervalle défini, le lancement de la tâche échoue et passe à l'`TIMED_OUT` état terminal. AWS IoT publie également une notification MQTT.

Pour plus d'informations sur la création de configurations concernant les déploiements et les abandons de tâches, voir Configuration du [déploiement et de l'abandon des tâches](#).

Note

Les documents de Job spécifiés sous forme de fichiers Amazon S3 sont récupérés au moment où vous créez le travail. Si vous modifiez le contenu du fichier Amazon S3 que vous avez utilisé comme source de votre document de travail après avoir créé le travail, ce qui est envoyé aux cibles du travail ne change pas.

Créer un modèle de tâche existant

La AWS CLI commande suivante crée 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 la tâche. Vous pouvez éventuellement modifier n'importe quelle configuration de la tâche existante 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 détails sur un modèle de tâche

La AWS CLI commande suivante permet d'obtenir des détails sur un modèle de tâche spécifié.

```
aws iot describe-job-template \
--job-template-id <template-id>
```

La commande affiche le résultat suivant.

```
{
  "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
  }
}
```

Répertorier les modèles de tâche

La AWS CLI commande suivante répertorie tous les modèles de tâches de votreCompte AWS.

```
aws iot list-job-templates
```

La commande affiche le résultat suivant.

```
{  
    "jobTemplates": [  
        {  
            "createdAt": number,  
            "description": "string",  
            "jobTemplateArn": "string",  
            "jobTemplateId": "string"  
        }  
    ],  
    "nextToken": "string"  
}
```

Pour récupérer des pages de résultats supplémentaires, utilisez la valeur du nextToken champ.

Supprimer un modèle de tâche

La AWS CLI commande suivante supprime un modèle de tâche spécifié.

```
aws iot delete-job-template \  
    --job-template-id template-id
```

La commande n'affiche aucune sortie.

Création d'une tâche à partir d'un modèle de tâche personnalisé

La AWS CLI commande suivante crée une tâche à partir d'un modèle de tâche personnalisé. Il cible un appareil nommé thing0ne et spécifie l'Amazon Resource Name (ARN) du modèle de tâche à utiliser comme base de la tâche. Vous pouvez remplacer les configurations avancées, telles que les configurations de temporisation et d'annulation, en transmettant les paramètres associés de la create-job commande.

Warning

L'`document-parameters` objet doit être utilisé avec la `create-job` commande uniquement lors de la création de tâches à partir de modèles AWS gérés. Cet objet ne doit pas être utilisé avec des modèles de tâches personnalisés. Pour obtenir un exemple pratique illustrant la création de tâche à l'aide de ce paramètre, consultez [Création d'une tâche à l'aide de modèles gérés \(p. 772\)](#).

```
aws iot create-job \  
    --targets arn:aws:iot:region:123456789012:thing/thingOne \  
    --job-template-arn arn:aws:iot:region:123456789012:jobtemplate/template-id
```

Configurations de Job

Vous pouvez disposer des configurations supplémentaires suivantes pour chaque tâche que vous déployez sur les cibles spécifiées.

- Déploiement : définit le nombre d'appareils qui reçoivent le document de travail chaque minute.

- Planification : planifie une tâche pour une date et une heure future en plus d'utiliser des fenêtres de maintenance récurrentes.
- Abandonner : annule une tâche lorsque certains appareils ne reçoivent pas la notification de tâche ou lorsque vos appareils signalent un échec lors de l'exécution de leur tâche.
- Délai d'expiration : si vos cibles de tâches ne répondent pas dans un certain délai après le début de leur exécution, la tâche peut échouer.
- Réessayer : Réessaie d'exécuter la tâche si votre appareil signale un échec lors de la tentative d'exécution d'une tâche ou si le délai d'exécution de votre tâche expire.

À l'aide de ces configurations, vous pouvez surveiller l'état d'exécution de vos tâches et éviter qu'une mauvaise mise à jour ne soit envoyée à l'ensemble d'un parc.

Rubriques

- [Comment fonctionnent les configurations de tâches \(p. 779\)](#)
- [Spécifiez des configurations supplémentaires \(p. 789\)](#)

Comment fonctionnent les configurations de tâches

Vous utilisez les configurations de déploiement et d'abandon lorsque vous déployez une tâche, ainsi que les configurations de délai d'expiration et de nouvelle tentative pour l'exécution de la tâche. Les sections suivantes fournissent des informations supplémentaires sur le fonctionnement de ces configurations.

Rubriques

- [Configurations de déploiement, de planification et d'abandon des Job \(p. 779\)](#)
- [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Configurations de déploiement, de planification et d'abandon des Job

Vous pouvez utiliser les configurations de déploiement, de planification et d'abandon des tâches pour définir le nombre d'appareils qui reçoivent le document de travail, planifier le déploiement d'une tâche et déterminer les critères d'annulation d'une tâche lorsqu'un certain nombre d'appareils ne reçoivent pas de document de tâche.

Configuration du déploiement des Job

Vous pouvez spécifier la vitesse à laquelle les cibles sont averties d'une exécution de tâche en attente. Vous pouvez également créer un déploiement échelonné pour gérer les mises à jour, les redémarrages et d'autres opérations. Pour spécifier la manière dont vos cibles sont notifiées, utilisez les taux de déploiement des tâches.

Taux de déploiement des Job

Vous pouvez créer une configuration de déploiement en utilisant un taux de déploiement constant ou un taux de déploiement exponentiel. Pour spécifier le nombre maximum de cibles de tâches à informer par minute, utilisez un taux de déploiement constant.

AWS IoT les tâches peuvent être déployées en utilisant des taux de déploiement exponentiels à mesure que divers critères et seuils sont atteints. Si le nombre de tâches ayant échoué correspond à un ensemble de critères que vous avez spécifiés, vous pouvez annuler le déploiement de la tâche. Vous définissez les critères de taux de déploiement des tâches lorsque vous créez une tâche à l'aide de l'[`JobExecutionsRolloutConfig`](#) objet. Vous définissez également les critères d'abandon de la tâche lors de la création de la tâche à l'aide de l'[`AbortConfig`](#) objet.

L'exemple suivant présente le fonctionnement des taux de déploiement Par exemple, un déploiement de tâches avec une fréquence de base de 50 par minute, un facteur d'incrémentation de 2 et un nombre de 1 000 appareils ayant reçu une notification et ayant réussi chacun fonctionnerait comme suit : la tâche démarra à un rythme de 50 exécutions de tâches par minute et se poursuivra à ce rythme jusqu'à ce que 1 000 objets aient reçu des notifications d'exécution de tâches ou que 1 000 exécutions de tâches aient eu lieu avec succès.

Le tableau suivant illustre la façon dont le déploiement procéderait sur les quatre premiers incrémentations.

Fréquence de lancement par minute	50	100	200	400
Nombre d'appareils notifiés ou d'exécutions de tâches réussies pour répondre à une augmentation du taux	1 000	2 000	3 000	4 000

Note

Si vous atteignez votre limite maximale de 500 tâches simultanées (`isConcurrent = True`), toutes les tâches actives conserveront le statut de `IN-PROGRESS` et ne seront exécutées aucune nouvelle tâche tant que le nombre de tâches simultanées ne sera pas inférieur ou égal à 499 (`isConcurrent = False`). Cela s'applique aux tâches instantanées et continues.

Si `isConcurrent = True`, la tâche est en train de déployer des exécutions de tâches sur tous les appareils de votre groupe cible. Si `isConcurrent = False` la tâche a terminé le déploiement de toutes les exécutions de tâches sur tous les appareils de votre groupe cible. Il mettra à jour son état une fois que tous les appareils de votre groupe cible atteindront un état terminal, ou un certain pourcentage de votre groupe cible si vous avez sélectionné une configuration d'abandon de tâche. Le statut du niveau du Job indique pour `isConcurrent = True` et `isConcurrent = False` sont les deux `IN_PROGRESS`.

Pour plus d'informations sur les limites de tâches actives et simultanées, consultez [Limites de tâches actives et simultanées \(p. 845\)](#).

Taux de déploiement des Job pour les tâches continues utilisant des groupes d'objets dynamiques

Lorsque vous utilisez une tâche continue pour déployer des opérations à distance sur votre parc, AWS IoT Jobs exécute des tâches pour les appareils de votre groupe cible. Pour les nouveaux appareils ajoutés au groupe d'objets dynamiques, ces exécutions de tâches continuent d'être déployées sur ces appareils même après la création de la tâche.

La configuration de déploiement peut contrôler les taux de déploiement uniquement pour les appareils ajoutés au groupe jusqu'à la création de la tâche. Une fois qu'une tâche a été créée, pour tout nouvel appareil, les exécutions de tâches sont créées quasiment en temps réel dès que les appareils rejoignent le groupe cible.

Configuration de la planification des Job

Vous pouvez planifier une tâche continue ou instantanée jusqu'à un an à l'avance en utilisant une heure de début, une heure de fin et un comportement de fin prédéterminés pour ce qui arrivera à l'exécution de chaque tâche une fois l'heure de fin atteinte. En outre, vous pouvez créer une fenêtre de maintenance récurrente facultative avec une fréquence, une heure de début et une durée flexibles pour les tâches continues afin de déployer un document de travail sur tous les appareils du groupe cible.

Configurations de planification des Job

Heure de début

L'heure de début d'une tâche planifiée est la date et l'heure future à laquelle cette tâche commencera à être déployée du document de travail sur tous les appareils du groupe cible. L'heure de début d'une tâche planifiée s'applique aux tâches continues et aux tâches instantanées. Lorsqu'une tâche planifiée

est initialement créée, elle conserve un état de statut deSCHEDULED. Lorsque vous arrivez à startTime ce que vous avez sélectionné, le document de travail est mis à jour IN_PROGRESS et commence à être déployé. Le délai startTime doit être inférieur ou égal à un an à compter de la date et de l'heure initiales de création de la tâche planifiée.

Pour une tâche avec la configuration de planification optionnelle qui a lieu pendant une fenêtre de maintenance récurrente dans un lieu respectant l'heure d'été (DST), l'heure changera d'une heure lors du passage de l'heure d'été à l'heure normale et de l'heure normale à l'heure d'été.

Note

Le fuseau horaire affiché dans le AWS Management Console est le fuseau horaire actuel de votre système. Toutefois, ces fuseaux horaires seront convertis en UTC dans le système.

Heure de fin

L'heure de fin d'une tâche planifiée est la date et l'heure future auxquelles la tâche arrêtera le déploiement du document de travail sur tous les appareils restants du groupe cible. L'heure de fin d'une tâche planifiée s'applique aux tâches continues et aux tâches instantanées. Lorsqu'une tâche planifiée arrive à l'état sélectionné endTime et que toutes les exécutions de tâches ont atteint un état terminal, elle met à jour son état de statut de IN_PROGRESS àCOMPLETED. Le délai endTime doit être inférieur ou égal à deux ans à compter de la date et de l'heure initiales de création de la tâche planifiée. La durée minimale entre startTime et endTime est de 30 minutes. Les nouvelles tentatives d'exécution de la Job se produiront jusqu'à ce que la tâche atteigne leendTime, puis elles endBehavior dicteront la marche à suivre.

Pour une tâche avec la configuration de planification optionnelle qui a lieu pendant une fenêtre de maintenance récurrente dans un lieu respectant l'heure d'été (DST), l'heure changera d'une heure lors du passage de l'heure d'été à l'heure normale et de l'heure normale à l'heure d'été.

Note

Le fuseau horaire affiché dans le AWS Management Console est le fuseau horaire actuel de votre système. Toutefois, ces fuseaux horaires seront convertis en UTC dans le système.

Comportement final

Le comportement final d'une tâche planifiée détermine ce qu'il advient de la tâche et de toutes les exécutions de tâches inachevées lorsque la tâche atteint la valeur sélectionnéeendTime.

La liste suivante répertorie les comportements finaux parmi lesquels vous pouvez sélectionner lors de la création de la tâche ou du modèle de tâche :

- STOP_ROLLOUT
 - STOP_ROLLOUTarrête le déploiement du document de travail sur tous les appareils restants du groupe cible de la tâche. En outre, toutes QUEUED les exécutions de IN_PROGRESS tâches se poursuivront jusqu'à ce qu'elles atteignent un état terminal. Il s'agit du comportement final par défaut, sauf si vous sélectionnez CANCEL ouFORCE_CANCEL.
- CANCEL
 - CANCELarrête le déploiement du document de travail sur tous les appareils restants du groupe cible de la tâche. En outre, toutes les exécutions de QUEUED tâches seront annulées tandis que toutes les exécutions de IN_PROGRESS tâches se poursuivront jusqu'à ce qu'elles atteignent un état terminal.
- FORCE_CANCEL
 - FORCE_CANCELarrête le déploiement du document de travail sur tous les appareils restants du groupe cible de la tâche. De plus, toutes QUEUED les exécutions de IN_PROGRESS tâches seront annulées.

Note

Si vous ne sélectionnez pas deendTime, vous ne pourrez pas sélectionner deendBehavior.

Durée maximale

La durée maximale d'un emploi prévu doit être inférieure ou égale à deux ans quel que soit le `startTime` et `endTime`.

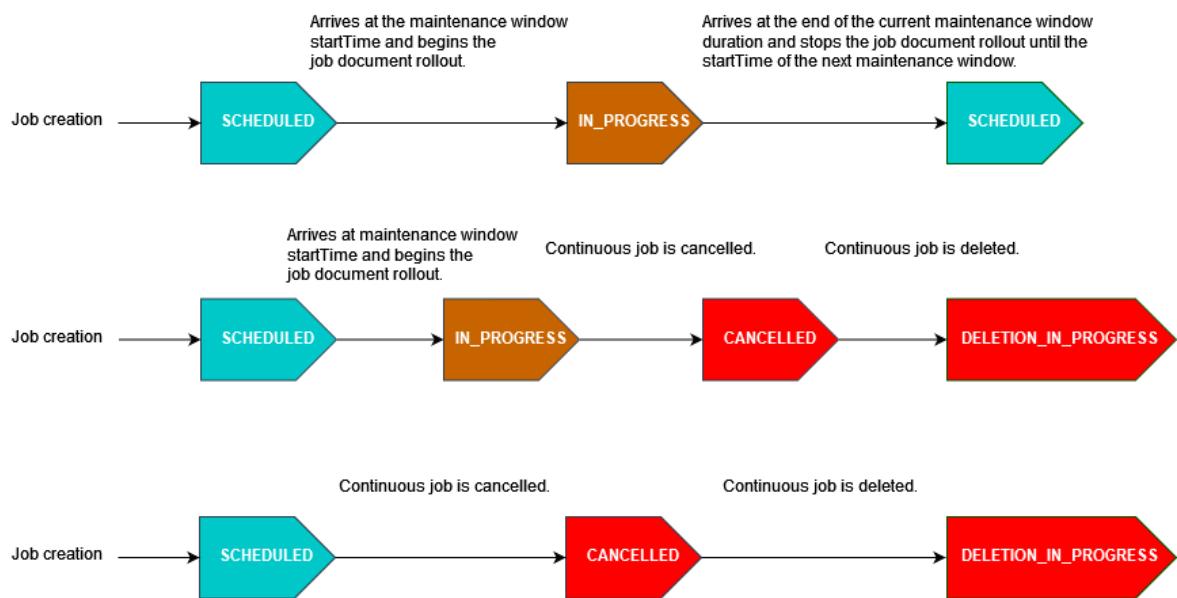
Le tableau suivant répertorie les scénarios de durée courants d'une tâche planifiée :

Numéro d'exemple de tâche planifiée	<code>startTime</code>	<code>endTime</code>	Durée maximale
1	Immédiatement après la création initiale de l'emploi.	Un an après la création initiale de l'emploi.	Un an
2	Un mois après la création initiale de l'emploi.	13 mois après la création initiale de l'emploi.	Un an
3	Un an après la création initiale de l'emploi.	Deux ans après la création initiale d'emplois.	Un an
4	Immédiatement après la création initiale de l'emploi.	Deux ans après la création initiale d'emplois.	Deux ans

Le créneau de maintenance périodique

La fenêtre de maintenance est une configuration facultative intégrée à la AWS Management Console configuration de planification `SchedulingConfig` des `CreateJobTemplate` API `CreateJob` et `StartJob`. Vous pouvez configurer une fenêtre de maintenance récurrente avec une heure de début, une durée et une fréquence (quotidienne, hebdomadaire ou mensuelle) prédéfinies à laquelle la fenêtre de maintenance se produit. Les fenêtres de maintenance ne s'appliquent qu'aux tâches continues. La durée maximale d'une fenêtre de maintenance périodique est de 23 heures et 50 minutes.

Le schéma suivant illustre les états d'état des tâches pour différents scénarios de tâches planifiées avec une fenêtre de maintenance facultative :



Pour de plus amples informations sur l'état d'une tâche [Tâches et états d'exécution des tâches \(p. 743\)](#)

Note

Si une tâche arrive au endTime cours d'une fenêtre de maintenance, elle sera mise à jour du IN_PROGRESS auCOMPLETED. En outre, toutes les exécutions de tâches restantes suivront le code endBehavior correspondant à la tâche.

Expressions Cron

Pour les tâches planifiées qui déplient le document de travail pendant une fenêtre de maintenance avec une fréquence personnalisée, la fréquence personnalisée est saisie à l'aide d'une expression cron. Une expression cron comporte six champs obligatoires, séparés par des espaces.

Syntaxe

```
cron(fields)
```

Champ	Valeurs	Caractères génériques
Minutes	0-59	, - * /
Heures	0-23	, - * /
D ay-of-month	1-31	, - * ? / L W
Mois	1-12 ou JAN-DEC	, - * /
D ay-of-week	1-7 ou DIM-SAM	, - * ? L #
Année	1970-2199	, - * /

Caractères génériques

- Le caractère générique , (virgule) inclut des valeurs supplémentaires. Dans le champ Mois, JAN,FEB,MAR englobe janvier, février et mars.
- Le caractère générique - (tiret) spécifie des plages. Dans le champ Jour, 1-15 englobe les jours 1 à 15 du mois spécifié.
- Le caractère générique * (astérisque) inclut toutes les valeurs du champ. Dans le champ Hours, * inclut toutes les heures. Vous ne pouvez pas utiliser* à la fois dans les ay-of-week champs D ay-of-month et D. Si vous l'utilisez dans un champ, vous devez utiliser ? dans l'autre.
- Le caractère générique / (barre oblique) spécifie les incrémentations. Dans le champ Minutes, vous pouvez entrer 1/10 pour spécifier toutes les dix minutes, à partir de la première minute de l'heure (par exemple, les 11e, 21e, 31e minutes, et ainsi de suite).
- Le caractère générique ? (point d'interrogation) indique l'un ou l'autre. Dans le ay-of-month champ D, vous pouvez saisir 7 et si vous ne vous souciez pas du jour de la semaine qui tombe le 7, vous pourriez saisir ? dans le ay-of-week champ D.
- Le caractère générique L dans les ay-of-week champs D ay-of-month ou D spécifie le dernier jour de la semaine
- Le W caractère générique dans le ay-of-month champ D indique un jour de la semaine. Le ay-of-month champ D 3W spécifie le jour le plus proche du troisième jour de la semaine le plus proche du troisième jour de la semaine le plus proche du troisième jour de la semaine le plus proche
- Le caractère générique # dans le ay-of-week champ D spécifie une certaine instance du jour de la semaine spécifié dans un mois. Par exemple, 3#2 correspond au deuxième mardi du mois : le 3 fait référence à mardi, car c'est le troisième jour de chaque semaine, et le 2 fait référence à la deuxième journée de ce type dans le mois.

Note

Si vous utilisez le caractère « # », vous ne pouvez définir qu'une seule expression dans le day-of-week champ. Par exemple, "3#1,6#3" n'est pas valide car il est interprété comme deux expressions.

Restrictions

- Vous ne pouvez pas spécifier les ay-of-week champs D ay-of-month et D dans une même expression cron Si vous spécifiez une valeur (ou le caractère *) dans l'un de ces champs, vous devez utiliser un caractère * dans l'autre.

Exemples

Reportez-vous aux exemples de chaînes cron suivants lorsque vous utilisez une expression cron pour une fenêtre startTime de maintenance récurrente.

Minutes	Heures	Jour du mois	Mois	Jour de la semaine	Année	Signification
0 USD	10	*	*	?	*	Exécuter à 10 h 00 (UTC) chaque jour
15	12	*	*	?	*	Exécuter à 12 h 15 (UTC) chaque jour
0	18	?	*	MON-FRI	*	Exécuter à 18 h 00 (UTC) du lundi au vendredi
0	8	1	*	?	*	Exécuter à 8 h 00 (UTC) chaque 1er jour du mois

Logique de fin de la période de maintenance récurrente

Lorsqu'un déploiement de tâche au cours d'une fenêtre de maintenance atteint la fin de la durée d'occurrence de la fenêtre de maintenance en cours, les actions suivantes se produisent :

- Le Job cessera tout déploiement du document de travail sur tous les appareils restants de votre groupe cible. Elle reprendra dès la fenêtre startTime de maintenance suivante.
- Toutes les exécutions de tâches dont le statut est QUEUED resteront actives QUEUED jusqu'à la prochaine occurrence startTime de la fenêtre de maintenance. Dans la fenêtre suivante, ils peuvent passer au IN_PROGRESS moment où l'appareil est prêt à commencer à exécuter les actions spécifiées dans le document de travail.
- Toutes les exécutions de tâches dont le statut est IN_PROGRESS continueront à exécuter les actions spécifiées dans le document de travail jusqu'à ce qu'elles atteignent un état terminal. Toute nouvelle

tentative telle que spécifiée dans `JobExecutionsRetryConfig` aura lieu lors `startTime` de la fenêtre de maintenance suivante.

Configuration de l'abandon d'une Job

Utilisez cette configuration pour créer un critère permettant d'annuler une tâche lorsqu'un certain pourcentage d'appareils répond à ces critères. Par exemple, vous pouvez utiliser cette configuration pour annuler une tâche dans les cas suivants :

- Lorsqu'un certain pourcentage d'appareils ne reçoivent pas les notifications d'exécution des tâches, par exemple lorsque votre appareil n'est pas compatible avec une mise à jour sans fil (OTA). Dans ce cas, votre appareil peut signaler un `REJECTED` état.
- Lorsqu'un certain pourcentage d'appareils signalent un échec lors de l'exécution de leurs tâches, par exemple lorsque votre appareil se déconnecte lorsqu'il tente de télécharger le document de travail à partir d'une URL Amazon S3. Dans ce cas, votre appareil doit être programmé pour signaler l'`FAILURE` état à AWS IoT.
- Lorsqu'un `TIMED_OUT` état est signalé parce que le délai d'exécution du travail expire pour un certain pourcentage d'appareils après le début de l'exécution du travail.
- En cas d'échec de plusieurs tentatives. Lorsque vous ajoutez une configuration de nouvelle tentative, chaque nouvelle tentative peut entraîner des frais supplémentaires pour vous. Compte AWS Dans de tels cas, l'annulation de la tâche peut annuler les exécutions de tâches en file d'attente et éviter de nouvelles tentatives pour ces exécutions. Pour plus d'informations sur la configuration des nouvelles tentatives et son utilisation avec la configuration d'abandon, consultez. [Configurations du délai d'exécution des Job et des nouvelles tentatives \(p. 785\)](#)

Vous pouvez configurer une condition d'abandon de tâche à l'aide de la AWS IoT console ou de l'API AWS IoT Jobs.

Configurations du délai d'exécution des Job et des nouvelles tentatives

Utilisez la configuration du délai d'exécution des tâches pour vous envoyer des [Notifications Jobs \(p. 802\)](#) informations lorsque l'exécution d'une tâche est en cours depuis plus longtemps que la durée définie. Utilisez la configuration de nouvelle tentative d'exécution de la tâche pour réessayer l'exécution lorsque la tâche échoue ou expire.

Configuration du délai d'attente d'exécution de Job

Utilisez la configuration du délai d'exécution des tâches pour vous avertir lorsque l'exécution d'une tâche reste bloquée dans `IN_PROGRESS` cet état pendant une période étonnamment longue. Lorsque la tâche est `IN_PROGRESS` terminée, vous pouvez suivre la progression de son exécution.

Minuteries pour les délais d'expiration des tâches

Il existe deux types de minutiers : minutiers d'avancement et minutiers d'étape.

Minutiers en cours

Lorsque vous créez une tâche ou un modèle de tâche, vous pouvez spécifier une valeur pour le temporisateur en cours comprise entre 1 minute et 7 jours. Vous pouvez mettre à jour la valeur de ce temporisateur jusqu'au début de l'exécution de votre tâche. Une fois que votre chronomètre démarre, il ne peut pas être mis à jour et la valeur du chronomètre s'applique à toutes les exécutions de tâches 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.

Chronomètre

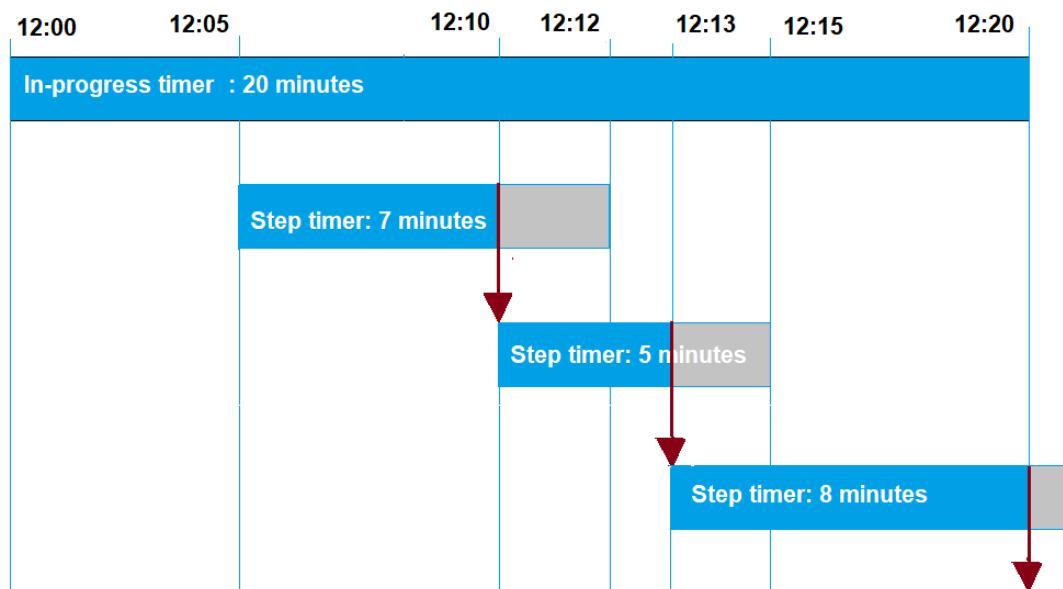
Vous pouvez également définir un minuteur qui s'applique uniquement à l'exécution de la tâche que vous souhaitez mettre à jour. Ce créneau n'a aucun effet sur le chronomètre en cours. Chaque fois que vous mettez à jour l'exécution d'une tâche, vous pouvez définir une nouvelle valeur pour le chronomètre. Vous pouvez également créer un nouveau créneau lorsque vous lancez l'exécution de tâche en attente suivante pour un objet. 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.

Note

Vous pouvez définir le chronomètre en cours à l'aide de la AWS IoT console ou de l'API AWS IoT Jobs. Pour spécifier le chronomètre, utilisez l'API.

Comment fonctionnent les minuteries pour les délais d'expiration des tâches

Ce qui suit illustre la manière dont les délais d'attente en cours et les délais d'expiration d'étape interagissent au cours d'une période de 20 minutes.



Voici les différentes étapes :

1. 12h00

Une nouvelle tâche est créée et un chronomètre en cours de vingt minutes est lancé lors de la création d'une tâche. Le chronomètre en cours commence à s'exécuter et l'exécution de la tâche passe à l'IN_PROGRESS état.

2. 12 H 05

Un nouveau chronomètre d'une valeur de 7 minutes est créé. L'exécution de la tâche expirera désormais à 12 h 12.

3. 12 H 10

Un nouveau chronomètre d'une valeur de 5 minutes est créé. Lorsqu'un nouveau chronomètre est créé, le chronomètre précédent est supprimé et l'exécution de la tâche expirera désormais à 12h15.

4. 12 H 13

Un nouveau chronomètre d'une valeur de 9 minutes est créé. Le chronomètre précédent est supprimé et l'exécution de la tâche expirera désormais à 12 h 20 car le chronomètre en cours expire à 12 h 20. Le chronomètre ne peut pas dépasser la limite absolue du chronomètre en cours.

Configuration de nouvelle tentative d'exécution de Job

Vous pouvez utiliser la configuration de nouvelle tentative pour réessayer d'exécuter la tâche lorsqu'un certain ensemble de critères est rempli. Une nouvelle tentative peut être tentée lorsqu'une tâche expire ou lorsque l'appareil tombe en panne. Pour réessayer l'exécution en raison d'un échec du délai d'expiration, vous devez activer la configuration du délai d'expiration.

Comment utiliser la configuration de nouvelle tentative

Effectuez les étapes suivantes pour nouvelle tentative de configuration

1. Déterminez s'il convient d'utiliser la configuration de nouvelle tentative pour FAILED ou TIMED_OUT les deux critères d'échec. Pour ce qui est de l'TIMED_OUTétat, une fois que l'état est signalé, AWS IoT Jobs réessaie automatiquement d'exécuter la tâche pour l'appareil.
2. Pour connaître l'FAILEDétat, vérifiez si l'échec de l'exécution de votre tâche peut être réessayé. S'il est possible de réessayer, programmez votre appareil pour qu'il signale un FAILURE état à. AWS IoT La section suivante décrit plus en détail les échecs pouvant être réessayés et non réessayables.
3. Spécifiez le nombre de nouvelles tentatives à utiliser pour chaque type d'échec en utilisant les informations précédentes. Pour un seul appareil, vous pouvez spécifier jusqu'à 10 nouvelles tentatives pour les deux types d'échec combinés. Les tentatives de nouvelle tentative s'arrêtent automatiquement lorsqu'une exécution réussit ou lorsqu'elle atteint le nombre de tentatives spécifié.
4. Ajoutez une configuration d'abandon pour annuler la tâche en cas d'échecs répétés afin d'éviter des frais supplémentaires liés à un grand nombre de tentatives.

Note

Lorsqu'une tâche arrive à la fin d'une occurrence récurrente de la fenêtre de maintenance, toutes les exécutions de IN_PROGRESS tâches continuent à exécuter les actions identifiées dans le document de travail jusqu'à ce qu'elles atteignent un état terminal. Si l'exécution d'une tâche atteint un état terminal pendant FAILED ou TIMED_OUT en dehors d'une fenêtre de maintenance, une nouvelle tentative aura lieu dans la fenêtre suivante si les tentatives ne sont pas épuisées.

À la prochaine occurrence startTime de la fenêtre de maintenance, une nouvelle exécution de tâche sera créée et l'état entrera dans l'état QUEUED jusqu'à ce que le périphérique soit prêt à démarrer.

Réessayer et abandonner la configuration

Chaque nouvelle tentative entraîne des frais supplémentaires pour vous. Compte AWS Pour éviter des frais supplémentaires en cas d'échecs répétés de nouvelles tentatives, nous vous recommandons d'ajouter une configuration d'abandon. Pour de plus amples informations sur la tarification, veuillez consulter [Tarification AWS IoT Device Management](#).

Il est possible que plusieurs tentatives échouent lorsqu'un pourcentage élevé de vos appareils expirent ou signalent un échec. Dans ce cas, vous pouvez utiliser la configuration d'abandon pour annuler la tâche et éviter toute exécution de tâche en file d'attente ou toute nouvelle tentative.

Note

Lorsque les critères d'abandon sont remplis pour annuler l'exécution d'une tâche, seules les exécutions de QUEUED tâches sont annulées. Aucune tentative en attente pour l'appareil ne sera

tentée. Toutefois, les exécutions de tâches en cours dotées d'un IN_PROGRESS statut ne seront pas annulées.

Avant de réessayer l'exécution d'une tâche qui a échoué, nous vous recommandons également de vérifier si l'échec de l'exécution de votre tâche peut être réessayé, comme décrit dans la section suivante.

Réessayez en cas de type d'échec de **FAILED**

Pour tenter de nouvelles tentatives pour le type d'échec deFAILED, vos appareils doivent être programmés pour signaler l'FAILUREétat de l'échec de l'exécution d'une tâche àAWS IoT. Définissez la configuration des nouvelles tentatives avec les critères permettant de réessayer l'exécution des FAILED tâches et spécifiez le nombre de nouvelles tentatives à effectuer. Lorsque AWS IoT Jobs détecte l'FAILUREétat, il tente automatiquement de réessayer d'exécuter la tâche pour l'appareil. Les tentatives se poursuivent jusqu'à ce que l'exécution de la tâche réussisse ou jusqu'à ce que le nombre maximum de tentatives de nouvelles tentatives soit atteint.

Vous pouvez suivre chaque nouvelle tentative et la tâche en cours d'exécution sur ces appareils. En suivant l'état d'exécution, une fois que le nombre de tentatives spécifié a été effectué, vous pouvez utiliser votre appareil pour signaler les échecs et lancer une nouvelle tentative.

Défaillances pouvant être réessayées et non réessayables

L'échec de l'exécution de votre tâche peut être réessayable ou non. Chaque nouvelle tentative peut entraîner des frais pour vous. Compte AWS Pour éviter des frais supplémentaires liés à de multiples tentatives, pensez d'abord à vérifier si l'échec de l'exécution de votre tâche peut être réessayé. Un exemple d'échec lors d'une nouvelle tentative inclut une erreur de connexion rencontrée par votre appareil lors d'une tentative de téléchargement du document de travail à partir d'une URL Amazon S3. Si l'échec de l'exécution de votre tâche est réessayable, programmez votre appareil pour qu'il signale un FAILURE état en cas d'échec de l'exécution de la tâche. Définissez ensuite la configuration des nouvelles tentatives pour réessayer les exécutionsFAILED.

S'il n'est pas possible de retenter l'exécution, nous vous recommandons de programmer l'appareil pour qu'il signale l'état de l'appareil, afin d'éviter toute nouvelle tentative et ainsi d'entraîner des frais supplémentaires sur votre compte. REJECTED AWS IoT Par exemple, lorsque votre appareil ne peut pas recevoir une nouvelle mise à jour d'une tâche ou lorsqu'il rencontre une erreur de mémoire lors de l'exécution d'une tâche, il peut notamment s'avérer impossible d'effectuer une nouvelle tentative. Dans ces cas, AWS IoT Jobs ne réessaiera pas d'exécuter la tâche car il ne réessaie d'exécuter la tâche que lorsqu'il détecte un état FAILED orTIMED_OUT.

Une fois que vous avez déterminé qu'un échec d'exécution d'une tâche peut être réessayé, si une nouvelle tentative échoue toujours, pensez à consulter les journaux de l'appareil.

Note

Lorsqu'une tâche dotée de la configuration de planification facultative atteint sa valeurendTime, la endBehavior personne sélectionnée arrête le déploiement du document de travail sur tous les appareils restants du groupe cible et dicte la marche à suivre pour les exécutions de tâches restantes. Les tentatives sont relancées si elles sont sélectionnées via la configuration des nouvelles tentatives.

Réessayez en cas de type d'échec de **TIMEOUT**

Si vous activez le délai d'expiration lors de la création d'une tâche, AWS IoT Jobs tentera de réessayer d'exécuter la tâche pour l'appareil lorsque l'état passe de àIN_PROGRESS. TIMED_OUT Ce changement d'état peut se produire lorsque le chronomètre en cours expire ou lorsqu'un chronomètre que vous avez spécifié est IN_PROGRESS activé puis expire. Les nouvelles tentatives se poursuivent jusqu'à ce que l'exécution de la tâche réussisse ou jusqu'à ce que le nombre maximum de tentatives de nouvelles tentatives pour ce type d'échec soit atteint.

Mises à jour continues des tâches et de l'appartenance aux groupes d'objets

Pour les tâches continues dont le statut est tel que `IN_PROGRESS`, le nombre de nouvelles tentatives est remis à zéro lorsque l'appartenance d'un objet à un groupe est mise à jour. Supposons, par exemple, que vous avez spécifié cinq nouvelles tentatives et que trois tentatives ont déjà été effectuées. Si un objet est maintenant supprimé du groupe d'objets puis rejoint le groupe, comme c'est le cas pour les groupes d'objets dynamiques, le nombre de nouvelles tentatives est remis à zéro. Vous pouvez désormais effectuer cinq nouvelles tentatives pour votre groupe d'objets au lieu des deux tentatives restantes. En outre, lorsqu'un objet est supprimé du groupe d'objets, les nouvelles tentatives sont annulées.

Spécifiez des configurations supplémentaires

Lorsque vous créez une tâche ou un modèle de tâche, vous pouvez spécifier ces configurations supplémentaires. La section suivante montre à quel moment vous pouvez spécifier ces configurations.

- Lors de la création d'un modèle de travail personnalisé. Les paramètres de configuration supplémentaires que vous spécifiez seront enregistrés lorsque vous créerez une tâche à partir du modèle.
- Lors de la création d'une tâche personnalisée à l'aide d'un fichier de tâche. Le fichier de travail peut être un fichier JSON chargé dans un compartiment S3.
- Lors de la création d'une tâche personnalisée à l'aide d'un modèle de tâche personnalisé. Si ces paramètres sont déjà spécifiés dans le modèle, vous pouvez les réutiliser ou les remplacer en spécifiant de nouveaux paramètres de configuration.
- Lors de la création d'une tâche personnalisée à l'aide d'un modèle AWS géré.

Rubriques

- [Spécifiez les configurations des tâches à l'aide du AWS Management Console \(p. 789\)](#)
- [Spécifiez les configurations des tâches à l'aide de l'API AWS IoT Jobs \(p. 791\)](#)

Spécifiez les configurations des tâches à l'aide du AWS Management Console

Vous pouvez ajouter les différentes configurations pour votre tâche à l'aide de la AWS IoT console. Une fois que vous avez créé un travail, vous pouvez voir les détails de l'état de vos configurations de travail sur la page des détails du travail. Pour de plus amples informations sur les différentes configurations et leur fonctionnement, veuillez consulter [Comment fonctionnent les configurations de tâches \(p. 779\)](#).

Ajoutez les configurations de travail lorsque vous créez un travail ou un modèle de travail.

Lors de la création d'un modèle de travail personnalisé

Pour spécifier la configuration du déploiement lors de la création d'un modèle de tâche personnalisé

1. Accédez au [hub de modèles de Job de la AWS IoT console](#) et choisissez Créer un modèle de tâche.
2. Spécifiez les propriétés du modèle de tâche, fournissez le document de travail, développez la configuration que vous souhaitez ajouter, puis spécifiez les paramètres de configuration.

Lors de la création d'une tâche personnalisée

Pour spécifier la configuration du déploiement lors de la création d'une tâche personnalisée

1. Accédez au [hub Job de la AWS IoT console](#) et choisissez Create job.
2. Choisissez Créer une tâche personnalisée et spécifiez les propriétés de la tâche, les cibles et indiquez si vous souhaitez utiliser un fichier de travail ou un modèle pour le document de travail. Vous pouvez utiliser un modèle personnalisé ou un modèle AWS géré.

3. Choisissez la configuration de la tâche, puis développez la configuration de déploiement pour spécifier si vous souhaitez utiliser un taux constant ou un taux exponentiel. Spécifiez ensuite les paramètres de configuration.

La section suivante présente les paramètres que vous pouvez spécifier pour chaque configuration.

Configuration du déploiement

Vous pouvez spécifier si vous souhaitez utiliser un taux de déploiement constant ou un taux exponentiel.

- Définissez un taux de déploiement constant

Pour définir un taux constant pour l'exécution des tâches, choisissez Taux constant, puis spécifiez le maximum par minute pour la limite supérieure du taux. Cette valeur est facultative et est comprise entre 1 et 1 000. Si vous ne le définissez pas, il utilise 1000 comme valeur par défaut.

- Définissez un taux de déploiement exponentiel

Pour définir un taux exponentiel, choisissez Taux exponentiel, puis spécifiez les paramètres suivants :

- Tarif de base par minute

La vitesse à laquelle les tâches sont exécutées jusqu'à ce que le seuil du nombre d'appareils notifiés ou du nombre d'appareils réussis soit atteint pour les critères d'augmentation du débit.

- Facteur d'incrémentation

Facteur exponentiel selon lequel le taux de déploiement augmente une fois que le seuil du nombre d'appareils notifiés ou du nombre d'appareils réussis est atteint pour les critères d'augmentation du débit.

- Critères d'augmentation des taux

Le seuil pour le nombre d'appareils notifiés ou pour le nombre d'appareils réussis.

Interruption de la configuration

Choisissez Ajouter une nouvelle configuration et spécifiez les paramètres suivants pour chaque configuration :

- Type de défaillance

Spécifie les types d'échec qui déclenchent l'abandon d'une tâche. Il s'agit notamment de FAILED, REJECTED, TIMED_OUT ou ALL.

- Facteur d'incrémentation

Spécifie le nombre d'exécutions de tâches terminées qui doivent avoir lieu avant que les critères d'abandon des tâches ne soient atteints.

- Pourcentage de seuil

Spécifie le nombre total d'opérations exécutées qui déclenchent l'abandon d'une tâche

Configuration de la planification

Chaque tâche peut démarrer immédiatement après sa création initiale, être programmée pour démarrer à une date et une heure ultérieures ou avoir lieu pendant une fenêtre de maintenance récurrente.

Choisissez Ajouter une nouvelle configuration et spécifiez les paramètres suivants pour chaque configuration :

- Début Job

Spécifiez la date et l'heure de lancement de tâche

- Le créneau de maintenance périodique

Une fenêtre de maintenance récurrente définit la date et l'heure spécifiques auxquelles une tâche peut déployer le document de travail sur les machines cibles de la tâche. La fenêtre de maintenance peut être répétée tous les jours, toutes les semaines, tous les mois ou selon une certaine instance.

- Fin Job

Spécifiez la date et l'heure de la fin de tâche

- Comportement en fin de Job

Sélectionnez un comportement final pour toutes les exécutions de tâches inachevées lorsque la tâche est terminée.

Note

Lorsqu'une tâche avec la configuration de planification facultative et l'heure de fin sélectionnée atteint l'heure de fin, la tâche arrête le déploiement sur tous les appareils restants du groupe cible. Il utilise également le comportement final sélectionné pour procéder aux exécutions de tâches restantes et à leurs tentatives de nouvelle tentative conformément à la configuration des nouvelles tentatives.

Configuration du délai d'expiration

Par défaut, il n'y a pas de délai d'expiration et l'exécution de votre tâche est annulée ou supprimée. Pour utiliser les délais d'expiration, choisissez Activer le délai d'expiration, puis spécifiez une valeur de délai comprise entre 1 minute et 7 jours.

Nouvelle tentative

Note

Une fois qu'une tâche a été créée, le nombre de tentatives ne peut pas être mis à jour. Vous ne pouvez supprimer la configuration de nouvelle tentative que pour tous les types d'échec. Lorsque vous créez une tâche, prenez en compte le nombre approprié de tentatives à utiliser pour votre configuration. Pour éviter d'encourir des coûts supplémentaires en raison d'éventuels échecs de nouvelles tentatives, ajoutez une configuration d'abandon.

Choisissez Ajouter une nouvelle configuration et spécifiez les paramètres suivants pour chaque configuration :

- Type de défaillance

Spécifie les types d'échec qui doivent déclencher une nouvelle tentative d'exécution de la tâche. Il s'agit notamment de Failed, Timeout et All.

- Nombre de nouvelles tentatives

Spécifie le nombre de nouvelles tentatives pour le type d'échec choisi. Pour les deux types d'échec combinés, il est possible de tenter jusqu'à 10 nouvelles tentatives.

Spécifiez les configurations des tâches à l'aide de l'API AWS IoT Jobs

Vous pouvez utiliser l'API [CreateJob](#) ou l'[CreateJobTemplate](#) API pour spécifier les différentes configurations de tâches. Les sections suivantes décrivent comment ajouter ces configurations. Une fois que vous avez

ajouté les configurations, vous pouvez [JobExecutionSummary](#)les [JobExecutionSummaryForJob](#)utiliser et voir leur état.

Pour de plus amples informations sur les différentes configurations et leur fonctionnement, veuillez consulter[Comment fonctionnent les configurations de tâches \(p. 779\)](#).

Configuration du déploiement

Vous pouvez spécifier un taux de déploiement constant ou un taux de déploiement exponentiel pour votre configuration de déploiement.

- Définissez un taux de déploiement constant

Pour définir un taux de déploiement constant, utilisez l'[JobExecutionsRolloutConfig](#)objet pour ajouter le `maximumPerMinute` paramètre à la `CreateJob` demande. Ce paramètre spécifie la limite supérieure de la vitesse d'exécution de tâche. Cette valeur est facultative et est comprise entre 1 et 1 000. Si vous ne définissez pas la valeur, elle utilise 1000 comme valeur par défaut.

```
"jobExecutionsRolloutConfig": {  
    "maximumPerMinute": 1000  
}
```

- Définissez un taux de déploiement exponentiel

Pour définir un taux de déploiement de tâches variable, utilisez l'[JobExecutionsRolloutConfig](#)objet. Vous pouvez configurer la `ExponentialRolloutRate` propriété lorsque vous exécutez l'opération `CreateJob` d'API. L'exemple suivant définit un taux de déploiement exponentiel à l'aide du `exponentialRate` paramètre. Pour de plus amples informations sur les paramètres, veuillez consulter [ExponentialRolloutRate](#).

```
{  
  ...  
  "jobExecutionsRolloutConfig": {  
    "exponentialRate": {  
      "baseRatePerMinute": 50,  
      "incrementFactor": 2,  
      "rateIncreaseCriteria": {  
        "number0fNotifiedThings": 1000,  
        "number0fSucceededThings": 1000  
      },  
      "maximumPerMinute": 1000  
    }  
  }  
  ...  
}
```

Où le paramètre :

`baseRatePerMinute`

Spécifie la vitesse à laquelle les tâches sont exécutées jusqu'à ce que le `number0fSucceededThings` seuil `number0fNotifiedThings` or soit atteint.

`incrementFactor`

Spécifie le facteur exponentiel selon lequel le taux de déploiement augmente une fois le `number0fSucceededThings` seuil `number0fNotifiedThings` or atteint.

`rateIncreaseCriteria`

Spécifie le `number0fSucceededThings` seuil `number0fNotifiedThings` ou.

Interruption de la configuration

Pour ajouter cette configuration à l'aide de l'API, spécifiez le [AbortConfig](#) paramètre lorsque vous exécutez l'[CreateJob](#) opération ou l'opération [CreateJobTemplate](#) d'API. L'exemple suivant montre une configuration d'abandon pour le déploiement d'une tâche qui a connu plusieurs échecs d'exécution, comme spécifié dans l'opération [CreateJob](#) API.

Note

La suppression de l'exécution d'une tâche affecte la valeur de calcul de l'exécution totale terminée. 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.

```
"abortConfig": {
    "criteriaList": [
        {
            "action": "CANCEL",
            "failureType": "FAILED",
            "minNumberOfExecutedThings": 100,
            "thresholdPercentage": 20
        },
        {
            "action": "CANCEL",
            "failureType": "TIMED_OUT",
            "minNumberOfExecutedThings": 200,
            "thresholdPercentage": 50
        }
    ]
}
```

Où le paramètre :

action

Spécifie l'action à effectuer lorsque les critères d'abandon sont respectés. Ce paramètre est obligatoire et CANCEL est la seule valeur valide.

failureType

Spécifie les types d'échec qui doivent entraîner l'abandon d'une tâche. Les valeurs valides sont FAILED, REJECTED, TIMED_OUT et ALL.

minNumberOfExecutedThings

Spécifie le nombre d'exécutions de tâches terminées qui doivent avoir lieu avant que les critères d'abandon des tâches ne soient atteints. 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.

thresholdPercentage

Spécifie le nombre total d'éléments pour lesquels des tâches sont exécutées et qui peuvent entraîner l'abandon d'une tâche. Dans cet exemple, AWS IoT effectue des vérifications séquentielles et initie l'abandon d'une tâche si le pourcentage de seuil est atteint. Si au moins 20 % des exécutions complètes échouent alors que 100 exécutions sont terminées, le déploiement de la tâche est annulé. Si ce critère n'est pas rempli, AWS IoT vérifie si au moins 50 % des exécutions terminées ont expiré au bout de 200 exécutions. Si tel est le cas, cela annule le déploiement de la tâche.

Configuration de la planification

Pour ajouter cette configuration à l'aide de l'API, spécifiez l'option facultative [SchedulingConfig](#) lorsque vous exécutez l'[CreateJob](#) opération ou l'opération [CreateJobTemplate](#) d'API.

```
"SchedulingConfig": {
```

```

        "endBehavior": string
        "endTime": string
        "maintenanceWindows": string
        "startTime": string
    }
}

```

Où le paramètre :

startTime

Spécifie la date et l'heure de lancement de tâche

endTime

Spécifie la date et l'heure de la fin de tâche

Fenêtres de maintenance

Spécifie si une fenêtre de maintenance facultative a été sélectionnée pour la tâche planifiée afin de déployer le document de travail sur tous les appareils du groupe cible. Le format de chaîne pour maintenanceWindow est YYYY/MM/DD pour la date et hh:mm pour l'heure.

Comportement final

Spécifie le comportement d'une tâche planifiée lorsqu'elle atteint le endTime.

Note

L'option SchedulingConfig correspondant à une tâche est visible dans les [DescribeJobTemplate](#) API [DescribeJob](#).

Configuration du délai d'expiration

Pour ajouter cette configuration à l'aide de l'API, spécifiez le [TimeoutConfig](#) paramètre lorsque vous exécutez l'[CreateJob](#) opération ou l'opération [CreateJobTemplate](#) d'API.

Pour utiliser la configuration du délai d'expiration

1. Pour définir le chronomètre en cours lorsque vous créez une tâche ou un modèle de tâche, définissez une valeur pour la `inProgressTimeoutInMinutes` propriété de l'[TimeoutConfig](#) objet facultatif.

```

    "timeoutConfig": {
        "inProgressTimeoutInMinutes": number
    }
}

```

2. Pour spécifier un temporisateur pour l'exécution d'une tâche, définissez une valeur pour le `stepTimeoutInMinutes` moment où 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.

Note

[UpdateJobExecution](#) peut supprimer un chronomètre qui a déjà été créé en créant un nouveau chronomètre avec une valeur de -1.

```

{
    ...
    "statusDetails": {
        "string" : "string"
    },
    "stepTimeoutInMinutes": number
}

```

3. Pour créer un nouveau chronomètre, vous pouvez également appeler l'opération [StartNextPendingJobExecutionAPI](#).

Nouvelle tentative

Note

Lorsque vous créez une tâche, prenez en compte le nombre approprié de tentatives à utiliser pour votre configuration. Pour éviter d'encourir des coûts supplémentaires en raison d'éventuels échecs de nouvelles tentatives, ajoutez une configuration d'abandon. Une fois qu'une tâche a été créée, le nombre de tentatives ne peut pas être mis à jour. Vous pouvez uniquement définir le nombre de tentatives sur 0 à l'aide de l'opération [UpdateJobAPI](#).

Pour ajouter cette configuration à l'aide de l'API, spécifiez le `jobExecutionsRetryConfig` paramètre lorsque vous exécutez l'[CreateJob](#) opération ou l'opération [CreateJobTemplate](#) d'API.

```
{  
  ...  
  "jobExecutionsRetryConfig": {  
    "criteriaList": [  
      {  
        "failureType": "string",  
        "numberOfRetries": number  
      }  
    ]  
  }  
  ...  
}
```

Où CriteriaList est un tableau indiquant la liste des critères qui déterminent le nombre de nouvelles tentatives autorisées pour chaque type d'échec d'une tâche.

Appareils et tâches

Les appareils peuvent communiquer avec AWS IoT Jobs à l'aide de MQTT, de HTTP Signature Version 4 ou de HTTP TLS. Pour déterminer le point de terminaison à utiliser lorsque votre appareil communique avec AWS IoT Jobs, exécutez la `DescribeEndpoint` commande. 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"  
}
```

Utilisation du protocole MQTT

Les appareils peuvent communiquer avec AWS IoT Jobs à l'aide du 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 des sujets MQTT pour interroger ou mettre à jour l'état du lancement 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 [Protocoles de communication des appareils \(p. 89\)](#).

Avec cette méthode de communication, votre appareil utilise son certificat et sa clé privée spécifiques pour s'authentifier auprès de Jobs. AWS IoT

Vos appareils peuvent s'abonner aux rubriques suivantes. **thing-name** Le nom de l'élément associé au périphérique.

- **\$aws/things/*thing-name*/jobs/notify**

Abonnez-vous à cette rubrique pour être averti lorsqu'un lancement de tâche est ajouté ou supprimé de la liste des lancements de tâches en attente.

- **\$aws/things/*thing-name*/jobs/notify-next**

Abonnez-vous à cette rubrique pour être averti lorsque la prochaine exécution de tâche en attente sera modifiée.

- **\$aws/things/*thing-name*/*request-name*/accepted**

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. *request-name* Voici le nom d'une requête telle que Get et le sujet peut être :\$aws/things/myThing/jobs/get. AWS IoT Jobs publie ensuite des messages de réussite sur le \$aws/things/myThing/jobs/get/accepted sujet.

- **\$aws/things/*thing-name*/*request-name*/rejected**

request-name Voici le nom d'une requête telle queGet. Si la demande échoue, AWS IoT Jobs publie des messages d'échec sur le \$aws/things/myThing/jobs/get/rejected sujet.

Vous pouvez également utiliser les opérations de l'API HTTPS suivantes :

- Mettre à jour le statut d'une exécution de tâche en appelant l'API [UpdateJobExecution](#).
- Interroger le statut d'une exécution de tâche en appelant l'API [DescribeJobExecution](#).
- Récupérer la liste des exécutions de tâche en attente en appelant l'API [GetPendingJobExecutions](#).
- Récupérez la prochaine exécution de tâche en attente en appelant l'[DescribeJobExecution](#) API avec jobId as\$next.
- Obtenir et démarrer la prochaine exécution de tâche en attente en appelant l'API [StartNextPendingJobExecution](#).

Utilisation de la Signature HTTP Version 4

Les appareils peuvent communiquer avec AWS IoT Jobs à l'aide de la version 4 de la signature HTTP sur le port 443. Il s'agit de la méthode utilisée par les AWS SDK et l'interface de ligne de commande. Pour plus d'informations sur ces outils, voir [AWS CLI Command Reference : iot-jobs-data](#) ou [AWSSDK et outils](#) et reportez-vous à la `iotJobsDataPlane` section correspondant à votre langue préférée.

Avec ce mode de communication, votre appareil utilise les informations d'identification IAM pour s'authentifier auprès AWS IoT de 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 ...
```

Utilisation du protocole HTTP TLS

Les appareils peuvent communiquer avec AWS IoT Jobs à l'aide du protocole HTTP TLS sur le port 8443 à l'aide d'un client logiciel tiers prenant en charge ce protocole.

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 MyThing qu'un appareil appelé est abonné 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 par code pour AWS IoT, le 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](#).

Rubriques

- [Flux de travail des appareils \(p. 797\)](#)
- [Flux de travail \(p. 799\)](#)
- [Notifications Jobs \(p. 802\)](#)

Flux de travail des appareils

Un appareil peut gérer les tâches qu'il exécute de l'une des manières suivantes.

- Décrochez le prochain emploi

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 \(p. 826\)](#) 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 \(p. 827\)](#) 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 \(p. 825\)](#).
4. Le cas échéant, vous pouvez ajouter un minuteur d'étape en définissant une valeur pour `stepTimeoutInMinutes` lorsque vous appelez [UpdateJobExecution \(p. 827\)](#) ou [StartNextPendingJobExecution \(p. 825\)](#).
5. Exécutez les actions spécifiées par le document de tâche à l'aide de l'API MQTT [UpdateJobExecution \(p. 827\)](#) pour faire état de l'avancement de la tâche.
6. Continuez de surveiller l'exécution de tâche en appelant l'API MQTT [DescribeJobExecution \(p. 826\)](#) avec ce jobId. Si l'exécution de la tâche est supprimée, [DescribeJobExecution \(p. 826\)](#) renvoie `unResourceNotFoundException`.

L'appareil doit pouvoir revenir à un état valide si l'exécution de la tâche est annulée ou supprimée alors que l'appareil exécute la tâche.

7. Une fois que la tâche est terminée,appelez l'API MQTT [UpdateJobExecution \(p. 827\)](#) 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 reste en ligne, il continue de recevoir des notifications concernant la prochaine exécution d'une tâche en attente. Cela inclut ses données d'exécution de tâches, lorsqu'elle termine une tâche ou lorsqu'une nouvelle exécution de tâche en attente est ajoutée. Dans ce cas, l'appareil continue comme décrit à l'étape 2.

- Sélectionnez parmi les offres d'emploi disponibles

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 \(p. 825\)](#) pour obtenir la liste des exécutions de tâche en attente.
3. Si la liste contient une ou plusieurs exécutions de tâches, sélectionnez-en une.
4. Appelez l'API MQTT [DescribeJobExecution \(p. 826\)](#) pour obtenir le document de tâche et autres détails, y compris tout état enregistré dans `statusDetails`.
5. Appelez l'API MQTT [UpdateJobExecution \(p. 827\)](#) 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 \(p. 827\)](#).
7. Exécutez les actions spécifiées par le document de tâche à l'aide de l'API MQTT [UpdateJobExecution \(p. 827\)](#) pour faire état de l'avancement de la tâche.
8. Continuez de surveiller l'exécution de tâche en appelant l'API MQTT [DescribeJobExecution \(p. 826\)](#) avec ce jobId. Si l'exécution de la tâche est annulée ou supprimée alors que l'appareil exécute la tâche, l'appareil doit pouvoir revenir à un état valide.

- Une fois que la tâche est terminée, appelez l'API MQTT [UpdateJobExecution \(p. 827\)](#) 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 [UpdateJobExecution \(p. 827\)](#) MQTT pour mettre à jour l'état de la tâche. REJECTED

Flux de travail

Ce qui suit montre les différentes étapes du flux de travail des tâches, depuis le démarrage d'une nouvelle tâche jusqu'à la notification de l'état d'avancement de l'exécution d'une tâche.

Démarrer un nouveau travail

Lorsqu'une nouvelle tâche est créée, AWS IoT Jobs publie un message sur le `$aws/things/thing-name/jobs/notify` sujet pour chaque appareil cible.

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.

Note

Pour les tâches comportant l'option facultative `SchedulingConfig`, l'état initial de la tâche sera maintenu à SCHEDULED. Lorsque la tâche atteint la valeur sélectionnée `startTime`, les événements suivants se produisent :

- L'état de la tâche sera mis à jour vers IN_PROGRESS.
- La tâche commencera à être déployée du document de travail sur tous les appareils du groupe cible.

Obtenir des informations sur l'emploi

Pour obtenir plus d'informations sur l'exécution d'une tâche, l'appareil appelle l'API MQTT [DescribeJobExecution \(p. 826\)](#) avec le champ `includeJobDocument` défini sur `true` (valeur par défaut).

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 >
    }
}
```

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 travail contient une URL présignée Amazon S3, l'appareil peut utiliser cette URL pour télécharger tous les fichiers requis pour la tâche.

Rapport du statut d'exécution de tâche

Au fur et à mesure que l'appareil exécute la tâche, il peut appeler l'API MQTT [UpdateJobExecution \(p. 827\)](#) pour mettre à jour le statut de l'exécution de la tâche.

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"
}
```

Les offres d'emploi répondent en publiant un message sur le \$aws/things/MyThing/jobs/0023/update/rejected sujet \$aws/things/MyThing/jobs/0023/update/accepted ou sur le sujet suivant :

```
{
    "clientToken": "client001",
    "timestamp": 1476289222841
}
```

Le périphérique permet de combiner les deux demandes précédentes en appelant [StartNextPendingJobExecution \(p. 825\)](#). 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 un [TimeoutConfig](#), le chronomètre en cours commence à s'exécuter. Vous pouvez également définir un minuteur pour l'exécution d'une tâche en définissant une valeur pour le stepTimeoutInMinutes moment où 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 également créer un chronomètre lorsque vousappelez [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.

Exécution du rapport terminée

Lorsque l'appareil a terminé l'exécution de la tâche, il appelle l'API MQTT [UpdateJobExecution \(p. 827\)](#). 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.

Par exemple :

```
{  
    "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 le \$aws/things/MyThing/jobs/notify sujet pour indiquer que l'exécution de la tâche est terminée :

```
{  
    "timestamp": 1476290692776,  
    "jobs": {}  
}
```

Tâches supplémentaires

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.

Par exemple :

```
{  
    "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.

Types de notifications de Job

Les notifications de tâche sont publiées dans les rubriques MQTT en tant que charges utilises JSON. Il existe deux types de notifications :

ListNotification

A ListNotification contient une liste d'au plus 15 exécutions de tâches en attente. 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).

Notification de liste (jusqu'à 15 exécutions de tâches en attente dans QUEUED ou IN_PROGRESS)	
Sans configuration de planification optionnelle ni fenêtre de maintenance récurrente (Jusqu'à 10 exécutions de tâches)	Avec configuration de planification optionnelle et fenêtre de maintenance récurrente (Jusqu'à 5 exécutions de tâches)
Apparaît toujours dans leListNotification.	N'apparaît que dans la fenêtre ListNotification pendant une maintenance.

NextNotification

- A NextNotification contient des informations récapitulatives sur l'exécution de la 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 sous forme deQUEUED, et c'est la première de la liste.
- Le statut d'une exécution de tâche existante qui n'était pas la première de la liste passe de QUEUED à IN_PROGRESS et devient la première 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 des appareils” \(p. 89\)](#).

Note

Les notifications ne sont pas disponibles lorsque vous utilisez HTTP Signature Version 4 ou HTTP TLS pour communiquer avec les tâches.

Le nom de la Job

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

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  
    } ]  
  }  
}
```

Si l'exécution de la tâche appelée *this-job* provient d'une tâche pour laquelle la configuration de planification facultative a été sélectionnée et que le déploiement du document de travail est prévu pour avoir lieu pendant une fenêtre de maintenance, elle n'apparaîtra que pendant une fenêtre de maintenance récurrente. En dehors d'une fenêtre de maintenance, la tâche appelée *this-job* sera exclue de la liste des exécutions de tâches en attente, comme indiqué dans l'exemple suivant.

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

```

        "lastUpdatedAt" : 10009,
        "executionNumber" : 1,
        "versionNumber" : 1
    } ],
    "QUEUED" : []
}
}

```

\$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"}
    }
}
```

Si l'exécution de la tâche appelée *other-job* provient d'une tâche pour laquelle la configuration de planification facultative a été sélectionnée et que le déploiement du document de travail est prévu pour avoir lieu pendant une fenêtre de maintenance, elle n'apparaîtra que pendant une fenêtre de maintenance récurrente. En dehors d'une fenêtre de maintenance, la tâche appelée *other-job* ne sera pas répertoriée comme la prochaine exécution de la tâche, comme le montre l'exemple suivant.

```
{} //No other pending jobs
```

```
{
    "timestamp" : 10011,
    "execution" : {
        "jobId" : "this-job",
        "queuedAt" : 10011,
        "lastUpdatedAt" : 10011,
        "executionNumber" : 1,
        "versionNumber" : 0,
        "jobDocument" : {"a":"b"}
    }
} // "this-job" is pending next to "other-job"
```

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 suivante montre les notifications publiées pour chaque sujet au fur et à mesure que des exécutions de tâches sont créées et passent d'un état 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
    }
}
```

```
{  
    "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
}
```

AWS IoTtâches, opérations d'API

AWS IoTL'API Jobs peut être utilisée pour l'une des catégories suivantes :

- Tâches administratives telles que la gestion et le contrôle des tâches. C'est le plan de contrôle.
- Les appareils qui exécutent ces tâches. Il s'agit du plan de données qui vous permet d'envoyer et de recevoir des données.

La gestion et le contrôle des Job utilisent une API de protocole HTTPS. Les appareils peuvent utiliser une API MQTT ou une API du protocole HTTPS. L'API du plan de contrôle est conçue pour un faible volume d'appels, comme c'est le cas lors de la création et du suivi de tâches. Elle ouvre généralement une connexion pour une seule demande, puis la ferme après réception de la réponse. Le protocole HTTPS du plan de données et l'API MQTT permettent des interrogations longues. Ces opérations d'API sont conçues pour des volumes de trafic importants pouvant atteindre des millions d'appareils.

Chaque API HTTPS AWS IoT Jobs possède une commande correspondante qui vous permet d'appeler l'API à partir du AWS Command Line Interface (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 ...
```

Si une erreur se produit pendant une opération, vous obtenez une réponse d'erreur contenant des informations sur l'erreur.

ErrorResponse

Contient les informations sur une erreur qui s'est produite au cours d'une opération du service AWS IoT Jobs.

L'exemple suivant illustre la syntaxe de cette opération :

```
{
  "code": "ErrorCode",
  "message": "string",
  "clientToken": "string",
  "timestamp": timestamp,
  "executionState": JobExecutionState
}
```

Ce qui suit en est une description ErrorResponse :

code

ErrorCode peut être réglé sur :

InvalidTopic

La demande a été envoyée à une rubrique de l'espace de noms AWS IoT Jobs qui ne correspond à aucune opération d'API.

InvalidJson

Le contenu de la requête n'a pas pu être interprété comme un code JSON valide codé en UTF-8.

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 modifier l'exécution de la tâche vers un état qui n'est pas valide en raison de l'état actuel de l'exécution de la tâche. Par exemple, une tentative de modification de l'état `SUCCEEDED` d'une requête en état `IN_PROGRESS`. Dans ce cas, le corps du message d'erreur contient aussi le champ `executionState`.

ResourceNotFound

Le sujet `JobExecution` spécifié par 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

Un objet [JobExecutionState](#). 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.

La liste suivante répertorie les opérations et les types de données de l'API Jobs.

- [API de gestion et de contrôle des tâches et types de données \(p. 809\)](#)
- [Tâches, opérations de l'appareil, API MQTT et HTTPS et types de données \(p. 823\)](#)

API de gestion et de contrôle des tâches et types de données

Les commandes suivantes sont disponibles pour la gestion et le contrôle des Job dans la CLI et via le protocole HTTPS.

- [Types de données de gestion et de contrôle des tâches \(p. 810\)](#)
- [Gestion des Job et opérations de l'API de contrôle \(p. 813\)](#)

Pour déterminer le paramètre `endpoint-url` de vos commandes CLI, exéutez 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 le protocole ALPNz-amzn-https-ca.

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 AWS IoT Jobs.

Tâche

L'objet Job contient des informations sur une tâche. L'exemple suivant montre la syntaxe :

```
{  
  "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": {  
    "criterialList": [  
        {  
            "action": "string",  
            "failureType": "string",  
            "minNumberOfExecutedThings": integer,  
            "thresholdPercentage": integer  
        }  
    ]  
},  
"SchedulingConfig": {  
    "startTime": string  
    "endTime": string  
    "timeZone": string  
  
    "endTimeBehavior": string  
},  
"timeoutConfig": {  
    "inProgressTimeoutInMinutes": long  
}  
}
```

Pour plus d'informations, consultez [Job](#) ou [job](#).

JobSummary

L'objet JobSummary contient un résumé de tâche. L'exemple suivant montre la syntaxe :

```
{  
    "jobArn": "string",  
    "jobId": "string",  
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED|SCHEDULED",  
    "targetSelection": "CONTINUOUS|SNAPSHOT",  
    "thingGroupId": "string",  
    "createdAt": timestamp,  
    "lastUpdatedAt": timestamp,  
    "completedAt": timestamp  
}
```

Pour plus d'informations, consultez [JobSummary](#) ou [job-summary](#).

JobExecution

L'objet JobExecution représente l'exécution d'une tâche sur un appareil. L'exemple suivant montre la syntaxe :

```
{  
    "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,  
    "lastUpdatedAt": timestamp  
}
```

```
"statusDetails": {  
    "detailsMap": {  
        "string": "string" ...  
    },  
    "status": "string"  
},  
"thingArn": "string",  
"versionNumber": 123  
}
```

Pour plus d'informations, consultez [JobExecution](#) ou [job-execution](#).

JobExecutionSummary

L'JobExecutionSummaryobjet contient des informations récapitulatives sur l'exécution des tâches.
L'exemple suivant montre la syntaxe :

```
{  
    "executionNumber": 1234567890,  
    "queuedAt": timestamp,  
    "lastUpdatedAt": timestamp,  
    "startedAt": timestamp,  
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED"  
}
```

Pour plus d'informations, consultez [JobExecutionSummary](#) ou [job-execution-summary](#).

JobExecutionSummaryForJob

L'objet JobExecutionSummaryForJob contient un récapitulatif des informations sur les exécutions de
tâche d'une tâche spécifique. L'exemple suivant montre la syntaxe :

```
{  
    "executionSummaries": [  
        {  
            "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyThing",  
            "jobExecutionSummary": {  
                "status": "IN_PROGRESS",  
                "lastUpdatedAt": 1549395301.389,  
                "queuedAt": 1541526002.609,  
                "executionNumber": 1  
            }  
        },  
        ...  
    ]  
}
```

Pour plus d'informations, consultez [JobExecutionSummaryForJob](#) ou [job-execution-summary-for-job](#).

JobExecutionSummaryForThing

L'objet JobExecutionSummaryForThing contient un récapitulatif des informations sur une exécution de
tâche sur un objet spécifique. L'exemple suivant illustre la syntaxe :

```
{  
    "executionSummaries": [  
        {  
            "jobExecutionSummary": {  
                "status": "IN_PROGRESS",  
                "lastUpdatedAt": 1549395301.389,  
                "queuedAt": 1541526002.609,  
                "executionNumber": 1  
            }  
        }  
    ]  
}
```

```
        "executionNumber": 1
    },
    "jobId": "MyThingJob"
},
...
]
```

Pour plus d'informations, consultez [JobExecutionSummaryForThing](#) ou [job-execution-summary-for-thing](#).

Gestion des Job et opérations de l'API de contrôle

Utilisez les opérations d'API ou les commandes CLI suivantes :

[AssociateTargetsWithJob](#)

Associe un groupe à une tâche continue. Les critères suivants doivent être satisfait :

- 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 request

```
POST /jobs/jobId/targets

{
  "targets": [ "string" ],
  "comment": "string"
}
```

Pour plus d'informations, veuillez consulter [AssociateTargetsWithJob](#).

CLI syntax

```
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"
}
```

Pour plus d'informations, veuillez consulter [associate-targets-with-job](#).

[CancelJob](#)

Annule une tâche.

HTTPS request

```
PUT /jobs/jobId/cancel

{
  "force": boolean,
  "comment": "string",
  "reasonCode": "string"
}
```

Pour plus d'informations, veuillez consulter [CancelJob](#).

CLI syntax

```
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"
}
```

Pour plus d'informations, veuillez consulter [cancel-job](#).

[CancelJobExecution](#)

Annule une exécution de tâche sur un appareil.

HTTPS request

```
PUT /things/thingName/jobs/jobId/cancel

{
  "force": boolean,
  "expectedVersion": "string",
  "statusDetails": {
    "string": "string"
    ...
  }
}
```

Pour plus d'informations, veuillez consulter [CancelJobExecution](#).

CLI syntax

```
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"  
  }  
}
```

Pour plus d'informations, veuillez consulter [cancel-job-execution](#).

CreateJob

Crée une tâche. Vous pouvez fournir le document de travail sous forme de lien vers un fichier dans un compartiment Amazon S3 (`documentSource` paramètre) ou dans le corps de la demande (`document` paramètre).

Une tâche peut être rendue continue en définissant le `targetSelection` paramètre facultatif sur `CONTINUOUS` (la valeur par défaut est `SNAPSHOT`). Une tâche continue peut être utilisée pour intégrer ou mettre à niveau des appareils au fur et à mesure qu'ils sont ajoutés à un groupe, car elle continue de s'exécuter et est lancée sur les éléments récemment ajoutés. Cela peut se produire même une fois que les éléments du groupe au moment de la création de la tâche ont terminé la tâche.

Une tâche peut comporter une option `TimeoutConfig`, qui définit la valeur du chronomètre en cours. 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 tous les groupes d'objets doivent se trouver dans votreCompte AWS.
- L'argument `documentSource` doit être une URL Amazon S3 valide pointant vers un document de travail. 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 travail est limitée à 32 Ko en raison de la limite de taille d'un message MQTT (128 Ko) et du chiffrement.
- Ils `jobId` doivent être uniques dans votreCompte AWS.

HTTPS request

```
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
        }
    ]
},
"SchedulingConfig": {
    "startTime": string
    "endTime": string
    "timeZone": string

    "endTimeBehavior": string

}
"timeoutConfig": {
    "inProgressTimeoutInMinutes": long
}
}
```

Pour plus d'informations, veuillez consulter [CreateJob](#).

CLI syntax

```
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"
}
}
```

Pour plus d'informations, veuillez consulter [create-job](#).

DeleteJob

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 request

```
DELETE /jobs/jobId?force=force
```

Pour plus d'informations, veuillez consulter [DeleteJob](#).

CLI syntax

```
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  
}
```

Pour plus d'informations, veuillez consulter [delete-job](#).

DeleteJobExecution

Supprime une exécution de tâche.

HTTPS request

```
DELETE /things/thingName/jobs/jobId/executionNumber/executionNumber?force=force
```

Pour plus d'informations, veuillez consulter [DeleteJobExecution](#).

CLI syntax

```
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  
}
```

Pour plus d'informations, veuillez consulter [delete-job-execution](#).

DescribeJob

Obtient des détails sur l'exécution de la tâche.

HTTPS request

```
GET /jobs/jobId
```

Pour plus d'informations, veuillez consulter [DescribeJob](#).

CLI syntax

```
aws iot describe-job \  
  --job-id <value> \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

Format cli-input-json :

```
{  
  "jobId": "string"  
}
```

Pour plus d'informations, veuillez consulter [describe-job](#).

DescribeJobExecution

Obtient les détails d'une exécution de tâche. Le statut de l'exécution de tâche doit être SUCCEEDED ou FAILED.

HTTPS request

```
GET /things/thingName/jobs/jobId?executionNumber=executionNumber
```

Pour plus d'informations, veuillez consulter [DescribeJobExecution](#).

CLI syntax

```
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  
}
```

Pour plus d'informations, veuillez consulter [describe-job-execution](#).

GetJobDocument

Obtient le document de tâche pour une tâche.

Note

Les URL d'espace réservé ne sont pas remplacées par des URL Amazon S3 présignalées dans le document renvoyé. Les URL présignalées sont générées uniquement lorsque le service AWS IoT Jobs reçoit une demande via MQTT.

HTTPS request

```
GET /jobs/jobId/job-document
```

Pour plus d'informations, veuillez consulter [GetJobDocument](#).

CLI syntax

```
aws iot get-job-document \  
  --job-id <value>
```

```
--job-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

Format `cli-input-json`:

```
{  
  "jobId": "string"  
}
```

Pour plus d'informations, veuillez consulter [get-job-document](#).

ListJobExecutionsForJob

Obtient la liste des exécutions de tâche d'une tâche.

HTTPS request

```
GET /jobs/jobId/things?status=status&maxResults=maxResults&nextToken=nextToken
```

Pour plus d'informations, veuillez consulter [ListJobExecutionsForJob](#).

CLI syntax

```
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"  
}
```

Pour plus d'informations, veuillez consulter [list-job-executions-for-job](#).

ListJobExecutionsForThing

Obtient la liste des exécutions de tâche d'un objet.

HTTPS request

```
GET /things/thingName/jobs?status=status&maxResults=maxResults&nextToken=nextToken
```

Pour plus d'informations, veuillez consulter [ListJobExecutionsForThing](#).

CLI syntax

```
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"
}
```

Pour plus d'informations, veuillez consulter [list-job-executions-for-thing](#).

ListJobs

Obtient la liste des tâches figurant dans votreCompte AWS.

HTTPS request

```
GET /jobs?
status=status&targetSelection=targetSelection&thingGroupName=thingGroupName&thingGroupId=thingGroupId
```

Pour plus d'informations, veuillez consulter [ListJobs](#).

CLI syntax

```
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"
}
```

Pour plus d'informations, veuillez consulter [list-jobs](#).

UpdateJob

Met à jour les champs pris en charge de la tâche spécifiée. Les valeurs ont été mises à jour pour ne `timeoutConfig` prendre effet que pour les nouveaux lancements en cours. Actuellement, les lancements en cours continuent de se lancer avec la configuration de délai d'expiration précédente.

HTTPS request

```
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
    }
  }
}
```

Pour plus d'informations, veuillez consulter [UpdateJob](#).

CLI syntax

```
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
    }
}
```

Pour plus d'informations, veuillez consulter [update-job](#).

Tâches, opérations de l'appareil, API MQTT et HTTPS et types de données

Les commandes suivantes sont disponibles sur les protocoles HTTPS et MQTT. Utilisez ces opérations d'API sur le plan de données pour les appareils exécutant les tâches.

Types de données MQTT et HTTPS de l'appareil Jobs

Les types de données suivants sont utilisés pour communiquer avec le service AWS IoTJobs via les protocoles MQTT et HTTPS.

JobExecution

Contient des données sur une exécution de tâche. L'exemple suivant montre la syntaxe :

```
{
  "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
}
```

Pour plus d'informations, consultez [JobExecution](#) ou [job-execution](#).

JobExecutionState

Contient les données sur l'état d'une exécution de tâche. L'exemple suivant montre la syntaxe :

```
{
  "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED",
```

```
"statusDetails": {  
    "string": "string"  
    ...  
}  
"versionNumber": "number"  
}
```

Pour plus d'informations, consultez [JobExecutionState](#) ou [job-execution-state](#).

JobExecutionSummary

Contient un sous-ensemble d'informations sur une exécution de tâche. L'exemple suivant montre la syntaxe :

```
{  
    "jobId": "string",  
    "queuedAt": timestamp,  
    "startedAt": timestamp,  
    "lastUpdatedAt": timestamp,  
    "versionNumber": "number",  
    "executionNumber": long  
}
```

Pour plus d'informations, consultez [JobExecutionSummary](#) ou [job-execution-summary](#).

Pour en savoir plus sur les opérations des API MQTT et HTTPS, consultez les sections suivantes :

- [Opérations de l'API MQTT de l'appareil de tâches \(p. 824\)](#)
- [API HTTP de la tâche \(p. 830\)](#)

Opérations de l'API MQTT de l'appareil de tâches

Vous pouvez émettre des commandes sur les appareils de tâches en publiant des messages MQTT dans les [rubriques réservées utilisées pour les commandes de tâches \(p. 124\)](#).

Votre client côté appareil doit être abonné aux rubriques des messages de réponse de ces commandes. Si vous utilisez le client de l'AWS IoT appareil, votre appareil s'abonnera automatiquement aux rubriques de réponse. Cela signifie que le gestionnaire de messages publiera les sujets des messages de réponse à l'attention du client qui a publié le message de commande, que votre client soit abonné ou non aux sujets du message de réponse. Ces messages de réponse ne passent pas par le courtier de messages et ne peuvent pas être souscrits par d'autres clients ou règles.

Lorsque vous abonnez aux rubriques relatives aux tâches et aux jobExecution événements pour votre solution de surveillance de flotte, activez [d'abord les événements d'exécution des tâches et \(p. 1253\)](#) des tâches afin de recevoir tous les événements du côté cloud. Les messages de progression des Job qui sont traités via le gestionnaire de messages et peuvent être utilisés par des AWS IoT règles sont publiés sous forme de [Événements Jobs \(p. 1264\)](#). Étant donné que le courtier de messages publie des messages de réponse, même sans y être abonné explicitement, votre client doit être configuré pour recevoir et identifier les messages qu'il reçoit. Votre client doit également confirmer que le **thingName** indiqué dans la rubrique du message entrant s'applique au nom de l'objet du client avant que celui-ci ne donne suite au message.

Note

Les messages AWS IoT envoyés en réponse aux messages de commande de l'API MQTT Jobs sont débités de votre compte, que vous y soyez abonné de manière explicite ou non.

Ce qui suit montre les opérations de l'API MQTT et leur syntaxe de demande et de réponse. Toutes les opérations d'API MQTT ont les paramètres suivants :

clientToken

Un jeton client facultatif utilisé pour corréler les demandes et les réponses. Entrez une valeur arbitraire ici et elle sera reflétée dans la réponse.

timestamp

Temps écoulé, en secondes, depuis l'époque à laquelle le message a été envoyé.

GetPendingJobExecutions

Obtient la liste de toutes les tâches qui ne se trouvent pas dans un état terminal, pour un objet déterminé.

Pour appeler cette API, publiez un message sur \$aws/things/*thingName*/jobs/get.

Charge utile de la demande :

```
{ "clientToken": "string" }
```

Le courtier de messages \$aws/things/*thingName*/jobs/get/accepted publiera, \$aws/things/*thingName*/jobs/get/rejected même sans abonnement spécifique. Toutefois, pour que votre client reçoive les messages, il doit les écouter. Pour plus d'informations, consultez [la note concernant les messages de l'API Jobs \(p. 824\)](#).

Charge utile de la réponse :

```
{
  "inProgressJobs" : [ JobExecutionSummary ... ],
  "queuedJobs" : [ JobExecutionSummary ... ],
  "timestamp" : 1489096425069,
  "clientToken" : "client-001"
}
```

Où `inProgressJobs` et `queuedJobs` renvoie une liste d'[JobExecutionSummary \(p. 824\)](#) objets dont le statut est IN_PROGRESS ou QUEUED.

StartNextPendingJobExecution

Obtient et démarre l'exécution de tâche en attente suivante pour un objet (statut IN_PROGRESS ou QUEUED).

- Toutes les exécutions de tâches comportant un 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 passe à IN_PROGRESS et les détails de l'état de l'exécution de la tâche sont définis comme spécifié.
- Si la prochaine exécution de tâche en attente est déjà en cours IN_PROGRESS, les détails de son statut ne sont pas modifiés.
- Si aucune exécution de tâche n'est en attente, la réponse n'inclut pas le execution champ.
- Vous pouvez éventuellement créer un chronomètre en définissant une valeur pour la stepTimeoutInMinutes propriété. 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.

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 l'état d'exécution de la tâche n'est pas défini sur un état terminal avant l'expiration de ce temporisateur ou avant que le temporisateur ne soit réinitialisé (en appelant `UpdateJobExecution`, en définissant le statut sur `IN_PROGRESS` et en spécifiant une nouvelle valeur de délai dans le champ `stepTimeoutInMinutes`), l'état d'exécution de la tâche est défini sur `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`).

Le courtier de messages `$aws/things/thingName/jobs/start-next/accepted` publiera, `$aws/things/thingName/jobs/start-next/rejected` même sans abonnement spécifique. Toutefois, pour que votre client reçoive les messages, il doit les écouter. Pour plus d'informations, consultez [la note concernant les messages de l'API Jobs \(p. 824\)](#).

Charge utile de la réponse :

```
{  
    "execution" : JobExecutionData,  
    "timestamp" : timestamp,  
    "clientToken" : "string"  
}
```

Où se `execution` trouve un [JobExecution \(p. 823\)](#) objet. Par exemple :

```
{  
    "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,  
}
```

DescribeJobExecution

Permet d'obtenir des informations détaillées sur une exécution de tâche.

Vous pouvez définir le `jobId $next` pour renvoyer la prochaine exécution de tâche en attente pour un objet (avec un statut de `IN_PROGRESS` ou `QUEUED`).

Pour appeler cette API, publiez un message sur \$aws/things/*thingName*/jobs/*jobId*/get.

Charge utile de la demande :

```
{  
  "jobId" : "022",  
  "thingName" : "MyThing",  
  "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 \$next à utiliser pour renvoyer l'exécution de tâche en attente suivante pour un objet (avec un statut de IN_PROGRESS ou QUEUED). Dans ce cas, toutes les exécutions de tâches avec statut IN_PROGRESS sont renvoyées en premier. Les exécutions de tâche sont renvoyées dans l'ordre selon lequel elles ont été créées.

executionNumber

(Facultatif) Un numéro qui identifie l'exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est renvoyée.

includeJobDocument

(Facultatif) À moins que ce paramètre ne soit défini sur false, la réponse contient le document de travail. La valeur par défaut est true.

Le courtier de messages \$aws/things/*thingName*/jobs/*jobId*/get/accepted publiera, \$aws/things/*thingName*/jobs/*jobId*/get/rejected même sans abonnement spécifique. Toutefois, pour que votre client reçoive les messages, il doit les écouter. Pour plus d'informations, consultez [la note concernant les messages de l'API Jobs \(p. 824\)](#).

Charge utile de la réponse :

```
{  
  "execution" : JobExecutionData,  
  "timestamp": "timestamp",  
  "clientToken": "string"  
}
```

Où se execution trouve un [JobExecution \(p. 823\)](#) objet.

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.

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

Le nouveau statut de l'exécution de la 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 la tâche enregistrée dans le service AWS IoT Jobs ne correspond pas, la mise à jour est rejetée avec une VersionMismatch erreur. Et [ErrorResponse \(p. 808\)](#) qui contient les données d'état d'exécution de la tâche en cours est également renvoyé. (Il est donc inutile d'effectuer une demande `DescribeJobExecution` distincte pour obtenir les données du statut d'exécution de tâche.)

executionNumber

(Facultatif) Un numéro qui identifie l'exécution de tâche sur un appareil. S'il n'est pas indiqué, la dernière exécution de tâche est utilisée.

includeJobExecutionState

(Facultatif) Lorsqu'elle est incluse et définie sur true, la réponse contient le JobExecutionState champ. La valeur par défaut est false.

includeJobDocument

(Facultatif) Lorsqu'elle est incluse et définie sur true, la réponse contient le JobDocument. La valeur par défaut est false.

stepTimeoutInMinutes

Spécifie la durée pendant laquelle cet appareil doit terminer l'exécution de la tâche. Si l'état d'exécution de la tâche n'est pas défini sur un état terminal avant l'expiration de ce temporisateur ou avant que le temporisateur ne soit réinitialisé, l'état d'exécution de la tâche est défini sur TIMED_OUT. La définition ou la réinitialisation de ce délai n'a aucun effet sur le délai d'exécution de la tâche qui aurait pu être spécifié lors de la création de la tâche.

Le courtier de messages \$aws/things/*thingName*/jobs/*jobId*/update/accepted publiera, \$aws/things/*thingName*/jobs/*jobId*/update/rejected même sans abonnement spécifique. Toutefois, pour que votre client reçoive les messages, il doit les écouter. Pour plus d'informations, consultez [la note concernant les messages de l'API Jobs \(p. 824\)](#).

Charge utile de la réponse :

```
{
    "executionState": JobExecutionState,
```

```
{"jobDocument": "string",
"timestamp": timestamp,
"clientToken": "string"
}
```

executionState

Un objet [JobExecutionState \(p. 823\)](#).

jobDocument

Objet de [document de tâche \(p. 740\)](#).

timestamp

Temps écoulé, en secondes, depuis l'époque à laquelle le message a été envoyé.

clientToken

Jeton client utilisé pour établir une corrélation entre les demandes et les réponses.

Lorsque vous utilisez le protocole MQTT, vous pouvez également effectuer les mises à jour suivantes :

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.

Utilisez la rubrique :

`$aws/things/thingName/jobs/notify`

Charge utile du message :

```
{
"jobs" : {
    "JobExecutionState": [ JobExecutionSummary ... ],
    },
    "timestamp": timestamp
}
```

NextJobExecutionChanged

Envoyé chaque fois que l'exécution d'une tâche est modifiée sur la liste des exécutions de tâches en attente pour un objet, comme défini [DescribeJobExecution](#) par jobId\$next. Ce message n'est pas envoyé lorsque les détails d'exécution de la tâche suivante changent, mais uniquement lorsque la tâche suivante qui serait renvoyée par [DescribeJobExecution](#) with jobId \$next a changé. Considérons les exécutions de tâches J1 et J2 dont le statut est. QUEUED J1 est l'exécution suivante sur la liste des exécutions de tâche en attente. Si l'état de J2 est modifié IN_PROGRESS alors que l'état de J1 reste inchangé, cette notification est envoyée et contient les détails de J2.

Utilisez la rubrique :

`$aws/things/thingName/jobs/notify-next`

Charge utile du message :

```
{
"execution" : JobExecution,
"timestamp": timestamp,
}
```

API HTTP de la tâche

Les appareils peuvent communiquer avec AWS IoT Jobs à l'aide de la version 4 de la signature HTTP sur le port 443. Il s'agit de la méthode utilisée par les AWS SDK et l'interface de ligne de commande. Pour plus d'informations sur ces outils, voir [AWS CLICommand Reference : iot-jobs-data](#) ou [AWSSDK et outils](#).

Les commandes suivantes sont disponibles pour les appareils qui exécutent les tâches. Pour plus d'informations sur l'utilisation des opérations d'API avec le protocole MQTT, reportez-vous [Opérations de l'API MQTT de l'appareil de tâches \(p. 824\)](#) à la section.

GetPendingJobExecutions

Obtient la liste de toutes les tâches qui ne se trouvent pas dans un état terminal, pour un objet déterminé.

HTTPS request

```
GET /things/thingName/jobs
```

Réponse :

```
{  
  "inProgressJobs" : [ JobExecutionSummary ... ],  
  "queuedJobs" : [ JobExecutionSummary ... ]  
}
```

Pour plus d'informations, veuillez consulter [GetPendingJobExecutions](#).

CLI syntax

```
aws iot-jobs-data get-pending-job-executions \  
  --thing-name <value> \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

Format cli-input-json :

```
{  
  "thingName": "string"  
}
```

Pour plus d'informations, veuillez consulter [get-pending-job-executions](#).

StartNextPendingJobExecution

Obtient et démarre l'exécution de tâche en attente suivante pour un objet (avec un statut IN_PROGRESS ou QUEUED).

- Toutes les exécutions de tâches comportant un 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 statut passe à IN_PROGRESS et les détails de l'état de l'exécution de la tâche sont définis comme spécifié.
- Si la prochaine exécution de tâche en attente est déjà en cours IN_PROGRESS, les détails de son statut ne changent pas.
- Si aucune exécution de tâche n'est en attente, la réponse n'inclut pas le execution champ.
- Vous pouvez éventuellement créer un chronomètre en définissant une valeur pour la stepTimeoutInMinutes propriété. 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.

HTTPS request

L'exemple suivant illustre la syntaxe de la requête :

```
PUT /things/thingName/jobs/$next
{
  "statusDetails": {
    "string": "string"
    ...
  },
  "stepTimeoutInMinutes": long
}
```

Pour plus d'informations, veuillez consulter [StartNextPendingJobExecution](#).

CLI syntax

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
}
```

Pour plus d'informations, veuillez consulter [start-next-pending-job-execution](#).

DescribeJobExecution

Permet d'obtenir des informations détaillées sur une exécution de tâche.

Vous pouvez définir le `jobId` pour `$next` renvoyer l'exécution de tâche en attente suivante pour un objet. Le statut de l'exécution de tâche doit être `QUEUED` ou `IN_PROGRESS`.

HTTPS request

Requête :

```
GET /things/thingName/jobs/jobId?
executionNumber=executionNumber&includeJobDocument=includeJobDocument
```

Réponse :

```
{
  "execution" : JobExecution,
}
```

Pour plus d'informations, veuillez consulter [DescribeJobExecution](#).

CLI syntax

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
}
```

Pour plus d'informations, veuillez consulter [describe-job-execution](#).

UpdateJobExecution

Met à jour le statut d'une exécution de tâche. Vous pouvez éventuellement créer un chronomètre en définissant une valeur pour la `stepTimeoutInMinutes` propriété. 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.

HTTPS request

Requête :

```
POST /things/thingName/jobs/jobId
{
  "status": "job-execution-state",
  "statusDetails": {
    "string": "string"
    ...
  },
  "expectedVersion": "number",
  "includeJobExecutionState": boolean,
  "includeJobDocument": boolean,
  "stepTimeoutInMinutes": long,
  "executionNumber": long
}
```

Pour plus d'informations, veuillez consulter [UpdateJobExecution](#).

CLI syntax

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
}
```

Pour plus d'informations, veuillez consulter [update-job-execution](#).

Sécurisation des utilisateurs et des appareils avec AWS IoT Jobs

Pour autoriser les utilisateurs à utiliser AWS IoT Jobs sur leurs appareils, vous devez leur accorder des autorisations en utilisant les politiques IAM. Les appareils doivent ensuite être autorisés à l'aide de AWS IoT Core politiques permettant de se connecter en toute sécurité AWS IoT, de recevoir des exécutions de tâches et de mettre à jour l'état d'exécution.

Type de politique requis pour AWS IoT Jobs

Le tableau suivant présente les différents types de politiques que vous devez utiliser pour l'autorisation. Pour plus d'informations sur la stratégie requise à utiliser, veuillez consulter [Autorisation \(p. 355\)](#).

Type de politique requis

Cas d'utilisation	Protocole	Authentification	Plan de contrôle/plan de données	Type d'identité	Type de politique requis
Autoriser un administrateur, un opérateur ou un service cloud à travailler en toute sécurité avec Jobs	HTTPS	AWSAuthentification de signature version 4 (port 443)	Plan de données et plan de contrôle	Amazon Cognito Identity, IAM ou utilisateur fédéré	Politique IAM
Autorisez votre appareil IoT à fonctionner en	MQTT/ HTTPS	Authentification mutuelle TCP ou TLS (port 8883 ou 443)	Plan de données	Certificats X.509	Stratégie AWS IoT Core

Cas d'utilisation	Protocole	Authentification	Plan de contrôle/plan de données	Type d'identité	Type de politique requis
toute sécurité avec Jobs					

Pour autoriser les opérations AWS IoT Jobs pouvant être effectuées à la fois sur le plan de contrôle et sur le plan de données, vous devez utiliser les politiques IAM. Les identités doivent avoir été authentifiées AWS IoT pour effectuer ces opérations, qui doivent être [identités Amazon Cognito \(p. 342\)](#) ou [Utilisateurs, groupes et rôles IAM \(p. 342\)](#). Pour de plus amples informations sur l'authentification, veuillez consulter [Authentification \(p. 316\)](#).

Les appareils doivent désormais être autorisés sur le plan de données en utilisant des AWS IoT Core politiques pour se connecter en toute sécurité à la passerelle des appareils. La passerelle permet aux appareils de communiquer en toute sécurité avec des tâches AWS IoT, de recevoir des exécutions de tâches et de mettre à jour l'état d'exécution des tâches. La communication entre appareils est sécurisée à l'aide de protocoles sécurisés [MQTT \(p. 92\)](#) ou [HTTPS \(p. 112\)](#) de communication. Ces protocoles utilisent [Certificats client X.509 \(p. 320\)](#) ceux fournis par AWS IoT pour authentifier les connexions des appareils.

La section suivante montre comment autoriser vos utilisateurs, vos services cloud et vos appareils à utiliser AWS IoT Jobs. Pour plus d'informations sur les opérations de l'API du plan de contrôle et du plan de données, consultez [AWS IoT Tâches, opérations d'API \(p. 808\)](#).

Rubriques

- [Autoriser les utilisateurs et les services cloud à utiliser Jobs AWS IoT \(p. 834\)](#)
- [Autoriser vos appareils à utiliser AWS IoT Jobs en toute sécurité sur le plan de données \(p. 842\)](#)

Autoriser les utilisateurs et les services cloud à utiliser Jobs AWS IoT

Pour autoriser vos utilisateurs et vos services cloud, vous devez utiliser les politiques IAM à la fois sur le plan de contrôle et sur le plan de données. Les politiques doivent être utilisées avec le protocole HTTPS et doivent utiliser l'authentification AWS Signature Version 4 (port 443) pour authentifier les utilisateurs.

Note

AWS IoT Core les politiques ne doivent pas être utilisées sur le plan de contrôle. Seules les politiques IAM sont utilisées pour autoriser les utilisateurs ou les services cloud. Pour plus d'informations sur l'utilisation du type de stratégie requis, veuillez consulter [Type de politique requis pour AWS IoT Jobs \(p. 833\)](#).

Les politiques IAM sont des documents JSON qui contiennent des déclarations de politique. Les déclarations de stratégie utilisent des éléments Efficacité, Action et ressources pour spécifier des ressources, les actions autorisées ou refusées et les conditions dans lesquelles les actions sont autorisées ou refusées. Pour de plus amples informations, veuillez consulter [Références des éléments de stratégie JSON IAM](#) dans le Guide de l'utilisateur IAM.

Warning

Nous vous recommandons de ne pas utiliser d'autorisations génériques, comme "Action": ["iot:*"] dans vos politiques ou AWS IoT Core politiques IAM. L'utilisation d'autorisations génériques n'est pas une bonne pratique de sécurité recommandée. Pour plus d'informations, veuillez consulter [AWS IoT Politiques trop permissives \(p. 1042\)](#).

Politiques IAM sur le plan de contrôle

Sur le plan de contrôle, les politiques IAM utilisent le `iot:` préfixe avec l'action pour autoriser le fonctionnement de l'API des tâches correspondantes. Par exemple, l'action `iot:CreateJob` de politique accorde à l'utilisateur l'autorisation d'utiliser l'[CreateJob API](#).

Actions de politique

Le tableau suivant contient une liste des actions de stratégie IAM et les autorisations d'utiliser les actions d'API. Pour plus d'informations sur les types de [ressources](#), voir [Types de ressources définis par AWS IoT](#). Pour plus d'informations sur AWS IoT les actions, consultez la section [Actions définies par AWS IoT](#).

Actions de politique IAM sur le plan de contrôle

Action politique	Opération API	Types de ressources	Description
<code>iot:AssociateTargets</code>	AssociateTargetsWithJob	Targets WithJob tâche	Représente l'autorisation d'associer un groupe à une tâche continue. L' <code>iot:AssociateTargetsWithJob</code> autorisation est vérifiée chaque fois qu'une demande est faite pour associer des cibles.
<code>iot:CancelJob</code>	CancelJob	tâche	Représente l'autorisation d'annuler une tâche. L' <code>iot:CancelJob</code> autorisation est vérifiée chaque fois qu'une demande d'annulation d'une tâche est faite.
<code>iot:CancelJobExecution</code>	CancelJobExecution	tâche thing	Représente l'autorisation d'annuler l'exécution d'une tâche. L' <code>iot:CancelJobExecution</code> autorisation est vérifiée chaque fois qu'une demande d'annulation de l'exécution d'une tâche est faite.
<code>iot>CreateJob</code>	CreateJob	tâche thing groupe d'objets modèle de travail	Représente l'autorisation de créer une tâche. L' <code>iot:CreateJob</code> autorisation est vérifiée chaque fois qu'une demande est faite pour créer une tâche.
<code>iot>CreateJobTemplate</code>	CreateJobTemplate	tâche modèle de travail	Représente l'autorisation de créer un modèle de tâche. L' <code>iot:CreateJobTemplate</code> autorisation est vérifiée chaque fois qu'une demande est faite pour créer un modèle de tâche.
<code>iot>DeleteJob</code>	DeleteJob	tâche	Représente l'autorisation de supprimer une tâche. L' <code>iot:DeleteJob</code> autorisation est vérifiée chaque fois qu'une demande de suppression d'une tâche est faite.
<code>iot>DeleteJobTemplate</code>	DeleteJobTemplate	modèle de travail	Représente l'autorisation de supprimer un modèle de tâche. L' <code>iot:DeleteJobTemplate</code> autorisation est vérifiée

Action politique	Opération API	Types de ressources	Description
			chaque fois qu'une demande de suppression d'un modèle de tâche est faite.
iot:DeleteJobExecution	DeleteJobTemplate	tâche • thing	Représente l'autorisation de supprimer l'exécution d'une. L'iot: DeleteJobExecution autorisation est vérifiée chaque fois qu'une demande est faite pour supprimer l'exécution d'une tâche.
iot:DescribeJob	DescribeJob	tâche	Représente l'autorisation de décrire une tâche. L'iot: DescribeJob autorisation est vérifiée chaque fois qu'une demande est faite pour décrire une tâche.
iot:DescribeJobExecution	DescribeJobExecution	tâche • thing	Représente l'autorisation de décrire l'exécution d'une. L'iot: DescribeJobExecution autorisation est vérifiée chaque fois qu'une demande est faite pour décrire l'exécution d'une tâche.
iot:DescribeJobTemplate	DescribeJobTemplate	modèle de travail	Représente l'autorisation de décrire un modèle de tâche. L'iot: DescribeJobTemplate autorisation est vérifiée chaque fois qu'une demande est faite pour décrire un modèle de tâche.
iot:DescribeManagedJobTemplate	DescribeManagedJobTemplate	modèle de travail	Représente l'autorisation de décrire un modèle de tâche géré. L'iot: DescribeManagedJobTemplate autorisation est vérifiée chaque fois qu'une demande est faite pour décrire un modèle de tâche géré.
iot:GetJobDocument	GetJobDocument	tâche	Représente l'autorisation d'obtenir le document de travail correspondant à une tâche. L'iot: GetJobDocument autorisation est vérifiée chaque fois qu'une demande est faite pour obtenir un document de travail.
iot>ListJobExecutions	ListJobExecutions	tâche	Représente l'autorisation de répertorier les exécutions de une tâche. L'iot: ListJobExecutions autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier les exécutions de une tâche.
iot>ListJobExecutionsForThing	ListJobExecutionsForThing	thing	Représente l'autorisation de répertorier les exécutions de une tâche. L'iot: ListJobExecutionsForThing autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier les exécutions de tâches pour un objet.
iot>ListJobs	ListJobs	Aucun(e)	Représente l'autorisation de répertorier les tâches. L'iot: ListJobs autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier les tâches.

Action politique	Opération API	Types de ressources	Description
iot:ListJobTemplate	ListJobTemplates	Aucun(e)	Représente l'autorisation de répertorier les modèles de tâche. L'iot:ListJobTemplates autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier les modèles de tâches.
iot:ListManagedJobTemplate	ListManagedJobTemplates	Aucuns(e)	Représente l'autorisation de répertorier les modèles de tâches gérés. L'iot:ListManagedJobTemplates autorisation est vérifiée chaque fois qu'une demande est faite pour répertorier les modèles de tâches gérés.
iot:UpdateJob	UpdateJob	tâche	Représente l'autorisation de mettre à jour une tâche. L'iot:UpdateJob autorisation est vérifiée chaque fois qu'une demande de mise à jour d'une tâche est faite.
iot:TagResource	TagResource	<ul style="list-style-type: none"> • tâche • modèle de travail • thing 	Accorde l'autorisation de baliser une ressource.
iot:UntagResource	UntagResource	<ul style="list-style-type: none"> • tâche • modèle de travail • thing 	Accorde l'autorisation d'annuler le balisage d'une ressource

Exemple de politique IAM de base

L'exemple suivant illustre une politique IAM qui autorise l'utilisateur à effectuer les actions suivantes pour votre objet et groupe d'objets IoT.

Dans l'exemple, remplacez :

- *région avec la* votreRégion AWS, par exemple us-east-1.
- *identifiant de compte* avec votre Compte AWS numéro, tel que 57EXAMPLE833
- *thing-group-name* avec le nom du groupe d'objets IoT pour lequel vous ciblez des offres d'emploi, par exemple FirmwareUpdateGroup.
- *nom-objet* avec le nom de votre objet IoT pour lequel vous ciblez des offres d'emploi, par exemple MyIoTThing

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iot:CreateJobTemplate",
                "iot:CreateJob",
            ],
        }
    ]
}
```

```

        "Effect": "Allow",
        "Resource": "arn:aws:iot:region:account-id:thinggroup/thing-group-name"
    },
    {
        "Action": [
            "iot:DescribeJob",
            "iot:CancelJob",
            "iot:DeleteJob",
        ],
        "Effect": "Allow",
        "Resource": "arn:aws:iot:region:account-id:job/*"
    },
    {
        "Action": [
            "iot:DescribeJobExecution",
            "iot:CancelJobExecution",
            "iot:DeleteJobExecution",
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:region:account-id:thing/thing-name"
            "arn:aws:iot:region:account-id:job/*"
        ]
    }
]
}

```

Exemple de politique IAM pour une autorisation basée sur l'adresse IP

Vous pouvez empêcher les principaux d'effectuer des appels d'API vers le point de terminaison de votre plan de contrôle à partir d'adresses IP spécifiques. Pour spécifier les adresses IP qui peuvent être autorisées, dans l'élément Condition de votre politique IAM, utilisez la clé de condition [aws:SourceIp](#) globale.

L'utilisation de cette clé de condition peut également empêcher d'autres Service AWS personnes d'effectuer ces appels d'API en votre nom, par exemple AWS CloudFormation. Pour autoriser l'accès à ces services, utilisez la clé de condition [aws:ViaAWSService](#) globale avec la SourceIp clé aws :. Cela permet de s'assurer que la restriction d'accès à l'adresse IP source s'applique uniquement aux requêtes faites directement par un mandataire. Pour plus d'informations, voir [AWS: Refuser l'accès à AWS en fonction de l'adresse IP source](#).

L'exemple suivant montre comment autoriser uniquement une adresse IP spécifique pouvant effectuer des appels d'API vers le point de terminaison du plan de contrôle. La aws:ViaAWSService clé est définie sur true, ce qui permet à d'autres services d'effectuer des appels d'API en votre nom.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot>CreateJobTemplate",
                "iot>CreateJob"
            ],
            "Resource": ["*"],
            "Condition": {
                "IpAddress": {
                    "aws:SourceIp": "123.45.167.89"
                }
            },
            "Bool": {"aws:ViaAWSService": "true"}
        }
    ]
}

```

```
    ],  
}
```

Politiques IAM sur le plan de données

Les politiques IAM du plan de données utilisent le `iotjobsdata:` préfixe pour autoriser les tâches, les opérations d'API que les utilisateurs peuvent effectuer. Sur le plan de données, vous pouvez accorder à un utilisateur l'autorisation d'utiliser l'[DescribeJobExecution API](#) à l'aide de l'action `iotjobsdata:DescribeJobExecution` de politique.

Warning

L'utilisation des politiques IAM sur le plan de données n'est pas recommandée lorsque vous ciblez AWS IoT des jobs pour vos appareils. Nous vous recommandons d'utiliser les politiques IAM sur le plan de contrôle pour que les utilisateurs puissent créer et gérer des tâches. Sur le plan de données, pour autoriser les appareils à récupérer les exécutions de tâches et à mettre à jour l'état d'exécution, utilisez [AWS IoT Core politiques pour le protocole HTTPS \(p. 843\)](#).

Exemple de politique IAM de base

Les opérations d'API qui doivent être autorisées sont généralement effectuées en saisissant des commandes CLI. L'exemple suivant montre un utilisateur qui exécute une `DescribeJobExecution` opération.

Dans l'exemple, remplacez :

- *région avec la* votre Région AWS, par exemple `us-east-1`.
- *identifiant de compte* avec votre Compte AWS numéro, tel que `57EXAMPLE833`
- *nom-objet* avec le nom de votre objet IoT pour lequel vous ciblez des offres d'emploi, par exemple `myRegisteredThing`
- *job-id* est l'identifiant unique de la tâche ciblée à l'aide de l'API.

```
aws iot-jobs-data describe-job-execution \  
  --endpoint-url "https://account-id.jobs.iot.region.amazonaws.com" \  
  --job-id jobID --thing-name thing-name
```

Voici un exemple de politique IAM qui autorise cette action :

```
{  
  "Version": "2012-10-17",  
  "Statement":  
  {  
    "Action": ["iotjobsdata:DescribeJobExecution"],  
    "Effect": "Allow",  
    "Resource": "arn:aws:iot:region:account-id:thing/thing-name",  
  }  
}
```

Exemples de politiques IAM pour l'autorisation basée sur l'adresse IP

Vous pouvez empêcher les principaux d'effectuer des appels d'API vers le point de terminaison de votre plan de données à partir d'adresses IP spécifiques. Pour spécifier les adresses IP qui peuvent être autorisées, dans l'élément Condition de votre politique IAM, utilisez la clé de condition [aws:SourceIp](#) globale.

L'utilisation de cette clé de condition peut également empêcher d'autres Service AWS personnes d'effectuer ces appels d'API en votre nom, par exemple AWS CloudFormation. Pour autoriser l'accès à ces services, utilisez la clé de condition [aws:ViaAWSService](#) globale avec la clé de aws:SourceIp condition. Cela garantit que la restriction d'accès à l'adresse IP ne s'applique qu'aux demandes directement effectuées par le principal. Pour plus d'informations, voir [AWS: Refuser l'accès à AWS en fonction de l'adresse IP source](#).

L'exemple suivant montre comment autoriser uniquement une adresse IP spécifique pouvant effectuer des appels d'API vers le point de terminaison du plan de données.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["iotjobsdata:*"],  
            "Resource": ["*"],  
            "Condition": {  
                "IpAddress": {  
                    "aws:SourceIp": "123.45.167.89"  
                }  
            },  
            "Bool": {"aws:ViaAWSService": "false"}  
        },  
    ],  
}
```

L'exemple suivant montre comment empêcher des adresses IP ou des plages d'adresses spécifiques d'effectuer des appels d'API vers le point de terminaison du plan de données.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Deny",  
            "Action": ["iotjobsdata:*"],  
            "Condition": {  
                "IpAddress": {  
                    "aws:SourceIp": [  
                        "123.45.167.89",  
                        "192.0.2.0/24",  
                        "203.0.113.0/24"  
                    ]  
                }  
            },  
            "Resource": ["*"],  
        },  
    ],  
}
```

Exemple de politique IAM pour le plan de contrôle et le plan de données

Si vous effectuez une opération d'API à la fois sur le plan de contrôle et sur le plan de données, l'action de stratégie de votre plan de contrôle doit utiliser le iot: préfixe et l'action de stratégie du plan de données doit utiliser le iotjobsdata: préfixe.

Par exemple, l'`DescribeJobExecution` API peut être utilisée à la fois dans le plan de contrôle et dans le plan de données. Sur le plan de contrôle, l'[DescribeJobExecution](#) API est utilisée pour décrire l'exécution d'une tâche. Sur le plan de données, l'[DescribeJobExecution](#) API est utilisée pour obtenir les détails de l'exécution d'une tâche.

La politique IAM suivante autorise un utilisateur à utiliser l'DescribeJobExecutionAPI à la fois sur le plan de contrôle et sur le plan de données.

Dans l'exemple, remplacez :

- *région avec la* votreRégion AWS, par exempleus-east-1.
- *identifiant de compte* avec votre Compte AWS numéro, tel que. 57EXAMPLE833
- *nom-objet* avec le nom de votre objet IoT pour lequel vous ciblez des offres d'emploi, par exemple. MyIoTThing

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Action": ["iotjobsdata:DescribeJobExecution"],  
            "Effect": "Allow",  
            "Resource": "arn:aws:iot:region:account-id:thing/thing-name"  
        },  
        {  
            "Action": [  
                "iot:DescribeJobExecution",  
                "iot:CancelJobExecution",  
                "iot:DeleteJobExecution",  
            ],  
            "Effect": "Allow",  
            "Resource": [  
                "arn:aws:iot:region:account-id:thing/thing-name"  
                "arn:aws:iot:region:account-id:job/*"  
            ]  
        }  
    ]  
}
```

Autoriser le balisage des ressources IoT

Pour mieux contrôler les tâches et les modèles de tâches que vous pouvez créer, modifier ou utiliser, vous pouvez associer des balises aux tâches ou aux modèles de tâches. Les étiquettes vous aident également à identifier la propriété, à attribuer et à répartir les coûts en les plaçant dans des groupes de facturation et en y attachant des balises.

Lorsque vous souhaitez baliser vos tâches ou modèles de tâches que vous avez créés à l'aide du AWS Management Console ou du AWS CLI, votre politique IAM doit autoriser ces balises. Pour accorder des autorisations, votre politique IAM doit utiliser l'iot:TagResourceaction.

Pour obtenir des informations générales sur l'étiquetage de vos ressources, veuillez consulter[Balisage de vos ressources AWS IoT \(p. 310\)](#).

Exemple de politique IAM

Pour un exemple qui montre comment accorder des autorisations de balisage, prenons l'exemple d'un utilisateur qui exécute la commande suivante pour créer une tâche et l'étiqueter dans un environnement spécifique.

Dans l'exemple, remplacez :

- *région avec la* votreRégion AWS, par exempleus-east-1.
- *identifiant de compte* avec votre Compte AWS numéro, tel que. 57EXAMPLE833
- *nom-objet* avec le nom de votre objet IoT pour lequel vous ciblez des offres d'emploi, par exemple. MyIoTThing

```
aws iot create-job
--job-id test_job
--targets "arn:aws:iot:region:account-id:thing/thingOne"
--document-source "https://s3.amazonaws.com/my-s3-bucket/job-document.json"
--description "test job description"
--tags Key=environment,Value=beta
```

Pour cet exemple, vous devez utiliser la stratégie IAM suivante :

```
{
    "Version": "2012-10-17",
    "Statement":
    {
        "Action": [ "iot:CreateJob", "iot:CreateJobTemplate", "iot:TagResource" ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:aws-region:account-id:job/*",
            "arn:aws:iot:aws-region:account-id:jobtemplate/*"
        ]
    }
}
```

Autoriser vos appareils à utiliser AWS IoT Jobs en toute sécurité sur le plan de données

Pour autoriser vos appareils à interagir en toute sécurité avec AWS IoT Jobs sur le plan de données, vous devez utiliser des AWS IoT Core politiques. AWS IoT Core les politiques relatives aux tâches sont des documents JSON contenant des déclarations de politique. Ces politiques utilisent également les éléments Effet, Action et Ressource et suivent une convention similaire à celle des politiques IAM. Pour plus d'informations sur les éléments, consultez la section [Référence des éléments de politique IAM JSON](#) dans le Guide de l'utilisateur d'IAM.

Les politiques peuvent être utilisées avec les protocoles MQTT et HTTPS et doivent utiliser l'authentification mutuelle TCP ou TLS pour authentifier les appareils. La section suivante explique comment utiliser ces stratégies dans les différents protocoles de communication.

Warning

Nous vous recommandons de ne pas utiliser d'autorisations génériques, comme "Action": ["iot:*"] dans vos politiques ou AWS IoT Core politiques IAM. L'utilisation d'autorisations génériques n'est pas une bonne pratique de sécurité recommandée. Pour plus d'informations, veuillez consulter [AWS IoT politiques trop permissives \(p. 1042\)](#).

AWS IoT Core politiques pour le protocole MQTT

AWS IoT Core les politiques du protocole MQTT vous autorisent à utiliser les actions de l'API MQTT du périphérique de tâches. Les opérations de l'API MQTT sont utilisées pour travailler avec des rubriques MQTT réservées aux commandes de tâches. Pour plus d'informations sur ces opérations d'API, veuillez consulter [Opérations de l'API MQTT de l'appareil de tâches \(p. 824\)](#).

Les politiques MQTT utilisent des actions politiques telles que `iot:Connect`, `iot:Publish`, `iot:Subscribe`, et `iot:Receive` pour travailler avec les sujets liés aux emplois. Ces politiques vous permettent de vous connecter au courtier de messages, de vous abonner aux rubriques MQTT relatives aux jobs et d'envoyer et de recevoir des messages MQTT entre vos appareils et le cloud. Pour plus d'informations sur ces actions, consultez [Actions de stratégie AWS IoT Core \(p. 358\)](#).

Pour plus d'informations sur les rubriques relatives aux AWS IoT jobs, consultez[Rubriques de tâche \(p. 124\)](#).

Exemple de politique MQTT de base

L'exemple suivant montre comment vous pouvez utiliser, publier `iot:Publish` et vous abonner `iot:Subscribe` à des tâches et à des exécutions de tâches.

Dans l'exemple, remplacez :

- *région avec la* votreRégion AWS, par exempleus-east-1.
- *identifiant de compte* avec votre Compte AWS numéro, tel que. 57EXAMPLE833
- *nom-objet* avec le nom de votre objet IoT pour lequel vous ciblez des offres d'emploi, par exemple. MyIoTThing

```
{  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish",  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iot:region:account-id:topic/$aws/events/job/*",  
                "arn:aws:iot:region:account-id:topic/$aws/events/jobExecution/*",  
                "arn:aws:iot:region:account-id:topic/$aws/things/thing-name/jobs/*"  
            ]  
        }  
    ],  
    "Version": "2012-10-17"  
}
```

AWS IoT Core politiques pour le protocole HTTPS

AWS IoT Core les politiques du plan de données peuvent également utiliser le protocole HTTPS avec le mécanisme d'authentification TLS pour autoriser vos appareils. Sur le plan des données, les politiques utilisent le `iot:jobsdata` : préfixe pour autoriser les tâches, les opérations d'API que vos appareils peuvent effectuer. Par exemple, l'action `iot:jobsdata:DescribeJobExecution` de politique accorde à l'utilisateur l'autorisation d'utiliser l'[DescribeJobExecution API](#).

Note

Les actions de politique du plan de données doivent utiliser le `iot:jobsdata:` préfixe. Sur le plan de contrôle, les actions doivent utiliser le `iot:` préfixe. Pour un exemple de politique IAM lorsque des actions de politique du plan de contrôle et du plan de données sont utilisées à la fois, voir[Exemple de politique IAM pour le plan de contrôle et le plan de données \(p. 840\)](#).

Actions de politique

Le tableau suivant présente la liste des actions de AWS IoT Core politique et des autorisations permettant d'autoriser les appareils à utiliser les actions de l'API. Pour obtenir la liste des opérations d'API que vous pouvez effectuer dans le plan de données, veuillez consulter[API HTTP de la tâche \(p. 830\)](#).

Note

Ces actions de stratégie d'exécution des tâches s'appliquent uniquement au point de terminaison HTTP TLS. Si vous utilisez le point de terminaison MQTT, vous devez utiliser les actions de politique MQTT définies précédemment.

AWS IoT Core actions politiques sur le plan de données

Action politique	Opération API	Types de ressource	Description
iotjobsdata:DescribeJobExecution	DescribeJobExecution	tâche	Représente l'autorisation de récupérer l'exécution d'une tâche. L'opération <code>iotjobsdata:DescribeJobExecution</code> est vérifiée chaque fois qu'une demande est faite pour récupérer l'exécution d'une tâche.
iotjobsdata:GetPendingJobExecutions	GetPendingJobExecutions	thing	Représente l'autorisation de récupérer la liste des tâches qui ne sont pas à un statut terminal pour un objet. L'autorisation <code>iotjobsdata:GetPendingJobExecutions</code> est vérifiée chaque fois qu'une demande est faite de récupérer la liste.
iotjobsdata:StartNextPendingJobExecution	StartNextPendingJobExecution	thing	Représente l'autorisation d'obtenir et de démarrer l'exécution de tâche en attente suivante pour un objet. C'est-à-dire pour mettre à jour l'exécution d'une tâche dont le statut <code>QUEUED</code> est <code>IN_PROGRESS</code> . L'autorisation <code>iot:StartNextPendingJobExecution</code> est vérifiée chaque fois qu'une demande est faite de démarrer l'exécution de tâche en attente suivante.
iotjobsdata:UpdateJobExecution	UpdateJobExecution	thing	Représente l'autorisation de mettre à jour une exécution de tâche. L'autorisation <code>iot:UpdateJobExecution</code> est vérifiée chaque fois qu'une demande est faite de mettre à jour l'état d'une exécution de tâche.

Exemple de politique de base

L'exemple suivant montre un exemple de AWS IoT Core politique qui autorise l'exécution des actions sur les opérations de l'API du plan de données pour n'importe quelle ressource. Vous pouvez étendre votre politique à une ressource spécifique, telle qu'un objet IoT. Dans votre exemple, remplacez :

- *région* avec votre Région AWS tel que `us-east-1`.
- *identifiant de compte* avec votre Compte AWS numéro, tel que `57EXAMPLE833`
- *nom-objet* avec le nom de l'objet IoT, tel que `MyIoTthing`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "iotjobsdata:GetPendingJobExecutions",
                "iotjobsdata:StartNextPendingJobExecution",
                "iotjobsdata:UpdateJobExecution"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

```
        "iotjobsdata:DescribeJobExecution",
        "iotjobsdata:UpdateJobExecution"
    ],
    "Effect": "Allow",
    "Resource": "arn:aws:iot:<region>:<account-id>:thing/<thing-name>"  
}  
]  
}
```

Par exemple, vous devez utiliser ces politiques lorsque vos appareils IoT utilisent une AWS IoT Core politique pour accéder à l'une de ces opérations d'API, comme l'exemple suivant d'DescribeJobExecutionAPI :

```
GET /things/<thingName>/jobs/<jobId>?  
executionNumber=<executionNumber>&includeJobDocument=includeJobDocument&namespaceId=<namespaceId>  
HTTP/1.1
```

Limites des tâches

AWS IoTJobs comporte des quotas, ou limites, qui représentent le nombre maximal de ressources ou d'opérations de service pour votre compte AWS.

Limites de tâches actives et simultanées

Cette section vous aidera à en savoir plus sur les tâches actives et simultanées ainsi que sur les limites qui s'y appliquent.

Tâches actives et limite de tâches actives

Lorsque vous créez une tâche à l'aide de la AWS IoT console ou de l'CreateJobAPI, le statut de la tâche devient IN_PROGRESS. Toutes les tâches en cours sont des tâches actives et sont prises en compte dans le calcul de la limite des tâches actives. Cela inclut les tâches qui déploient de nouvelles exécutions de tâches ou les tâches qui attendent que les appareils terminent leur exécution. Cette limite s'applique à la fois aux tâches continues et aux tâches instantanées.

Tâches simultanées et limite de simultanéité des tâches

Les tâches en cours qui déploient de nouvelles exécutions de tâches ou qui annulent des exécutions de tâches précédemment créées sont des tâches simultanées et sont prises en compte dans la limite de simultanéité des tâches. AWS IoT Les tâches peuvent être déployées et annulées rapidement à une cadence de 1 000 appareils par minute. Chaque tâche est concurrent et n'est pas prise en compte dans le calcul de la limite de simultanéité des tâches que pendant une courte période. Une fois que les exécutions des tâches ont été déployées ou annulées, celles-ci ne sont plus simultanées et ne sont pas prises en compte dans le calcul de la limite de simultanéité des tâches. Vous pouvez utiliser la simultanéité des tâches pour créer un grand nombre de tâches en attendant que les appareils terminent l'exécution des tâches.

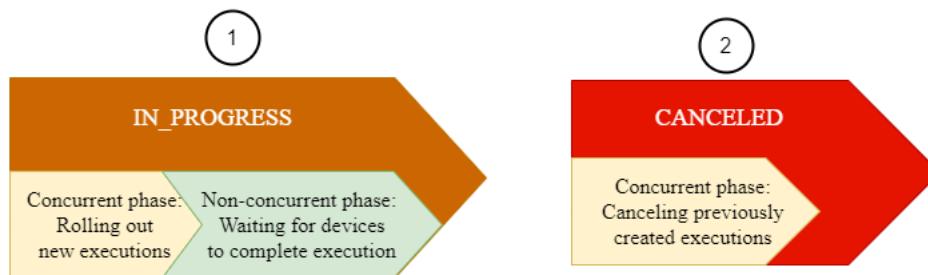
Note

Si une tâche avec la configuration de planification facultative et le déploiement du document de travail prévu pour avoir lieu pendant une fenêtre de maintenance atteint la limite sélectionnée startTime et que vous atteignez votre limite maximale de simultanéité des tâches, alors cette tâche planifiée passera à un état de CANCELED.

Pour déterminer si une tâche est simultanée, vous pouvez utiliser la IsConcurrent propriété d'une tâche depuis la AWS IoT console ou à l'aide de l'ListJobAPI DescribeJob or. Cette limite s'applique à la fois aux tâches continues et aux tâches instantanées.

Pour consulter les tâches actives et les limites de simultanéité des tâches, ainsi que les autres quotas de AWS IoT tâches qui vous concernent Compte AWS et pour demander une augmentation de la limite, consultez la section [Points de terminaison et quotas de gestion des AWS IoT appareils](#) dans le Références générales AWS

Le schéma suivant montre comment la simultanéité des tâches s'applique aux tâches en cours et aux tâches annulées.



Note

Les nouvelles tâches dont l'option est facultative `SchedulingConfig` conserveront leur statut initial `SCHEDULED` et seront mises à jour une `IN_PROGRESS` fois qu'elles auront atteint le statut sélectionné `startTime`. Une fois que la nouvelle tâche avec l'option facultative `SchedulingConfig` atteint la valeur sélectionnée `startTime` et est mise à jour vers `IN_PROGRESS`, elle sera prise en compte dans la limite des tâches actives et la limite de simultanéité des tâches. Les tâches dont l'état est de `SCHEDULED` seront prises en compte dans la limite des tâches actives, mais ne seront pas prises en compte dans la limite de simultanéité des tâches.

Le tableau suivant indique les limites qui s'appliquent aux tâches actives et simultanées ainsi qu'aux phases concurrentes et non concurrentes des états des tâches.

Limites de tâches actives et simultanées

Job status	Phase	Limite de tâches actives	Limite de simultanéité des Job
<code>SCHEDULED</code>	Phase non simultanée : AWS IoT Jobs attend que la date prévue <code>startTime</code> de la tâche commence à envoyer des notifications d'exécution sur vos appareils. Les tâches de cette phase ne sont prises en compte que dans la limite des tâches actives et leur <code>IsConcurrent</code> propriété sera définie sur <code>false</code> .	S'applique	Ne s'applique pas
<code>IN_PROGRESS</code>	Phase simultanée : AWS IoT Jobs accepte la demande de création du travail et commence à envoyer des notifications d'exécution du travail sur vos appareils. Les tâches de cette phase sont concurrentes, comme l'indique la <code>IsConcurrent</code> propriété définie sur <code>true</code> , et sont prises en compte à la fois dans les tâches actives et dans les limites de simultanéité des tâches.	S'applique	S'applique

Job status	Phase	Limite de tâches actives	Limite de simultanéité des Job
	Phase non simultanée : les AWS IoT tâches attendent que les appareils signalent les résultats de leurs exécutions de tâches. Les tâches de cette phase ne sont prises en compte que dans la limite des tâches actives et leur IsConcurrent propriété sera définie sur false.	S'applique	Ne s'applique pas
Canceled	Phase simultanée : AWS IoT Jobs accepte la demande d'annulation de la tâche et commence à annuler les exécutions de tâches précédemment créées pour vos appareils. Les tâches de cette phase sont concurrentes et leur IsConcurrent propriété sera définie sur true. Une fois que la tâche et son exécution ont été annulées, la tâche n'est plus simultanée et n'est pas prise en compte dans le calcul de la limite de simultanéité des tâches.	Ne s'applique pas	S'applique

Note

La durée maximale d'une fenêtre de maintenance récurrente est de 23 heures et 50 minutes.

Tunneling sécurisé AWS IoT

Lorsque des appareils sont déployés derrière des pare-feux restreints sur des sites distants, vous avez besoin d'un moyen d'accéder à ces appareils pour le dépannage, les mises à jour de configuration et d'autres tâches opérationnelles. Utilisez le tunneling sécurisé pour établir une communication bidirectionnelle avec des appareils distants via une connexion sécurisée gérée par AWS IoT. Le tunneling sécurisé ne nécessite pas de mise à jour de vos règles de pare-feu entrant existantes. 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.

Qu'est-ce que le tunneling sécurisé ?

Utilisez un tunnel sécurisé pour accéder aux appareils déployés derrière des pare-feux à ports restreints sur des sites distants. Vous pouvez vous connecter au périphérique de destination à partir de votre ordinateur portable ou de bureau en tant que périphérique source à l'aide du AWS Cloud. La source et la destination communiquent à l'aide d'un proxy local open source qui s'exécute sur chaque appareil. Le proxy local communique avec le AWS Cloud via un port ouvert autorisé par le pare-feu, généralement 443. Les données transmises via le tunnel sont cryptées à l'aide du protocole TLS (Transported Layer Security).

Rubriques

- [Concepts de tunneling sécurisés \(p. 848\)](#)
- [Comment fonctionne le tunneling sécurisé \(p. 849\)](#)
- [Cycle de vie des tunnels sécurisés \(p. 850\)](#)

Concepts de tunneling sécurisés

Les termes suivants sont utilisés par le tunneling sécurisé lors de l'établissement d'une communication avec des appareils distants. Pour plus amples informations sur le fonctionnement du tunneling sécurisé, veuillez consulter [Comment fonctionne le tunneling sécurisé \(p. 849\)](#).

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. Le CAT ne peut être utilisé qu'une seule fois pour se connecter au tunnel. Pour vous reconnecter au tunnel, faites pivoter les jetons d'accès client à l'aide de l'opération [RotateTunnelAccessToken](#) API ou de la commande [rotate-tunnel-access-token](#) CLI.

Jeton client

Valeur unique générée par le client que AWS IoT Secure Tunneling peut utiliser pour toutes les tentatives de connexion ultérieures au même tunnel. Ce champ est facultatif. Si le jeton client n'est pas fourni, le jeton d'accès client (CAT) ne peut être utilisé qu'une seule fois pour le même tunnel. Les tentatives de connexion suivantes utilisant le même CAT seront rejetées. Pour plus d'informations sur l'utilisation des jetons clients, consultez l'[implémentation de référence du proxy local dans GitHub](#).

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 de l'appareil

Application IoT qui se connecte à la passerelle d'appareils AWS IoT et écoute les nouvelles notifications de tunnel via MQTT. Pour plus d'informations, veuillez consulter [Extrait de l'agent IoT \(p. 883\)](#).

Proxy local

Proxy logiciel qui s'exécute sur les appareils source et de destination et qui relaie un flux de données entre le tunneling sécurisé et l'application de l'appareil. Le proxy local peut être exécuté en mode source ou en mode destination. Pour plus d'informations, veuillez consulter [Proxy local \(p. 868\)](#).

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.

Comment fonctionne le tunneling sécurisé

Ce qui suit montre comment le tunneling sécurisé établit une connexion entre votre appareil source et votre appareil de destination. Pour plus d'informations sur les différents termes tels que jeton d'accès client (CAT), consultez [Concepts de tunneling sécurisés \(p. 848\)](#).

1. Ouvrir un tunnel

Pour ouvrir un tunnel afin d'ouvrir une session avec votre appareil de destination distant, vous pouvez utiliser la AWS Management Console commande [AWS CLIopen-tunnel](#) ou l'[OpenTunnelAPI](#).

2. Téléchargez la paire de jetons d'accès au client

Après avoir ouvert un tunnel, vous pouvez télécharger le jeton d'accès client (CAT) pour votre source et votre destination et l'enregistrer sur votre appareil source. Vous devez récupérer le CAT et l'enregistrer maintenant avant de démarrer le proxy local.

3. Démarrer le proxy local en mode destination

L'agent IoT qui a été installé et qui s'exécute sur votre appareil de destination sera abonné à la rubrique MQTT réservée \$aws/things/*thing-name*/tunnels/notify et recevra le CAT. Ici, *thing-name* est le nom de l'AWS IoT objet que vous créez pour votre destination. Pour plus d'informations, veuillez consulter [Rubriques liées au tunneling sécurisé \(p. 128\)](#).

L'agent IoT utilise ensuite le CAT pour démarrer le proxy local en mode destination et établir une connexion du côté destination du tunnel. Pour plus d'informations, veuillez consulter [Extrait de l'agent IoT \(p. 883\)](#).

4. Démarrer le proxy local en mode source

Une fois le tunnel ouvert, AWS IoT Device Management fournit le CAT pour la source que vous pouvez télécharger sur le périphérique source. Vous pouvez utiliser le CAT pour démarrer le proxy local en

mode source, qui connecte ensuite le côté source du tunnel. Pour de plus amples informations sur le proxy local, veuillez consulter[Proxy local \(p. 868\)](#).

5. Ouvrez une session SSH

Les deux côtés du tunnel étant connectés, vous pouvez démarrer une session SSH en utilisant le proxy local du côté source.

Pour plus d'informations sur la façon d'utiliser le AWS Management Console pour ouvrir un tunnel et démarrer une session SSH, consultez[Ouvrez un tunnel et démarrez une session SSH vers un appareil distant \(p. 851\)](#).

La vidéo suivante décrit le fonctionnement du tunneling sécurisé et vous explique comment configurer une session SSH sur un périphérique Raspberry Pi.

Cycle de vie des tunnels sécurisés

Les tunnels peuvent avoir le statutOPEN ouCLOSED. Les connexions au tunnel peuvent avoir le statutCONNECTED ouDISCONNECTED. Ce qui suit montre comment fonctionnent les différents états de tunnel et de connexion.

1. Lorsque vous ouvrez un tunnel, son état est OPEN. L'état de connexion source et de destination du tunnel est défini sur DISCONNECTED.
2. Lorsqu'un appareil (source ou destination) se connecte au tunnel, l'état de connexion correspondant passe àCONNECTED.
3. Lorsqu'un appareil se déconnecte du tunnel alors que l'état du tunnel est maintenuOPEN, l'état de connexion correspondant revient àDISCONNECTED. Un appareil peut se connecter à un tunnel et s'en déconnecter à plusieurs reprises tant que l'état du tunnel reste OPEN.

Note

Les jetons d'accès client (CAT) ne peuvent être utilisés qu'une seule fois pour se connecter à un tunnel. Pour vous reconnecter au tunnel, faites pivoter les jetons d'accès client à l'aide de l'opération [RotateTunnelAccessToken](#)API ou de la commande [rotate-tunnel-access-token](#)CLI. Pour obtenir des exemples, consultez[Résolution des problèmes de connectivité par tunnelingAWS IoT sécurisé en alternant les jetons d'accès client \(p. 890\)](#).

4. Lorsque vous appelezCloseTunnel ou que le tunnel reste plusOPEN longtemps que laMaxLifetimeTimeout valeur, le statut du tunnel devientCLOSED. Vous pouvez configurer MaxLifetimeTimeout lorsque vousappelez OpenTunnel1. Si vous ne spécifiez pas de valeur, MaxLifetimeTimeout est défini par défaut sur 12 heures.

Note

Un tunnel ne peut pas être rouvert lorsque son état est CLOSED.

5. Vous pouvez appelerDescribeTunnel etListTunnels consulter les métadonnées du tunnel lorsque celui-ci est visible. Le tunnel peut être visible dans laAWS IoT console pendant au moins trois heures avant d'être supprimé.

AWS IoT tutoriels de tunneling sécurisé

AWS IoT tunneling sécurisé permet aux clients d'établir une communication bidirectionnelle avec des appareils distants protégés par un pare-feu via une connexion sécurisée gérée parAWS IoT.

Pour faire une démonstrationAWS IoT de tunneling [AWS IoT sécurisé, utilisez notre démo de tunneling sécurisé sur GitHub](#).

Les didacticiels suivants vous aideront à découvrir comment démarrer et utiliser le tunneling sécurisé. Vous allez apprendre comment :

1. Créez un tunnel sécurisé à l'aide des méthodes de configuration rapide et de configuration manuelle pour accéder au périphérique distant.
2. Configurez le proxy local lorsque vous utilisez la méthode de configuration manuelle et connectez-vous au tunnel pour accéder au périphérique de destination.
3. Connectez-vous via SSH à l'appareil distant à partir d'un navigateur sans avoir à configurer le proxy local.
4. Convertissez un tunnel créé à l'aide de la méthode de configuration manuelle AWS CLI ou à l'aide de la méthode de configuration rapide.

Didacticiels dans cette section

Les didacticiels de cette section se concentrent sur la création d'un tunnel à l'aide du AWS Management Console et de l'AWS IoT API Reference. Dans la AWS IoT console, vous pouvez créer un tunnel depuis la page du [hub Tunnels](#) ou depuis la page de détails d'un objet que vous avez créé. Pour plus d'informations, veuillez consulter [Méthodes de création de tunnels dans AWS IoT la console \(p. 852\)](#).

Les didacticiels de cette section sont présentés ci-dessous :

- [Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant \(p. 853\)](#)

Ce didacticiel explique comment ouvrir un tunnel à partir de la page [Tunnels hub](#) à l'aide de la méthode de configuration rapide. Vous apprendrez également à utiliser le protocole SSH basé sur un navigateur pour accéder à l'appareil distant à l'aide d'une interface de ligne de commande contextuelle intégrée à la AWS IoT console.

- [Ouvrez un tunnel à l'aide de la configuration manuelle et connectez-vous à un appareil distant \(p. 859\)](#)

Ce didacticiel explique comment ouvrir un tunnel à partir de la page [Tunnels hub](#) à l'aide de la méthode de configuration manuelle. Vous apprendrez également à configurer et à démarrer le proxy local à partir d'un terminal de votre appareil source et à vous connecter au tunnel.

- [Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur \(p. 865\)](#)

Ce didacticiel explique comment ouvrir un tunnel à partir de la page de détails d'un objet que vous avez créé. Vous apprendrez à créer un nouveau tunnel et à utiliser un tunnel existant. Le tunnel existant correspond au tunnel ouvert le plus récent créé pour l'appareil. Vous pouvez également utiliser le SSH d'un navigateur pour accéder à l'appareil distant.

AWS IoT tutoriels de tunneling sécurisé

- [Ouvrez un tunnel et démarrez une session SSH vers un appareil distant \(p. 851\)](#)
- [Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur \(p. 865\)](#)

Ouvrez un tunnel et démarrez une session SSH vers un appareil distant

Dans ces didacticiels, vous allez apprendre à accéder à distance à un appareil protégé par un pare-feu. Vous ne pouvez pas démarrer de session SSH directe sur l'appareil car le pare-feu bloque tout le trafic entrant. Les didacticiels vous montrent comment ouvrir un tunnel, puis utiliser ce tunnel pour démarrer une session SSH sur un appareil distant.

Conditions préalables pour les didacticiels

Les conditions préalables à l'exécution du didacticiel peuvent varier selon que vous utilisez les méthodes de configuration manuelle ou rapide pour ouvrir un tunnel et accéder au périphérique distant.

Note

Pour les deux méthodes de configuration, vous devez autoriser le trafic sortant sur le port 443.

- Pour plus d'informations sur les conditions requises pour le didacticiel relatif à la méthode de configuration rapide, consultez [Prérequis pour la méthode de configuration rapide \(p. 854\)](#).
- Pour plus d'informations sur les conditions requises pour le didacticiel relatif à la méthode de configuration manuelle, consultez [Prérequis pour la méthode de configuration manuelle \(p. 859\)](#). Si vous utilisez cette méthode de configuration, vous devez configurer le proxy local sur votre appareil source. Pour télécharger le code source du proxy local, consultez la section [Implémentation de référence du proxy local sur GitHub](#).

Méthodes de configuration de tunnel

Dans ces didacticiels, vous découvrirez les méthodes de configuration manuelle et rapide permettant d'ouvrir un tunnel et de vous connecter au périphérique distant. Le tableau suivant montre la différence entre les méthodes de configuration. Après avoir créé le tunnel, vous pouvez utiliser une interface de ligne de commande intégrée au navigateur pour accéder au périphérique distant via SSH. Si vous égarez les jetons ou si le tunnel est déconnecté, vous pouvez envoyer de nouveaux jetons d'accès pour vous reconnecter au tunnel.

Méthodes de configuration rapides et manuelles

Critères	Configuration rapide	Configuration manuelle
Création de tunnel	Créez un nouveau tunnel avec des configurations modifiables par défaut. Pour accéder à votre appareil distant, vous ne pouvez utiliser SSH que comme service de destination.	Créez un tunnel en spécifiant manuellement les configurations du tunnel. Vous pouvez utiliser cette méthode pour vous connecter au périphérique distant à l'aide de services autres que SSH.
Jetons d'identification	Le jeton d'accès à la destination sera automatiquement envoyé à votre appareil sur la rubrique MQTT réservée , si un nom d'objet est spécifié lors de la création du tunnel. Vous n'êtes pas obligé de télécharger ou de gérer le jeton sur votre appareil source.	Vous devrez télécharger et gérer manuellement le jeton sur votre appareil source. Le jeton d'accès à la destination est automatiquement transmis au périphérique distant sur la rubrique MQTT réservée , si un nom d'objet est spécifié lors de la création du tunnel.
Proxy local	Un proxy local basé sur le Web est automatiquement configuré pour que vous puissiez interagir avec l'appareil. Vous n'avez pas besoin de configurer manuellement le proxy local.	Vous devrez configurer et lancer manuellement le proxy local. Pour configurer le proxy local, vous pouvez utiliser le AWS IoT Device Client ou télécharger l' implémentation de référence du proxy local sur GitHub .

Méthodes de création de tunnels dans AWS IoT la console

Les didacticiels de cette section vous montrent comment créer un tunnel à l'aide de l'[OpenTunnel API](#) AWS Management Console et. Si vous configurez la destination lors de la création d'un tunnel, le tunneling AWS

IoT sécurisé fournit le jeton d'accès au client de destination au périphérique distant via MQTT et la rubrique MQTT réservée \$aws/things/RemoteDeviceA/tunnels/notify). À la réception du message MQTT, l'agent IoT du périphérique distant démarre le proxy local en mode destination. Pour plus d'informations, veuillez consulter [Rubriques réservées \(p. 117\)](#).

Note

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, veuillez consulter [Configuration d'un appareil distant et utilisation d'un agent IoT \(p. 883\)](#).

Dans la AWS IoT console, vous pouvez créer un tunnel à l'aide de l'une des méthodes suivantes. Pour plus d'informations sur les didacticiels qui vous aideront à créer un tunnel à l'aide de ces méthodes, reportez-vous à la section [Didacticiels dans cette section \(p. 851\)](#).

- [Centre de tunnels](#)

Lorsque vous créez le tunnel, vous pouvez spécifier si vous souhaitez utiliser la méthode de configuration rapide ou la méthode de configuration manuelle pour créer le tunnel et fournir les détails facultatifs de configuration du tunnel. Les détails de configuration incluent également le nom de l'appareil de destination et le service que vous souhaitez utiliser pour vous connecter à l'appareil. Après avoir créé un tunnel, vous pouvez utiliser le protocole SSH dans le navigateur ou ouvrir un terminal en dehors de la AWS IoT console pour accéder à votre appareil distant.

- [Page de détails de l'objet](#)

Lorsque vous créez le tunnel, vous pouvez également spécifier si vous souhaitez utiliser le tunnel ouvert le plus récent ou créer un nouveau tunnel pour l'appareil, en plus de choisir les méthodes de configuration et de fournir les détails facultatifs de configuration du tunnel. Vous ne pouvez pas modifier les détails de configuration d'un tunnel existant. Vous pouvez utiliser la méthode de configuration rapide pour transférer les jetons d'accès et le SSH vers le périphérique distant dans le navigateur. Pour ouvrir un tunnel à l'aide de cette méthode, vous devez avoir créé un objet IoT (par exemple RemoteDeviceA) dans le AWS IoT registre. Pour de plus amples informations, consultez [Enregistrement d'un appareil dans le AWS IoT registre](#).

Didacticiels dans cette section

- [Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant \(p. 853\)](#)
- [Ouvrez un tunnel à l'aide de la configuration manuelle et connectez-vous à un appareil distant \(p. 859\)](#)

Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant

Vous pouvez utiliser la méthode de configuration rapide ou la méthode de configuration manuelle pour créer un tunnel. Ce didacticiel explique comment ouvrir un tunnel à l'aide de la méthode de configuration rapide et utiliser le protocole SSH basé sur un navigateur pour se connecter au périphérique distant. Pour obtenir un exemple qui montre comment d'ouvrir un exemple de lancement de tunnel à l'aide de la méthode de configuration, consultez [Ouvrez un tunnel à l'aide de la configuration manuelle et connectez-vous à un appareil distant \(p. 859\)](#).

À l'aide de la méthode de configuration rapide, vous pouvez créer un nouveau tunnel avec des configurations par défaut modifiables. Un proxy local basé sur le Web est configuré pour vous et le jeton d'accès est automatiquement envoyé à votre appareil de destination distant à l'aide de MQTT. Après avoir créé un tunnel, vous pouvez commencer à interagir avec votre appareil distant à l'aide d'une interface de ligne de commande intégrée à la console.

Avec la méthode de configuration rapide, vous devez utiliser SSH comme service de destination pour accéder au périphérique distant. Pour de plus amples informations sur les différentes méthodes de configuration, consultez [Méthodes de configuration de tunnel \(p. 852\)](#).

Prérequis pour la méthode de configuration rapide

- Les pare-feux derrière lesquels se trouve le périphérique distant doivent autoriser le trafic sortant sur le port 443. Le tunnel que vous créez utilisera ce port pour se connecter au périphérique distant.
- Vous disposez d'un agent d'appareil IoT (voir [Extrait de l'agent IoT \(p. 883\)](#)) exécuté sur l'appareil distant qui se connecte à la passerelle de l'AWS IoTAppareil et qui est configuré avec un abonnement à une rubrique MQTT. Pour plus d'informations, consultez la section [Connecter un appareil à la passerelle de l'AWS IoTAppareil](#).
- Vous devez disposer d'un démon SSH s'exécutant sur l'appareil distant.

Ouvrir un tunnel

Vous pouvez ouvrir un tunnel sécurisé à l'aide du AWS Management Console, de l'AWS IoT API Reference ou du AWS CLI. Vous pouvez éventuellement configurer un nom de destination, mais ce n'est pas obligatoire pour ce didacticiel. Si vous configurez la destination, le tunneling sécurisé fournira automatiquement le jeton d'accès au périphérique distant à l'aide de MQTT. Pour plus d'informations, veuillez consulter [Méthodes de création de tunnels dans AWS IoT la console \(p. 852\)](#).

Pour ouvrir un tunnel à l'aide de la console

1. Accédez au [hub Tunnels de la AWS IoT console](#) et choisissez Créez un tunnel.

The screenshot shows the AWS IoT Management Console interface. In the top navigation bar, the path is: AWS IoT > Manage > Remote actions > Tunnels. Below the navigation, there is a search bar labeled 'Find tunnels'. At the top right of the main area, there are three buttons: 'Close tunnel', 'Delete tunnel', and a prominent orange 'Create tunnel' button. Below these buttons, the text 'Tunnels (0) Info' is displayed. The main content area shows a table with three columns: 'Tunnel ID', 'Tunnel status', and 'Date created'. A message 'No tunnels' is centered in the table, with the subtext 'You don't have any tunnels.' underneath it. At the bottom of the table area, there is a blue 'Create tunnel' button.

2. Pour ce didacticiel, choisissez Configuration rapide comme méthode de création du tunnel, puis choisissez Suivant.

Note

Si vous créez un tunnel sécurisé à partir de la page de détails d'un objet que vous avez créé, vous pouvez choisir de créer un nouveau tunnel ou d'utiliser un tunnel existant. Pour plus d'informations, veuillez consulter [Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur \(p. 865\)](#).

Setup method

Quick setup (SSH)
 Manual setup

Quick setup (SSH)

Use quick setup to create a new tunnel with default, editable tunnel configurations. When you use quick setup:

- A web-based local proxy will be automatically configured for you to SSH into the remote device.
- The destination access token will be automatically delivered to your device on the [reserved MQTT topic](#), if a thing name is specified.

3. Vérifiez et confirmez les détails de configuration de la configuration de tunnel. Pour créer un tunnel, choisissez Confirmer et créer. Si vous souhaitez modifier ces informations, choisissez Précédent pour revenir à la page précédente, puis confirmez et créez le tunnel.

Note

Lorsque vous utilisez la configuration rapide, le nom du service ne peut pas être modifié. Vous devez utiliser SSH en tant que service.

4. Pour créer le tunnel, choisissez OK.

Pour ce didacticiel, vous n'avez pas besoin de charger les jetons d'accès source ou de destination. Ces jetons ne peuvent être utilisés qu'une seule fois pour se connecter au tunnel. Si votre tunnel est déconnecté, vous pouvez générer et envoyer de nouveaux jetons à votre appareil distant pour vous reconnecter au tunnel. Pour plus d'informations, veuillez consulter [Renvoyer les jetons d'accès à un tunnel \(p. 861\)](#).

Tunnel created

Download these tokens to your source and destination devices. If a thing name was specified, the destination access token will be delivered to your connected device on the reserved MQTT topic.

Rotating the access tokens will revoke the current tokens and generate new access tokens. These tokens are single use only. If your tokens get misplaced or your tunnel gets disconnected, you can resend new access tokens to reconnect to the tunnel.

Access token for source
c6255bc0-sourceAccessToken

Access token for destination
c6255bc0-destinationAccessToken

Done

Pour ouvrir un tunnel à l'aide de l'API

Pour ouvrir un nouveau tunnel, vous pouvez utiliser l'opération [OpenTunnelAPI](#).

Note

Vous pouvez créer un tunnel à l'aide de la méthode de configuration rapide uniquement à partir de la AWS IoT console. Lorsque vous utilisez l'AWS IoT API de référence ou le AWS CLI, la méthode de configuration manuelle est utilisée. Vous pouvez ouvrir le tunnel existant que vous avez créé, puis modifier la méthode de configuration du tunnel pour utiliser la configuration rapide. Pour plus d'informations, veuillez consulter [Ouvrez un tunnel existant et utilisez le protocole SSH basé sur un navigateur \(p. 867\)](#).

Voici un exemple de la façon de de de de d'exécuter cette opération d'API. Facultativement, si vous souhaitez spécifier le nom de l'objet et le service de destination, utilisez le `DestinationConfig` paramètre. Pour obtenir un exemple qui montre comment d'utiliser ce paramètre, consultez [Ouvrez un nouveau tunnel pour l'appareil distant \(p. 865\)](#).

```
aws iotsecuretunneling open-tunnel
```

L'exécution de cette commande crée un nouveau tunnel et vous fournit les jetons d'accès source et de destination.

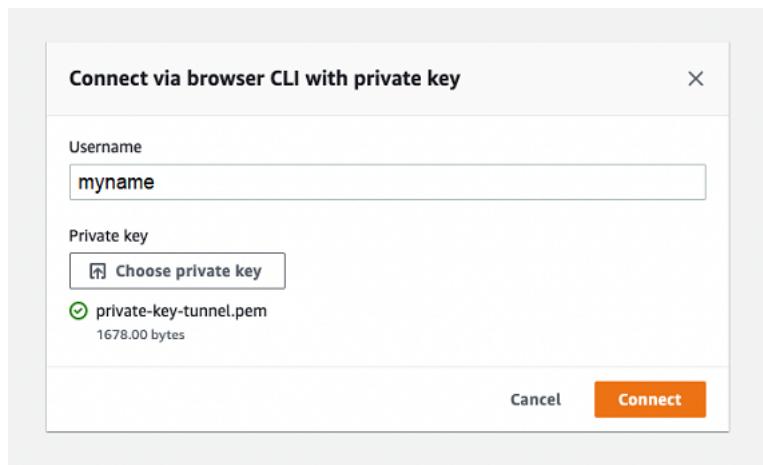
```
{  
    "tunnelId": "01234567-89ab-0123-4c56-789a01234bcd",  
    "tunnelArn": "arn:aws:iot:us-  
east-1:123456789012:tunnel/01234567-89ab-0123-4c56-789a01234bcd",  
    "sourceAccessToken": "<SOURCE_ACCESS_TOKEN>",  
    "destinationAccessToken": "<DESTINATION_ACCESS_TOKEN>"  
}
```

Utilisation de SSH basé sur un navigateur

Une fois que vous avez créé un tunnel à l'aide de la méthode de configuration rapide et que votre appareil de destination s'est connecté au tunnel, vous pouvez accéder au périphérique distant à l'aide d'un SSH basé sur un navigateur. À l'aide du SSH basé sur un navigateur, vous pouvez communiquer directement avec le périphérique distant en saisissant des commandes dans une interface de ligne de commande contextuelle de la console. Cette fonctionnalité vous permet d'interagir plus facilement avec le périphérique distant, car vous n'avez pas besoin d'ouvrir un terminal en dehors de la console ou de configurer le proxy local.

Pour utiliser le SSH basé sur un navigateur

1. Accédez au [hub Tunnels de la AWS IoT console](#) et choisissez le tunnel que vous avez créé pour afficher ses détails.
2. Développez la section Secure Shell (SSH), puis choisissez Connect.
3. Choisissez si vous souhaitez vous authentifier dans la connexion SSH en fournissant votre nom d'utilisateur et votre mot de passe ou, pour une authentification plus sécurisée, vous pouvez utiliser la clé privée de votre appareil. Si vous vous authentifiez à l'aide de la clé privée, vous pouvez utiliser les types de clés RSA, DSA, ECDSA (nistp-*) et ED25519, aux formats PEM (PKCS #1, PKCS #8) et OpenSSH.
 - Pour vous connecter à l'aide de votre nom d'utilisateur et de votre mot de passe, choisissez Utiliser le mot de passe. Vous pouvez ensuite saisir votre nom d'utilisateur et votre mot de passe et commencer à utiliser l'interface de ligne de commande intégrée au navigateur.
 - Pour vous connecter à l'aide de la clé privée de votre appareil de destination, choisissez Utiliser la clé privée. Spécifiez votre nom d'utilisateur et téléchargez le fichier de clé privée de l'appareil, puis choisissez Connect pour commencer à utiliser l'interface de ligne de commande intégrée au navigateur.



Après vous être authentifié dans la connexion SSH, vous pouvez rapidement commencer à saisir des commandes et à interagir avec l'appareil à l'aide de l'interface de ligne de commande du navigateur, car le proxy local a déjà été configuré pour vous.

▼ Command line interface [Info](#)

```
const [preferences, setPreferences] = React.useState(
  undefined
);
const [loading, setLoading] = React.useState(false);
return (
  <CodeEditor
    ace={ace}
    language="javascript"
    value="const pi = 3.14;"
```

Si l'interface de ligne de commande du navigateur reste ouverte après la durée du tunnel, elle risque d'expirer, entraînant la déconnexion de l'interface de ligne de commande. Vous pouvez dupliquer le tunnel et démarrer une autre session pour interagir avec le périphérique distant au sein de la console elle-même.

Dépannage de problèmes lors de l'utilisation de SSH basé sur un navigateur

Ce qui suit montre comment résoudre certains problèmes que vous pouvez rencontrer lors de l'utilisation du SSH basé sur un navigateur.

- Vous voyez une erreur à la place de l'interface de ligne de commande

Le message d'erreur s'affiche peut-être parce que votre appareil de destination a été déconnecté. Vous pouvez choisir Générer de nouveaux jetons d'accès pour générer de nouveaux jetons d'accès et les envoyer à votre appareil distant à l'aide de MQTT. Les nouveaux jetons peuvent être utilisés pour se reconnecter au tunnel. La reconnexion au tunnel efface l'historique et actualise la session de ligne de commande.

- Une erreur de déconnexion du tunnel s'affiche lors de l'authentification à l'aide d'une clé privée

Le message d'erreur s'affiche peut-être parce que votre clé privée n'a peut-être pas été acceptée par l'appareil de destination. Pour résoudre cette erreur, vérifiez le fichier de clé privée que vous avez chargé à des fins d'authentification. Si le message d'erreur persiste, consultez les journaux de votre appareil. Vous pouvez également essayer de vous reconnecter au tunnel en envoyant de nouveaux jetons d'accès à votre appareil distant.

- Votre tunnel était fermé lors de l'utilisation de la session

Si votre tunnel a été fermé parce qu'il est resté ouvert plus longtemps que la durée spécifiée, votre session de ligne de commande risque d'être déconnectée. Un tunnel ne peut pas être rouvert une fois fermé. Pour vous reconnecter, vous devez ouvrir un autre tunnel vers l'appareil.

Vous pouvez dupliquer un tunnel pour créer un nouveau tunnel avec les mêmes configurations que le tunnel fermé. Vous pouvez dupliquer un tunnel fermé depuis laAWS IoT console. Pour dupliquer le tunnel, choisissez le tunnel qui a été fermé pour afficher ses détails, puis choisissez Dupliquer le tunnel. Spécifiez la durée du tunnel que vous souhaitez utiliser, puis créez le nouveau tunnel.

Nettoyage

- Fermer le tunnel

Nous vous recommandons de fermer le tunnel une fois que vous avez fini de l'utiliser. Un tunnel peut également être fermé s'il est resté ouvert plus longtemps que la durée spécifiée. Un tunnel ne peut pas être rouvert une fois fermé. Vous pouvez toujours dupliquer un tunnel en choisissant le tunnel fermé, puis en choisissant Dupliquer le tunnel. Spécifiez la durée du tunnel que vous souhaitez utiliser, puis créez le nouveau tunnel.

- Pour fermer un tunnel individuel ou plusieurs tunnels depuis laAWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez fermer, puis choisissez Fermer le tunnel.
- Pour fermer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'[CloseTunnelAPI](#).

```
aws iotsecuretunneling close-tunnel \
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"
```

- Supprimer le tunnel

Vous pouvez supprimer définitivement un tunnel de votreCompte AWS.

Warning

Les actions de suppression sont permanentes et ne peuvent pas être annulées.

- Pour supprimer un tunnel individuel ou plusieurs tunnels depuis laAWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez supprimer, puis choisissez Supprimer le tunnel.
- Pour supprimer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'[CloseTunnelAPI](#). Lorsque vous utilisez l'API, définissez l'indicateur `--delete true`.

```
aws iotsecuretunneling close-tunnel \
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"
--delete true
```

Ouvrez un tunnel à l'aide de la configuration manuelle et connectez-vous à un appareil distant

Lorsque vous ouvrez un tunnel, vous pouvez choisir la méthode de configuration rapide ou la méthode de configuration manuelle pour ouvrir un tunnel dans le périphérique distant. Ce didacticiel explique comment ouvrir un tunnel à l'aide de la méthode de configuration manuelle et comment configurer et démarrer le proxy local pour se connecter au périphérique distant.

Lorsque vous utilisez la méthode de configuration manuelle, vous devez spécifier manuellement les configurations du tunnel lors de la création du tunnel. Après avoir créé le tunnel, vous pouvez utiliser SSH dans le navigateur ou ouvrir un terminal en dehors de la AWS IoT console. Ce didacticiel explique comment utiliser le terminal en dehors de la console pour accéder au périphérique distant. Vous apprendrez également à configurer le proxy local, puis à vous connecter au proxy local pour interagir avec le périphérique distant. Pour vous connecter au proxy local, vous devez télécharger le jeton d'accès à la source lors de la création du tunnel.

Avec cette méthode de configuration, vous pouvez utiliser des services autres que SSH, tels que le FTP pour vous connecter au périphérique distant. Pour de plus amples informations sur les différentes méthodes de configuration, consultez [Méthodes de configuration de tunnel \(p. 852\)](#).

Prérequis pour la méthode de configuration manuelle

- Les pare-feux derrière lesquels se trouve le périphérique distant doivent autoriser le trafic sortant sur le port 443. Le tunnel que vous créez utilisera ce port pour se connecter au périphérique distant.
- Vous disposez d'un agent d'appareil IoT (voir [Extrait de l'agent IoT \(p. 883\)](#)) exécuté sur l'appareil distant qui se connecte à la passerelle de l'AWS IoTAppareil et qui est configuré avec un abonnement à une rubrique MQTT. Pour plus d'informations, consultez la section [Connecter un appareil à la passerelle de l'AWS IoTAppareil](#).
- Vous devez disposer d'un démon SSH s'exécutant sur l'appareil distant.
- Vous avez téléchargé le code source du proxy local [GitHub](#)et l'avez créé pour la plateforme de votre choix. Dans ce didacticiel, nous utilisons `localproxy` pour nous référer au fichier exécutable du proxy local.

Ouvrir un tunnel

Vous pouvez ouvrir un tunnel sécurisé à l'aide duAWS Management Console, de l'AWS IoTAPI Reference ou duAWS CLI. Vous pouvez éventuellement configurer un nom de destination, mais ce n'est pas obligatoire pour ce didacticiel. Si vous configurez la destination, le tunneling sécurisé fournira automatiquement le jeton d'accès au périphérique distant à l'aide de MQTT. Pour plus d'informations, veuillez consulter [Méthodes de création de tunnels dansAWS IoT la console \(p. 852\)](#).

Pour ouvrir un tunnel dans la console

1. Accédez au [hub Tunnels de laAWS IoT console](#) et choisissez Créez un tunnel.

The screenshot shows the AWS IoT console interface under the 'Manage' section, specifically the 'Tunnels' page. At the top, there are buttons for 'Close tunnel' and 'Delete tunnel', and a prominent orange 'Create tunnel' button. Below these are search and filter options. A table header includes columns for 'Tunnel ID', 'Tunnel status', and 'Date created'. The main content area displays a message: 'No tunnels' and 'You don't have any tunnels.' with a 'Create tunnel' button below it.

- Pour ce didacticiel, choisissez Configuration manuelle comme méthode de création du tunnel, puis choisissez Suivant. Pour plus d'informations sur l'utilisation de la méthode de configuration rapide pour créer un tunnel, consultez [Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant \(p. 853\)](#).

Note

Si vous créez un tunnel sécurisé à partir de la page de détails d'un objet, vous pouvez choisir de créer un nouveau tunnel ou d'utiliser un tunnel existant. Pour plus d'informations, veuillez consulter [Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur \(p. 865\)](#).

Setup method

- Quick setup (SSH)
 Manual setup

Manual setup

When creating a tunnel using manual setup, you must manually specify the tunnel configurations. You must manually:

- Configure and launch the local proxy. Learn more about setting up your local proxy [here](#).
- Download, enter, and manage the access tokens for connecting to the remote device.

- (Facultatif) Entrez les paramètres de configuration de votre tunnel. Vous pouvez également ignorer cette étape et passer à l'étape suivante pour créer un tunnel.

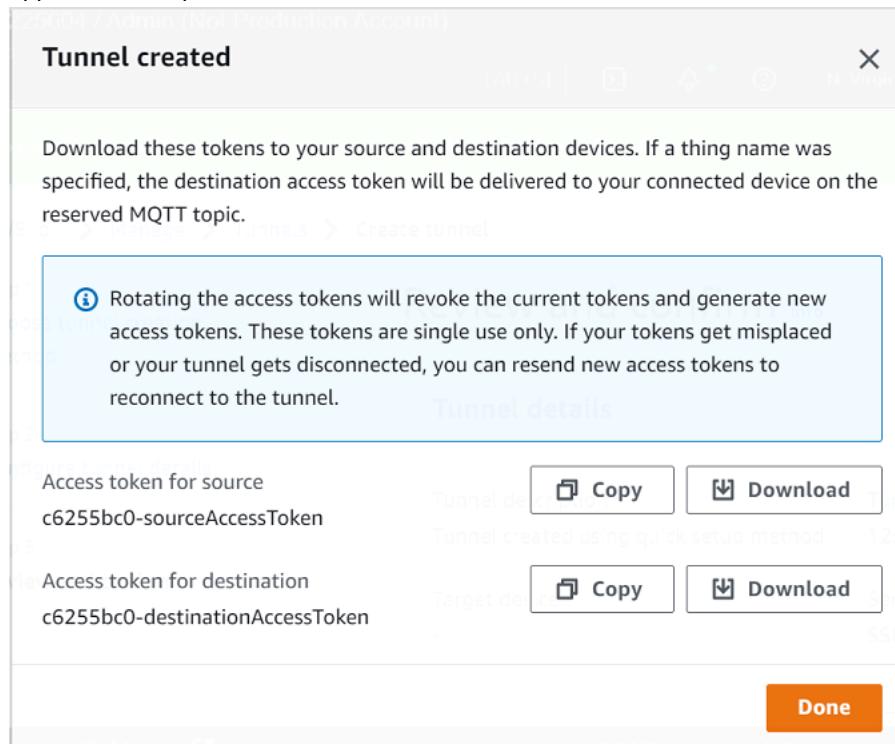
Entrez une description du tunnel, une durée d'expiration du tunnel et des balises de ressource sous forme de paires clé-valeur pour vous aider à identifier votre ressource. Pour ce didacticiel, vous pouvez ignorer la configuration de la destination.

Note

Vous ne serez pas facturé en fonction de la durée pendant laquelle vous maintenez un tunnel ouvert. Des frais ne vous sont facturés que lors de la création d'un nouveau tunnel. Pour plus d'informations sur les prix, consultez la section [Secure Tunneling](#) dans la section [AWS IoT Device Management Tarification](#).

- Téléchargez les jetons d'accès client, puis choisissez OK. Les jetons ne pourront pas être téléchargés une fois que vous aurez sélectionné Terminé.

Ces jetons ne peuvent être utilisés qu'une seule fois pour se connecter au tunnel. Si vous égarez les jetons ou si le tunnel est déconnecté, vous pouvez générer et envoyer de nouveaux jetons à votre appareil distant pour vous reconnecter au tunnel.



Pour ouvrir un tunnel à l'aide de l'API

Pour ouvrir un nouveau tunnel, vous pouvez utiliser l'opération [OpenTunnel](#) API. Vous pouvez également spécifier des configurations supplémentaires à l'aide de l'API, telles que la durée du tunnel et la configuration de destination.

```
aws iotsecuretunneling open-tunnel \
--region us-east-1 \
--endpoint https://api.us-east-1.tunneling.iot.amazonaws.com
```

L'exécution de cette commande crée un nouveau tunnel et vous fournit les jetons d'accès source et de destination.

```
{
  "tunnelId": "01234567-89ab-0123-4c56-789a01234bcd",
  "tunnelArn": "arn:aws:iot:us-
east-1:123456789012:tunnel/01234567-89ab-0123-4c56-789a01234bcd",
  "sourceAccessToken": "<SOURCE_ACCESS_TOKEN>",
  "destinationAccessToken": "<DESTINATION_ACCESS_TOKEN>"
}
```

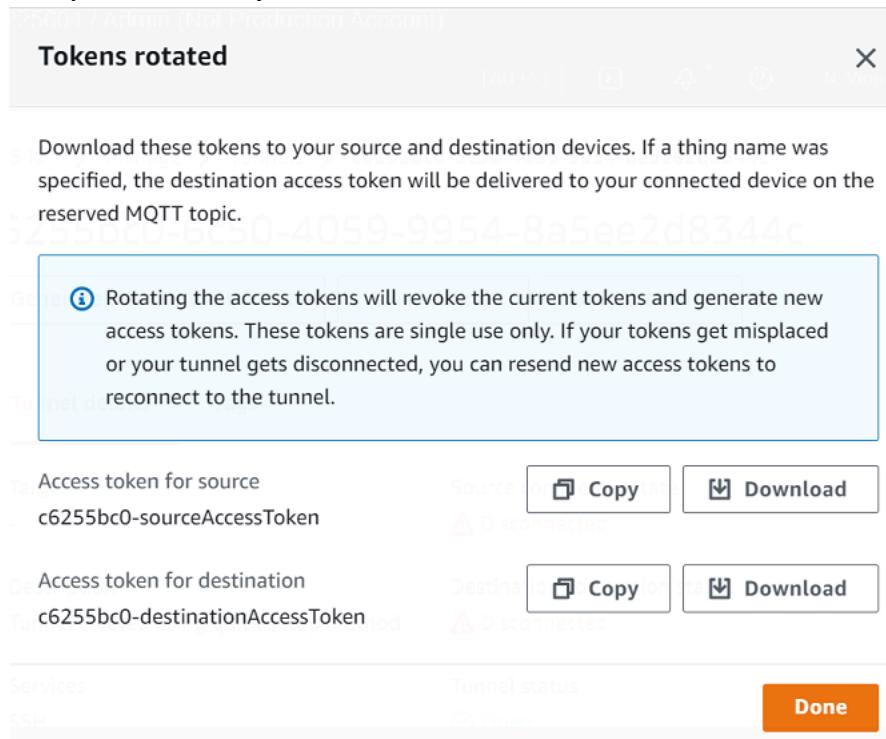
Renvoyer les jetons d'accès à un tunnel

Les jetons que vous avez obtenus lors de la création d'un tunnel ne peuvent être utilisés qu'une seule fois pour vous connecter au tunnel. Si vous égarez le jeton d'accès ou si le tunnel est déconnecté, vous pouvez renvoyer de nouveaux jetons d'accès au périphérique distant à l'aide de MQTT sans frais supplémentaires.

AWS IoT tunneling sécurisé révoquera les jetons actuels et renverra de nouveaux jetons d'accès pour la reconnexion au tunnel.

Pour faire pivoter les jetons depuis la console

1. Accédez au [hub Tunnels de la AWS IoT console](#) et choisissez le tunnel que vous avez créé.
2. Sur la page des détails du tunnel, choisissez Générer de nouveaux jetons d'accès, puis cliquez sur Suivant.
3. Téléchargez les nouveaux jetons d'accès pour votre tunnel et choisissez OK. Ces jetons ne peuvent être utilisés qu'une seule fois. Si vous égarez ces jetons ou si le tunnel est déconnecté, vous pouvez renvoyer de nouveaux jetons d'accès.



Pour alterner les jetons d'accès à l'aide de l'API

Pour faire pivoter les jetons d'accès au tunnel, vous pouvez utiliser l'opération [d'RotateTunnelAccessToken](#) API pour révoquer les jetons actuels et renvoyer de nouveaux jetons d'accès pour vous reconnecter au tunnel. Par exemple, la commande suivante fait pivoter les jetons d'accès pour le périphérique de destination, *RemoteThing1*.

```
aws iotsecuretunneling rotate-tunnel-access-token \
--tunnel-id <tunnel-id> \
--client-mode DESTINATION \
--destination-config thingName=<RemoteThing1>,services=SSH \
--region <region>
```

L'exécution de cette commande génère le nouveau jeton d'accès, comme indiqué dans l'exemple suivant. Le jeton est ensuite envoyé à l'appareil à l'aide de MQTT pour se connecter au tunnel, si l'agent du périphérique est correctement configuré.

```
{
```

```
        "destinationAccessToken": "destination-access-token",  
        "tunnelArn": "arn:aws:iot:region:account-id:tunnel/tunnel-id"  
    }
```

Pour des exemples montrant comment et quand faire pivoter les jetons d'accès, consultez [Résolution des problèmes de connectivité par tunneling AWS IoT sécurisé en alternant les jetons d'accès client \(p. 890\)](#).

Configuration et démarrage du proxy local

Pour vous connecter au périphérique distant, ouvrez un terminal sur votre ordinateur portable, configurez et démarrez le proxy local. Le proxy local transmet les données envoyées par l'application exécutée sur le périphérique source en utilisant un tunnel sécurisé via une connexion WebSocket sécurisée. Vous pouvez télécharger la source proxy locale à partir de [GitHub](#).

Après avoir configuré le proxy local, copiez le jeton d'accès au client source et utilisez-le pour démarrer le proxy local en mode source. Voici un exemple de commande pour démarrer le proxy local. Dans la commande suivante, le proxy local est configuré pour écouter les nouvelles connexions sur le port 5555. Dans cette commande :

- -r spécifie la Région AWS, qui doit être la même région que celle dans laquelle votre tunnel a été créé.
- -s spécifie le port auquel le proxy doit se connecter.
- -t spécifie le texte du jeton client.

```
./localproxy -r us-east-1 -s 5555 -t source-client-access-token
```

L'exécution de cette commande démarre le proxy local en mode source. Si vous recevez le message d'erreur suivant après avoir exécuté la commande, configurez le chemin de l'autorité de l'autorité de certification. Pour plus d'informations, consultez la section [Proxy local de tunneling sécurisé activé GitHub](#).

```
Could not perform SSL handshake with proxy server: certificate verify failed
```

Ce qui suit montre un exemple de sortie de l'exécution du proxy local en source mode.

```
...  
...  
Starting proxy in source mode  
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-east-1.amazonaws.com:443  
Resolved proxy server IP: 10.10.0.11  
Connected successfully with proxy server  
Performing SSL handshake with proxy server  
Successfully completed SSL handshake with proxy server  
HTTP/1.1 101 Switching Protocols  
...  
  
Connection: upgrade  
channel-id: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456  
upgrade: websocket  
...  
  
Web socket session ID: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456  
Web socket subprotocol selected: aws.iot.securetunneling-2.0
```

```
Successfully established websocket connection with proxy server: wss://  
data.tunneling.iot.us-east-1.amazonaws.com:443  
Setting up web socket pings for every 5000 milliseconds  
Scheduled next read:  
  
...  
Starting web socket read loop continue reading...  
Resolved bind IP: 127.0.0.1  
Listening for new connection on port 5555
```

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.

Nettoyage

- Fermer le tunnel

Nous vous recommandons de fermer le tunnel une fois que vous avez fini de l'utiliser. Un tunnel peut également être fermé s'il est resté ouvert plus longtemps que la durée spécifiée. Un tunnel ne peut pas être rouvert une fois fermé. Vous pouvez toujours dupliquer un tunnel en ouvrant le tunnel fermé, puis en choisissant Dupliquer le tunnel. Spécifiez la durée du tunnel que vous souhaitez utiliser, puis créez le nouveau tunnel.

- Pour fermer un tunnel individuel ou plusieurs tunnels depuis la AWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez fermer, puis choisissez Fermer le tunnel.
- Pour fermer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'opération [CloseTunnel](#)API.

```
aws iotsecuretunneling close-tunnel \  
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"
```

- Supprimer le tunnel

Vous pouvez supprimer définitivement un tunnel de votre Compte AWS.

Warning

Les actions de suppression sont permanentes et ne peuvent pas être annulées.

- Pour supprimer un tunnel individuel ou plusieurs tunnels depuis la AWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez supprimer, puis choisissez Supprimer le tunnel.
- Pour supprimer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'opération [CloseTunnel](#)API. Lorsque vous utilisez l'API, définissez l'indicateur `delete` sur `true`.

```
aws iotsecuretunneling close-tunnel \  
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"  
--delete true
```

Ouvrez un tunnel pour un appareil distant et utilisez le SSH basé sur un navigateur

Depuis laAWS IoT console, vous pouvez créer un tunnel depuis le hub Tunnels ou depuis la page de détails d'un objet IoT que vous avez créé. Lorsque vous créez un tunnel à partir du hub Tunnels, vous pouvez spécifier si vous souhaitez créer un tunnel à l'aide de la configuration rapide ou de la configuration manuelle. Pour voir un exemple de didacticiel, consultez la section [Ouvrez un tunnel et démarrez une session SSH vers un appareil distant \(p. 851\)](#).

Lorsque vous créez un tunnel à partir de la page de détails de l'objet de laAWS IoT console, vous pouvez également spécifier si vous souhaitez créer un nouveau tunnel ou ouvrir un tunnel existant pour cet objet, comme illustré dans ce didacticiel. Si vous choisissez un tunnel existant, vous pouvez accéder au tunnel ouvert le plus récent que vous avez créé pour cet appareil. Vous pouvez ensuite utiliser l'interface de ligne de commande du terminal pour accéder au périphérique via SSH.

Prérequis

- Les pare-feux derrière lesquels se trouve le périphérique distant doivent autoriser le trafic sortant sur le port 443. Le tunnel que vous créez utilisera ce port pour se connecter au périphérique distant.
- Vous avez créé un objet IoT (par exemple `RemoteDevice1`) dans leAWS IoT registre. Cela correspond à la représentation de votre appareil distant dans le cloud. Pour de plus amples informations, consultez [Enregistrement d'un appareil dans leAWS IoT registre](#).
- Vous disposez d'un agent d'appareil IoT (voir [Extrait de l'agent IoT \(p. 883\)](#)) exécuté sur l'appareil distant qui se connecte à la passerelle de l'AWS IoT appareil et qui est configuré avec un abonnement à une rubrique MQTT. Pour plus d'informations, consultez la section [Connecter un appareil à la passerelle de l'AWS IoT appareil](#).
- Vous devez disposer d'un démon SSH s'exécutant sur l'appareil distant.

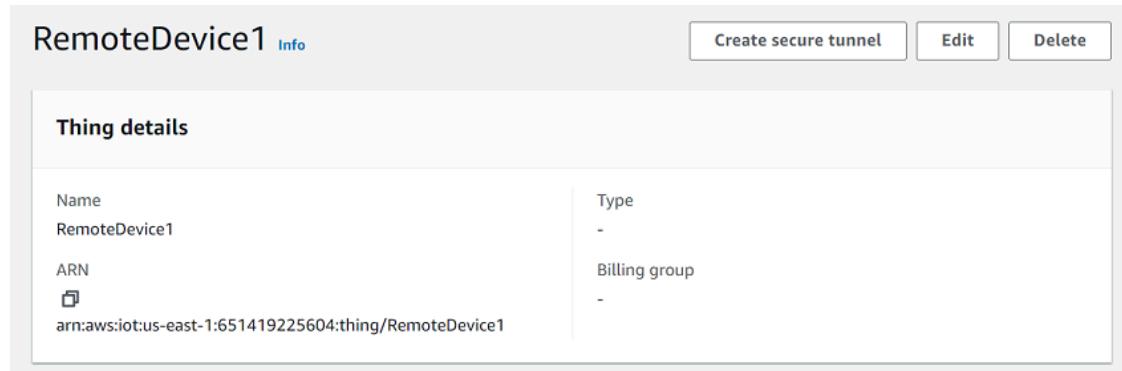
Ouvrez un nouveau tunnel pour l'appareil distant

Supposons que vous souhaitez ouvrir un tunnel vers votre appareil distant, `RemoteDevice1`. Tout d'abord, créez un objet IoT dont le nom `RemoteDevice1` figure dans leAWS IoT registre. Vous pouvez ensuite créer un tunnel à l'AWS Management Console à l'aide de l'AWS IoT API de référence ou duAWS CLI.

En configurant une destination lors de la création d'un tunnel, le service de tunneling sécurisé fournit le jeton d'accès client de destination au périphérique distant via MQTT et la rubrique MQTT réservée (`$aws/things/RemoteDeviceA/tunnels/notify`). Pour plus d'informations, veuillez consulter [Méthodes de création de tunnels dans AWS IoT la console \(p. 852\)](#).

Pour créer un tunnel pour un appareil distant à partir de la console

1. Choisissez l'objet `RemoteDevice1`, pour afficher ses détails, puis choisissez Créer un tunnel sécurisé.



- Choisissez de créer un nouveau tunnel ou d'ouvrir un tunnel existant. Pour créer un nouveau tunnel, choisissez Créer un nouveau tunnel. Vous pouvez ensuite choisir d'utiliser la configuration manuelle ou la méthode de configuration rapide pour créer le tunnel. Pour plus d'informations, consultez [Ouvrez un tunnel à l'aide de la configuration manuelle et connectez-vous à un appareil distant \(p. 859\)](#) et [Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant \(p. 853\)](#).

Pour créer un tunnel pour un appareil distant à l'aide de l'API

Pour ouvrir un nouveau tunnel, vous pouvez utiliser l'opération [OpenTunnel](#) API. Le code suivant montre un exemple de l'exécution de cette commande.

```
aws iotsecuretunneling open-tunnel \
--region us-east-1 \
--endpoint https://api.us-east-1.tunneling.iot.amazonaws.com
--cli-input-json file://input.json
```

Ce qui suit montre le contenu du input.json fichier. Vous pouvez utiliser le destinationConfig paramètre pour spécifier le nom du périphérique de destination (par exemple, `RemoteDevice1`) et le service que vous souhaitez utiliser pour accéder au périphérique de destination, tel que `SSH`. Le cas échéant, vous pouvez également spécifier des paramètres supplémentaires tels que la description de tunnel et les balises.

Contenu du fichier input.json

```
{
  "description": "Tunnel to remote device1",
  "destinationConfig": {
    "services": [ "SSH" ],
    "thingName": "RemoteDevice1"
  }
}
```

L'exécution de cette commande crée un nouveau tunnel et vous fournit les jetons d'accès source et de destination.

```
{
  "tunnelId": "01234567-89ab-0123-4c56-789a01234bcd",
  "tunnelArn": "arn:aws:iot:us-
east-1:123456789012:tunnel/01234567-89ab-0123-4c56-789a01234bcd",
  "sourceAccessToken": "<SOURCE_ACCESS_TOKEN>",
  "destinationAccessToken": "<DESTINATION_ACCESS_TOKEN>"
}
```

Ouvrez un tunnel existant et utilisez le protocole SSH basé sur un navigateur

Supposons que vous ayez créé le tunnel pour votre appareil distant à l'aide de la méthode de configuration manuelle ou de l'AWS IoT API de référence.`RemoteDevice1` Vous pouvez ensuite ouvrir le tunnel existant pour l'appareil et choisir Configuration rapide pour utiliser la fonctionnalité SSH basée sur un navigateur. Les configurations d'un tunnel existant ne peuvent pas être modifiées. Vous ne pouvez donc pas utiliser la méthode de configuration manuelle.

Pour utiliser la fonctionnalité SSH basée sur un navigateur, vous n'avez pas à télécharger le jeton d'accès à la source ni à configurer le proxy local. Un proxy local basé sur le Web sera automatiquement configuré pour vous afin que vous puissiez commencer à interagir avec votre appareil distant.

Pour utiliser la méthode de configuration rapide et le SSH basé sur un navigateur

1. Accédez à la page de détails de l'objet que vous avez créé`RemoteDevice1`, puis créez un tunnel sécurisé.
2. Choisissez Utiliser le tunnel existant pour ouvrir le tunnel ouvert le plus récent que vous avez créé pour le périphérique distant. Les configurations du tunnel ne peuvent pas être modifiées. Vous ne pouvez donc pas utiliser la méthode de configuration manuelle du tunnel. Pour utiliser la méthode de configuration rapide, choisissez Configuration rapide.
3. Passez en revue et confirmez les détails de configuration du tunnel et créez le tunnel. Les configurations de tunnel ne peuvent pas être modifiées.

Lorsque vous créez le tunnel, le tunneling sécurisé utilise l'opération d'[RotateTunnelAccessToken](#) API pour révoquer les jetons d'accès d'origine et générer de nouveaux jetons d'accès. Si votre appareil distant utilise MQTT, ces jetons seront automatiquement envoyés à l'appareil distant sur la rubrique MQTT à laquelle il est abonné. Vous pouvez également choisir de télécharger ces jetons manuellement sur votre appareil source.

Après avoir créé le tunnel, vous pouvez utiliser le SSH basé sur un navigateur pour interagir avec le périphérique distant directement depuis la console à l'aide de l'interface de ligne de commande contextuelle. Pour utiliser cette interface de ligne de commande, choisissez le tunnel correspondant à l'objet que vous avez créé et, sur la page de détails, développez la section Interface de ligne de commande. Comme le proxy local a déjà été configuré pour vous, vous pouvez commencer à saisir des commandes pour commencer rapidement à accéder à votre appareil distant et à interagir avec celui-ci`RemoteDevice1`.

Pour plus d'informations sur la méthode de configuration rapide et l'utilisation du protocole SSH basé sur un navigateur, consultez[Ouvrez un tunnel et utilisez le protocole SSH basé sur un navigateur pour accéder à un appareil distant \(p. 853\)](#).

Nettoyage

- Fermer le tunnel

Nous vous recommandons de fermer le tunnel une fois que vous avez fini de l'utiliser. Un tunnel peut également être fermé s'il est resté ouvert plus longtemps que la durée spécifiée. Un tunnel ne peut pas être rouvert une fois fermé. Vous pouvez toujours dupliquer un tunnel en ouvrant le tunnel fermé, puis en choisissant Dupliquer le tunnel. Spécifiez la durée du tunnel que vous souhaitez utiliser, puis créez le nouveau tunnel.

- Pour fermer un tunnel individuel ou plusieurs tunnels depuis la AWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez fermer, puis choisissez Fermer le tunnel.
- Pour fermer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'opération [CloseTunnel](#) API.

```
aws iotsecuretunneling close-tunnel \
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"
```

- Supprimer le tunnel

Vous pouvez supprimer définitivement un tunnel de votreCompte AWS.

Warning

Les actions de suppression sont permanentes et ne peuvent pas être annulées.

- Pour supprimer un tunnel individuel ou plusieurs tunnels depuis laAWS IoT console, accédez au [hub Tunnels](#), choisissez les tunnels que vous souhaitez supprimer, puis choisissez Supprimer le tunnel.
- Pour supprimer un tunnel individuel ou plusieurs tunnels à l'aide de l'AWS IoT API de référence, utilisez l'opération [CloseTunnel](#)API. Lorsque vous utilisez l'API, définissez l'`delete`indicateur surtrue.

```
aws iotsecuretunneling close-tunnel \
--tunnel-id "01234567-89ab-0123-4c56-789a01234bcd"
--delete true
```

Proxy local

Le proxy local transmet les données envoyées par l'application exécutée sur le périphérique source en utilisant un tunnel sécurisé via une connexion WebSocket sécurisée. Vous pouvez télécharger la source proxy locale 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. Un seul tunnel peut prendre en charge jusqu'à trois flux de données à la fois en utilisant le multiplexage par tunnel. Pour chaque flux de données, le tunneling sécurisé utilise plusieurs connexions TCP, ce qui réduit le risque de délai d'attente. Pour plus d'informations, veuillez consulter [Multiplex des flux de données et utilisation de connexions TCP simultanées dans un tunnel sécurisé \(p. 878\)](#).

Comment utiliser le proxy local

Vous pouvez exécuter le proxy local sur les appareils source et de destination pour transmettre des données aux points de terminaison sécurisés du tunneling. Si vos appareils se trouvent sur un réseau qui utilise un proxy Web, celui-ci peut intercepter les connexions avant de les transférer vers Internet. Dans ce cas, vous devez configurer votre proxy local pour utiliser le proxy Web. Pour plus d'informations, veuillez consulter [Configuration du proxy local pour les appareils utilisant un proxy Web \(p. 872\)](#).

Flux de travail de proxy local

Les étapes suivantes montrent comment le proxy local est exécuté sur les appareils source et de destination.

1. Connect un proxy local à un tunneling sécurisé

Tout d'abord, le proxy local doit établir une connexion pour sécuriser le tunneling. Lorsque vous démarrez le proxy local, utilisez les arguments suivants :

- `-r`Argument permettant de spécifier l'Région AWSendroit dans lequel le tunnel est ouvert.
- L'`-t`argument pour transmettre le jeton d'accès au client source ou de destination renvoyé par`openTunnel`.

Note

Deux proxy locaux utilisant la même valeur de jeton d'accès client ne peuvent pas être connectés en même temps.

2. Exécuter des actions de source ou de destination

Une fois la WebSocket connexion établie, le proxy local exécute des actions en mode source ou en mode destination, selon sa configuration.

Par défaut, le proxy local tente de se reconnecter au tunneling sécurisé en cas d'erreur d'entrée/sortie (E/S) ou si la WebSocket connexion est fermée de manière 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.

3. Arrêter le proxy local

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`.

Pour plus d'informations sur la façon d'utiliser le AWS Management Console pour ouvrir un tunnel et démarrer une session SSH, consultez[Ouvrez un tunnel et démarrez une session SSH vers un appareil distant \(p. 851\)](#).

Bonnes pratiques du proxy local

Lorsque vous exécutez le proxy local, suivez ces bonnes pratiques :

- 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, des conteneurs, des sandbox, une archive chroot 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.

Exemple de commande et de sortie

Vous pouvez voir ci-dessous un exemple de commande que vous exécutez et la sortie correspondante. L'exemple montre comment le proxy local peut être configuré dans `source` les deux `destination` modes. Le proxy local met à niveau le protocole HTTPS WebSockets pour établir une connexion de longue durée, puis commence à transmettre des données via la connexion aux points de terminaison sécurisés du dispositif de tunneling.

Avant d'exécuter ces commandes :

Vous devez avoir ouvert un tunnel et obtenu les jetons d'accès client pour la source et la destination. Vous devez également avoir créé le proxy local comme décrit précédemment. Pour créer le proxy local, ouvrez le

[code source du proxy local](#) dans le GitHub référentiel et suivez les instructions de création et d'installation du proxy local.

Note

Les commandes suivantes utilisées dans les exemples utilisent l'`verbosity` indicateur pour illustrer une vue d'ensemble des différentes étapes décrites précédemment après l'exécution du proxy local. Nous vous recommandons de n'utiliser cet indicateur qu'à des fins de test.

Exécution d'un proxy local en mode source

Les commandes suivantes montrent comment exécuter le proxy local en mode source.

Linux/macOS

Sous Linux ou macOS, exécutez les commandes suivantes dans le terminal pour configurer et démarrer le proxy local sur votre source.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
./localproxy -s 5555 -v 5 -r us-west-2
```

Où :

- `-s` est le port d'écoute source, qui démarre le proxy local en mode source.
- `-v` est la verbosité de la sortie, qui peut être une valeur comprise entre zéro et six.
- `-r` est la région terminale où le tunnel est ouvert.

Pour plus d'informations sur les paramètres, consultez la section [Options définies à l'aide d'arguments de ligne de commande](#).

Windows

Sous Windows, vous configurez le proxy local de la même manière que pour Linux ou macOS, mais la façon dont vous définissez les variables d'environnement est différente de celle des autres plateformes. Exécutez les commandes suivantes dans la cmd fenêtre pour configurer et démarrer le proxy local sur votre source.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
.\\localproxy -s 5555 -v 5 -r us-west-2
```

Où :

- `-s` est le port d'écoute source, qui démarre le proxy local en mode source.
- `-v` est la verbosité de la sortie, qui peut être une valeur comprise entre zéro et six.
- `-r` est la région terminale où le tunnel est ouvert.

Pour plus d'informations sur les paramètres, consultez la section [Options définies à l'aide d'arguments de ligne de commande](#).

Ce qui suit montre un exemple de sortie de l'exécution du proxy local en source mode.

```
...  
Starting proxy in source mode  
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-west-2.amazonaws.com:443
```

```
Resolved proxy server IP: 10.10.0.11
Connected successfully with proxy server
Performing SSL handshake with proxy server
Successfully completed SSL handshake with proxy server
HTTP/1.1 101 Switching Protocols

...
Connection: upgrade
channel-id: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
upgrade: websocket

...
Web socket session ID: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Setting up web socket pings for every 5000 milliseconds
Scheduled next read:

...
Starting web socket read loop continue reading...
Resolved bind IP: 127.0.0.1
Listening for new connection on port 5555
```

Exécution d'un proxy local en mode destination

Les commandes suivantes montrent comment exécuter le proxy local en mode destination.

Linux/macOS

Sous Linux ou macOS, exécutez les commandes suivantes dans le terminal pour configurer et démarrer le proxy local sur votre destination.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
./localproxy -d 22 -v 5 -r us-west-2
```

Où :

- **-d**est l'application de destination qui démarre le proxy local en mode destination.
- **-v**est la verbosité de la sortie, qui peut être une valeur comprise entre zéro et six.
- **-r**est la région terminale où le tunnel est ouvert.

Pour plus d'informations sur les paramètres, consultez la section [Options définies à l'aide d'arguments de ligne de commande](#).

Windows

Sous Windows, vous configurez le proxy local de la même manière que pour Linux ou macOS, mais la façon dont vous définissez les variables d'environnement est différente de celle des autres plateformes. Exécutez les commandes suivantes dans la cmd fenêtre pour configurer et démarrer le proxy local sur votre destination.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
.\localproxy -d 22 -v 5 -r us-west-2
```

Où :

- -dest l'application de destination qui démarre le proxy local en mode destination.
- -vest la verbosité de la sortie, qui peut être une valeur comprise entre zéro et six.
- -rest la région terminale où le tunnel est ouvert.

Pour plus d'informations sur les paramètres, consultez la section [Options définies à l'aide d'arguments de ligne de commande](#).

Ce qui suit montre un exemple de sortie de l'exécution du proxy local en destination mode.

```
...
Starting proxy in destination mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-west-2.amazonaws.com:443
Resolved proxy server IP: 10.10.0.11
Connected successfully with proxy server
Performing SSL handshake with proxy server
Successfully completed SSL handshake with proxy server
HTTP/1.1 101 Switching Protocols

...
Connection: upgrade
channel-id: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
upgrade: websocket

...
Web socket session ID: 01234567890abc23-00001234-0005678a-b1234c5de677a001-2bc3d456
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Setting up web socket pings for every 5000 milliseconds
Scheduled next read:

...
Starting web socket read loop continue reading...
```

Configuration du proxy local pour les appareils utilisant un proxy Web

Vous pouvez utiliser un proxy local sur AWS IoT pour communiquer avec des API de tunneling AWS IoT sécurisées. Le proxy local transmet les données envoyées par l'application de l'appareil à l'aide d'un tunnel sécurisé via une connexion WebSocket sécurisée. Le proxy local peut fonctionner en destination mode ou en source mode. En destination mode, il s'exécute sur le même appareil ou réseau qui initie la connexion TCP. En destination mode, le proxy local s'exécute sur le périphérique distant, en même temps que l'application de destination. Pour plus d'informations, veuillez consulter [Proxy local \(p. 868\)](#).

Le proxy local doit se connecter directement à Internet pour utiliser le tunneling AWS IoT sécurisé. Pour une connexion TCP de longue durée avec tunneling sécurisé, le proxy local met à niveau la requête HTTPS pour établir une WebSockets connexion à l'un des [points de connexion du périphérique de tunneling sécurisé](#).

Si vos appareils se trouvent sur un réseau qui utilise un proxy Web, celui-ci peut intercepter les connexions avant de les transférer vers Internet. Pour établir une connexion de longue durée avec les points de

connexion du périphérique de tunneling sécurisé, configurez votre proxy local pour utiliser le proxy Web comme décrit dans la [spécification WebSocket](#).

Note

[AWS IoT Appareil client \(p. 1497\)](#) ne prend pas en charge les appareils utilisant un proxy Web. Pour utiliser le proxy Web, vous devez utiliser un proxy local et le configurer pour qu'il fonctionne avec un proxy Web, comme décrit ci-dessous.

Les étapes suivantes montrent comment le proxy local fonctionne avec un proxy Web.

1. Le proxy local envoie une CONNECT requête HTTP au proxy Web qui contient l'adresse distante du service de tunneling sécurisé, ainsi que les informations d'authentification du proxy Web.
2. Le proxy Web créera ensuite une connexion de longue durée avec les points de terminaison du tunneling sécurisé à distance.
3. La connexion TCP est établie et le proxy local fonctionne désormais à la fois en mode source et en mode destination pour la transmission de données.

Pour terminer cette procédure, effectuez les étapes suivantes.

- [Créez le proxy local \(p. 873\)](#)
- [Configurez votre proxy Web \(p. 873\)](#)
- [Configuration et démarrage du proxy local \(p. 874\)](#)

Créez le proxy local

Ouvrez le [code source du proxy local](#) dans le GitHub référentiel et suivez les instructions pour créer et installer le proxy local.

Configurez votre proxy Web

Le proxy local repose sur le mécanisme de tunneling HTTP décrit par la [spécification HTTP/1.1](#). Pour respecter les spécifications, votre proxy Web doit autoriser les appareils à utiliser CONNECT cette méthode.

La façon dont vous configurez votre proxy Web dépend du proxy Web que vous utilisez et de la version du proxy Web. Pour vous assurer de configurer correctement le proxy Web, consultez la documentation de votre proxy Web.

Pour configurer votre proxy Web, identifiez d'abord l'URL de votre proxy Web et vérifiez si votre proxy Web prend en charge le tunneling HTTP. L'URL du proxy Web sera utilisée ultérieurement lors de la configuration et du démarrage du proxy local.

1. Identifiez l'URL de votre proxy Web

Le format de l'URL de votre proxy web sera au format suivant.

`protocol://web_proxy_host_domain:web_proxy_port`

AWS IoT tunneling sécurisé ne prend en charge que l'authentification de base pour le proxy Web. Pour utiliser l'authentification de base, vous devez spécifier le **username** et dans le **password** cadre de l'URL du proxy Web. L'URL du proxy web sera au format suivant.

`protocol://username:password@web_proxy_host_domain:web_proxy_port`

- *le protocole* peut être http ou https. Nous vous recommandons d'utiliser https.

- **web_proxy_host_domain** est l'adresse IP de votre proxy Web ou un nom DNS qui correspond à l'adresse IP de votre proxy Web.
- **web_proxy_port** est le port sur lequel le proxy Web écoute.
- Le proxy Web utilise **username et password** pour authentifier la demande.

2. Testez l'URL de votre proxy web

Pour vérifier si votre proxy Web prend en charge le tunneling TCP, utilisez une `curl` commande et assurez-vous d'obtenir une 2xx ou une 3xx réponse.

Par exemple, si l'URL de votre proxy Web est `https://server.com:1235`, utilisez un `proxy-insecure` indicateur avec la `curl` commande, car le proxy Web peut s'appuyer sur un certificat auto-signé.

```
export HTTPS_PROXY=https://server.com:1235
curl -I https://aws.amazon.com --proxy-insecure
```

Si l'URL de votre proxy Web possède un `http` port (par exemple, `http://server.com:1234`), vous n'êtes pas obligé d'utiliser l'`proxy-insecure` indicateur.

```
export HTTPS_PROXY=http://server.com:1234
curl -I https://aws.amazon.com
```

Configuration et démarrage du proxy local

Pour configurer le proxy local afin qu'il utilise un proxy Web, vous devez configurer la variable d'`HTTPS_PROXY` environnement avec les noms de domaine DNS ou les adresses IP et les numéros de port utilisés par votre proxy Web.

Après avoir configuré le proxy local, vous pouvez utiliser le proxy local comme expliqué dans ce document [README](#).

Note

Votre déclaration de variable d'environnement est sensible à la casse. Nous vous recommandons de définir chaque variable une seule fois en majuscules ou en minuscules. Les exemples suivants montrent la variable d'environnement déclarée en lettres majuscules. Si la même variable est spécifiée à la fois en majuscules et en minuscules, la variable spécifiée en lettres minuscules a la priorité.

Les commandes suivantes indiquent comment configurer le proxy local qui s'exécute sur votre destination pour utiliser le proxy Web et démarrer le proxy local.

- `AWSIOT_TUNNEL_ACCESS_TOKEN`: Cette variable contient le jeton d'accès client (CAT) pour la destination.
- `HTTPS_PROXY`: cette variable contient l'URL du proxy Web ou l'adresse IP pour configurer le proxy local.

Les commandes présentées dans les exemples suivants dépendent du système d'exploitation que vous utilisez et du fait que le proxy Web écoute sur un port HTTP ou HTTPS.

Proxy Web écoutant sur un port HTTP

Si votre proxy Web écoute sur un port HTTP, vous pouvez fournir l'URL ou l'adresse IP du proxy Web pour la `HTTPS_PROXY` variable.

Linux/macOS

Sous Linux ou macOS, exécutez les commandes suivantes dans le terminal pour configurer et démarrer le proxy local sur votre destination afin d'utiliser un proxy Web écoutant un port HTTP.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
export HTTPS_PROXY=http://proxy.example.com:1234  
.localproxy -r us-east-1 -d 22
```

Si vous devez vous authentifier auprès du proxy, vous devez spécifier un**username** et dans le**password** cadre de laHTTPS_PROXY variable.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
export HTTPS_PROXY=http://username:password@proxy.example.com:1234  
.localproxy -r us-east-1 -d 22
```

Windows

Sous Windows, vous configurez le proxy local de la même manière que pour Linux ou macOS, mais la façon dont vous définissez les variables d'environnement est différente de celle des autres plateformes. Exécutez les commandes suivantes dans lacmd fenêtre pour configurer et démarrer le proxy local sur votre destination afin d'utiliser un proxy Web écoutant un port HTTP.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
set HTTPS_PROXY=http://proxy.example.com:1234  
.localproxy -r us-east-1 -d 22
```

Si vous devez vous authentifier auprès du proxy, vous devez spécifier un**username** et dans le**password** cadre de laHTTPS_PROXY variable.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
set HTTPS_PROXY=http://username:password@10.15.20.25:1234  
.localproxy -r us-east-1 -d 22
```

Proxy Web écoutant sur un port HTTPS

Exécutez les commandes suivantes si votre proxy Web écoute sur un port HTTPS.

Note

Si vous utilisez un certificat auto-signé pour le proxy Web ou si vous exécutez le proxy local sur un système d'exploitation qui ne prend pas en charge OpenSSL de manière native et ne dispose pas de configurations par défaut, vous devrez configurer vos certificats de proxy Web comme décrit dans la section [Configuration des certificats](#) du GitHub référentiel.

Les commandes suivantes ressembleront à la façon dont vous avez configuré votre proxy Web pour un proxy HTTP, à l'exception du fait que vous devez également spécifier le chemin d'accès aux fichiers de certificat que vous avez installés comme décrit précédemment.

Linux/macOS

Sous Linux ou macOS, exécutez les commandes suivantes dans le terminal pour configurer le proxy local exécuté sur votre destination afin qu'il utilise un proxy Web écoutant un port HTTPS.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
export HTTPS_PROXY=http://proxy.example.com:1234  
.localproxy -r us-east-1 -d 22 -c /path/to/certs
```

Si vous devez vous authentifier auprès du proxy, vous devez spécifier un**username** et dans le**password** cadre de laHTTPS_PROXY variable.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
export HTTPS_PROXY=http://username:password@proxy.example.com:1234  
./localproxy -r us-east-1 -d 22 -c /path/to/certs
```

Windows

Dans Windows, exécutez les commandes suivantes dans lacmd fenêtre pour configurer et démarrer le proxy local exécuté sur votre destination afin d'utiliser un proxy Web écoutant un port HTTP.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
set HTTPS_PROXY=http://proxy.example.com:1234  
.\localproxy -r us-east-1 -d 22 -c \path\to\certs
```

Si vous devez vous authentifier auprès du proxy, vous devez spécifier un**username** et dans le**password** cadre de laHTTPS_PROXY variable.

```
set AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
set HTTPS_PROXY=http://username:password@10.15.20.25:1234  
.\localproxy -r us-east-1 -d 22 -c \path\to\certs
```

Exemple de commande et de sortie

Ce qui suit montre un exemple de commande que vous exécutez sur un système d'exploitation Linux et la sortie correspondante. L'exemple montre un proxy Web qui écoute sur un port HTTP et comment le proxy local peut être configuré pour utiliser le proxy Web dans source les deux destination modes. Avant de pouvoir exécuter ces commandes, vous devez avoir ouvert un tunnel et obtenu les jetons d'accès client pour la source et la destination. Vous devez également avoir créé le proxy local et configuré votre proxy Web comme décrit précédemment.

Voici une vue d'ensemble des étapes à suivre après avoir démarré le proxy local. Le proxy local :

- Identifie l'URL du proxy Web afin qu'il puisse l'utiliser pour se connecter au serveur proxy.
- Établit une connexion TCP avec le proxy Web.
- Envoie uneCONNECT requête HTTP au proxy Web et attend laHTTP/1.1 200 réponse, qui indique que la connexion a été établie.
- Met à niveau le protocole HTTPS WebSockets pour établir une connexion de longue durée.
- Commence à transmettre des données via la connexion aux points de terminaison sécurisés du dispositif de tunneling.

Note

Les commandes suivantes utilisées dans les exemples utilisent l'`verbosity` indicateur pour illustrer une vue d'ensemble des différentes étapes décrites précédemment après l'exécution du proxy local. Nous vous recommandons de n'utiliser cet indicateur qu'à des fins de test.

Exécution d'un proxy local en mode source

Les commandes suivantes montrent comment exécuter le proxy local en mode source.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}  
export HTTPS_PROXY=http://username:password@10.15.10.25:1234
```

```
./localproxy -s 5555 -v 5 -r us-west-2
```

Ce qui suit montre un exemple de sortie de l'exécution du proxy local en source mode.

```
...
Parsed basic auth credentials for the URL
Found Web proxy information in the environment variables, will use it to connect via the proxy.
...
Starting proxy in source mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-west-2.amazonaws.com:443
Resolved Web proxy IP: 10.10.0.11
Connected successfully with Web Proxy
Successfully sent HTTP CONNECT to the Web proxy
Full response from the Web proxy:
HTTP/1.1 200 Connection established
TCP tunnel established successfully
Connected successfully with proxy server
Successfully completed SSL handshake with proxy server
Web socket session ID: 0a109afffee745f5-00001341-000b8138-cc6c878d80e8adb0-f186064b
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://data.tunneling.iot.us-west-2.amazonaws.com:443
Setting up web socket pings for every 5000 milliseconds
Scheduled next read:
...
Starting web socket read loop continue reading...
Resolved bind IP: 127.0.0.1
Listening for new connection on port 5555
```

Exécution d'un proxy local en mode destination

Les commandes suivantes montrent comment exécuter le proxy local en mode destination.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=${access_token}
export HTTPS_PROXY=http:username:password@10.15.10.25:1234
./localproxy -d 22 -v 5 -r us-west-2
```

Ce qui suit montre un exemple de sortie de l'exécution du proxy local en destination mode.

```
...
Parsed basic auth credentials for the URL
Found Web proxy information in the environment variables, will use it to connect via the proxy.
...
Starting proxy in destination mode
Attempting to establish web socket connection with endpoint wss://data.tunneling.iot.us-west-2.amazonaws.com:443
Resolved Web proxy IP: 10.10.0.1
Connected successfully with Web Proxy
Successfully sent HTTP CONNECT to the Web proxy
Full response from the Web proxy:
HTTP/1.1 200 Connection established
```

```
TCP tunnel established successfully
Connected successfully with proxy server
Successfully completed SSL handshake with proxy server
Web socket session ID: 06717bffffed3fd05-00001355-000b8315-da3109a85da804dd-24c3d10d
Web socket subprotocol selected: aws.iot.securetunneling-2.0
Successfully established websocket connection with proxy server: wss://
data.tunneling.iot.us-west-2.amazonaws.com:443
Seting up web socket pings for every 5000 milliseconds
Scheduled next read:

...
Starting web socket read loop continue reading...
```

Multiplex des flux de données et utilisation de connexions TCP simultanées dans un tunnel sécurisé

Vous pouvez utiliser plusieurs flux de données par tunnel à l'aide de la fonction de multiplexage par tunneling sécurisé. Le multiplexage vous permet de dépanner des appareils utilisant plusieurs flux de données. Vous pouvez également réduire votre charge opérationnelle en éliminant la nécessité de créer, de déployer et de démarrer plusieurs proxys locaux ou d'ouvrir plusieurs tunnels vers le même appareil. Par exemple, le multiplexage peut être utilisé dans le cas d'un navigateur Web qui nécessite l'envoi de plusieurs flux de données HTTP et SSH.

Pour chaque flux de données, le tunnelingAWS IoT sécurisé prend en charge les connexions TCP simultanées. L'utilisation de connexions simultanées réduit le risque de délai d'attente en cas de demandes multiples de la part du client. Par exemple, cela peut réduire le temps de chargement lors de l'accès à distance à un serveur Web local à l'appareil de destination.

Les sections suivantes expliquent plus en détail le multiplexage et l'utilisation de connexions TCP simultanées, ainsi que leurs différents cas d'utilisation.

Rubriques

- [Multiplexage de plusieurs flux de données dans un tunnel sécurisé \(p. 878\)](#)
- [Utilisation de connexions TCP simultanées dans un tunnel sécurisé \(p. 881\)](#)

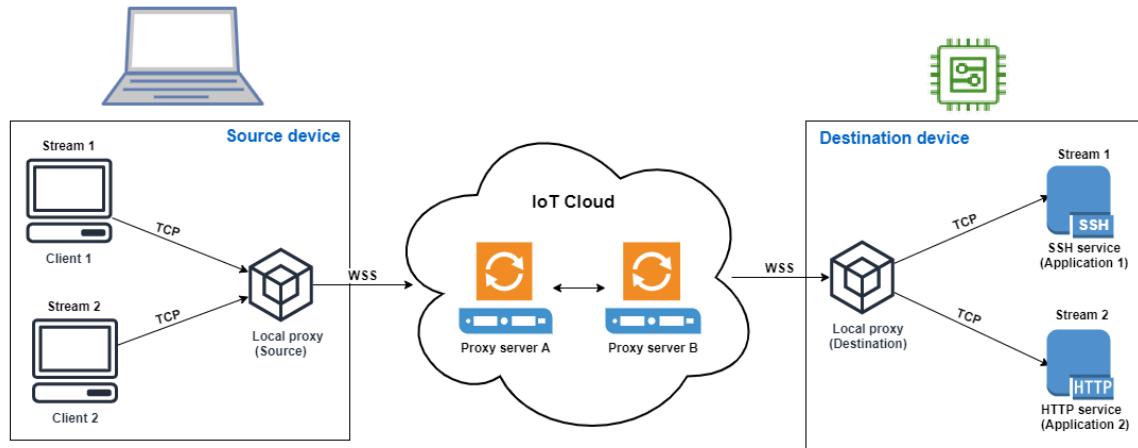
Multiplexage de plusieurs flux de données dans un tunnel sécurisé

Vous pouvez utiliser la fonction de multiplexage pour les appareils qui utilisent plusieurs connexions ou ports. Le multiplexage peut également être utilisé lorsque vous avez besoin de plusieurs connexions à un périphérique distant pour résoudre des problèmes. Par exemple, il peut être utilisé dans le cas d'un navigateur Web qui nécessite l'envoi de plusieurs flux de données HTTP et SSH. Les données d'application des deux flux sont envoyées simultanément au dispositif via le tunnel multiplexé.

Exemple de cas d'utilisation

Supposons que vous deviez vous connecter à une application Web intégrée à l'appareil pour modifier certains paramètres réseau, tout en émettant 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 à l'appareil 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 de multiplexage, ces deux flux indépendants peuvent être transférés simultanément via le même tunnel.



Configuration d'un tunnel multiplexé

La procédure suivante explique comment configurer un tunnel multiplexé pour dépanner des périphériques utilisant des applications nécessitant des connexions à plusieurs ports. Vous allez configurer un tunnel avec deux flux multiplexés : un flux HTTP et un flux SSH.

1. (Facultatif) Création de fichiers de configuration

Vous pouvez éventuellement configurer le périphérique source et le périphérique de destination à l'aide de fichiers de configuration. Utilisez des fichiers de configuration si les mappages de ports sont susceptibles de changer fréquemment. Vous pouvez ignorer cette étape si vous préférez spécifier le mappage des ports de manière explicite à l'aide de la CLI, ou si vous n'avez pas besoin de démarrer le proxy local sur les ports d'écoute désignés. Pour de plus amples informations sur l'utilisation des fichiers de configuration, consultez la section [Options définies via --config](#) dans GitHub.

- Sur votre appareil source, dans le dossier dans lequel votre proxy local sera exécuté, créez un dossier de configuration appelé `Config`. Dans ce dossier, créez un fichier appelé `SSHSource.ini` avec le contenu suivant :

```
HTTP1 = 5555
SSH1 = 3333
```

- Sur votre appareil de destination, dans le dossier dans lequel votre proxy local sera exécuté, créez un dossier de configuration appelé `Config`. Dans ce dossier, créez un fichier appelé `SSHDestination.ini` avec le contenu suivant :

```
HTTP1 = 80
SSH1 = 22
```

2. Ouvrir un tunnel

Ouvrez un tunnel à l'aide de l'opération `OpenTunnel` API ou de la commande `open-tunnel` CLI. Configurez la destination en spécifiant `SSH1` et `enHTTP1` tant que services et le nom de l'AWS IoT object qui correspond à votre appareil distant. Vos applications SSH et HTTP s'exécutent sur cet appareil distant. Vous devez déjà avoir créé l'objet IoT dans le AWS IoT registre. Pour plus d'informations, veuillez consulter [Comment gérer des objets avec le registre \(p. 287\)](#).

```
aws iotsecuretunneling open-tunnel \
```

```
--destination-config thingName=RemoteDevice1,services=HTTP1,SSH1
```

L'exécution de cette commande génère les jetons d'accès source et de destination que vous utiliserez pour exécuter le proxy local.

```
{
  "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
}
```

3. Configuration et démarrage du proxy local

Avant de pouvoir exécuter le proxy local, configurez le AWS IoT Device Client ou téléchargez le code source du proxy local [GitHub](#) et créez-le pour la plate-forme de votre choix. Vous pouvez ensuite démarrer la destination et le proxy local source pour vous connecter au tunnel sécurisé. Pour de plus amples informations sur la configuration et l'utilisation du proxy local, veuillez consulter [Comment utiliser le proxy local \(p. 868\)](#).

Note

Sur votre appareil source, si vous n'utilisez aucun fichier de configuration ou si vous ne spécifiez pas le mappage des ports à l'aide de la CLI, vous pouvez toujours utiliser la même commande pour exécuter le proxy local. Le proxy local en mode source sélectionnera automatiquement les ports disponibles à utiliser et les mappages pour vous.

Start local proxy using configuration files

Exécutez les commandes suivantes pour exécuter le proxy local dans les modes source et destination à l'aide de fichiers de configuration.

```
// ----- Start the destination local proxy -----
./localproxy -r us-east-1 -m dst -t destination_client_access_token

// ----- Start the source local proxy -----
// You also run the same command below if you want the local proxy to
// choose the mappings for you instead of using configuration files.
./localproxy -r us-east-1 -m src -t source_client_access_token
```

Start local proxy using CLI port mapping

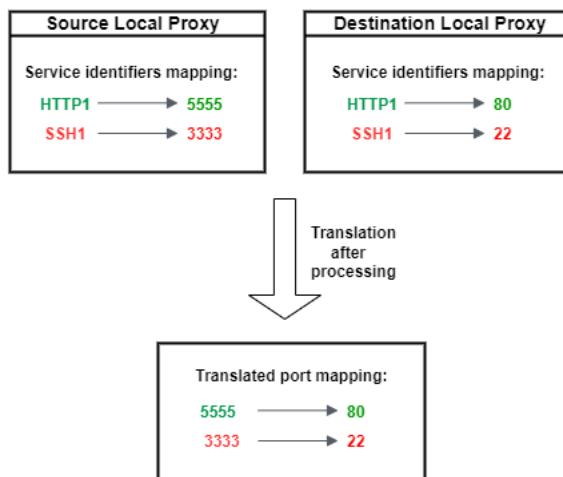
Exécutez les commandes suivantes pour exécuter le proxy local dans les modes source et destination en spécifiant explicitement les mappages de ports à l'aide de l'interface de ligne de commande.

```
// ----- Start the destination local proxy
-----
./localproxy -r us-east-1 -d HTTP1=80,SSH1=22 -t destination_client_access_token

// ----- Start the source local proxy
-----
./localproxy -r us-east-1 -s HTTP1=5555,SSH1=33 -t source_client_access_token
```

Les données d'application issues des connexions SSH et HTTP peuvent désormais être transférées simultanément via le tunnel multiplexé. Comme le montre la carte ci-dessous, l'identifiant de service agit comme un format lisible pour traduire le mappage des ports entre le périphérique source et le périphérique

de destination. Avec cette configuration, le tunneling sécurisé transfère tout trafic HTTP entrant du port **5555** du périphérique source vers le port **80** du périphérique de destination, et tout trafic SSH entrant du port **3333** vers le port **22** du périphérique de destination.



Utilisation de connexions TCP simultanées dans un tunnel sécurisé

AWS IoT tunneling sécurisé prend en charge plusieurs connexions TCP simultanément pour chaque flux de données. Vous pouvez utiliser cette fonctionnalité lorsque vous avez besoin de connexions simultanées à un appareil distant. L'utilisation de connexions TCP simultanées réduit le risque de délai d'attente en cas de demandes multiples émanant du client. Par exemple, lorsque vous accédez à un serveur Web sur lequel plusieurs composants s'exécutent, des connexions TCP simultanées peuvent réduire le temps de chargement du site.

Note

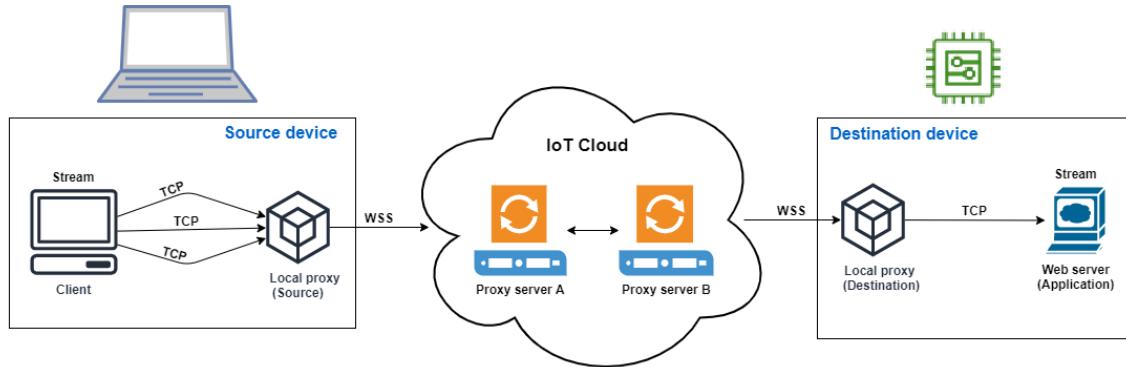
Les connexions TCP simultanées ont une limite de bande passante de 800 kilo-octets par seconde pour chacuneCompte AWS. AWS IoT tunneling sécurisé peut configurer cette limite pour vous en fonction du nombre de demandes entrantes.

Exemple de cas d'utilisation

Supposons que vous deviez accéder à distance à un serveur Web local à l'appareil de destination et sur lequel plusieurs composants s'exécutent. Avec une seule connexion TCP, tout en essayant d'accéder au serveur Web, le chargement séquentiel peut augmenter le temps nécessaire pour charger les ressources sur le site. Les connexions TCP simultanées peuvent réduire le temps de chargement en répondant aux besoins en ressources du site, réduisant ainsi le temps d'accès. Le schéma suivant montre comment les connexions TCP simultanées sont prises en charge pour le flux de données vers l'application de serveur Web exécutée sur le périphérique distant.

Note

Si vous souhaitez accéder à plusieurs applications exécutées sur le périphérique distant à l'aide du tunnel, vous pouvez utiliser le multiplexage par tunnel. Pour plus d'informations, veuillez consulter [Multiplexage de plusieurs flux de données dans un tunnel sécurisé \(p. 878\)](#).



Comment utiliser des connexions TCP simultanées

La procédure suivante explique comment utiliser des connexions TCP simultanées pour accéder au navigateur Web du périphérique distant. Lorsque le client reçoit plusieurs demandes, le tunnelingAWS IoT sécurisé configure automatiquement des connexions TCP simultanées pour traiter les demandes, réduisant ainsi le temps de chargement.

1. Ouvrir un tunnel

Ouvrez un tunnel à l'aide de l'opération `OpenTunnel` API ou de la commande `open-tunnel` CLI. Configurez la destination en spécifiant `HTTP` comme service et le nom de l'AWS IoT objet qui correspond à votre appareil distant. Votre application de serveur Web est en cours d'exécution sur cet appareil distant. Vous devez déjà avoir créé l'objet IoT dans le AWS IoT registre. Pour plus d'informations, veuillez consulter [Comment gérer des objets avec le registre \(p. 287\)](#).

```
aws iotsecuretunneling open-tunnel \
--destination-config thingName=RemoteDevice1,service=HTTP
```

L'exécution de cette commande génère les jetons d'accès source et de destination que vous utiliserez pour exécuter le proxy local.

```
{
  "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"
}
```

2. Configuration et démarrage du proxy local

Avant de pouvoir exécuter le proxy local, téléchargez le code source du proxy local [GitHub](#) et créez-le pour la plate-forme de votre choix. Vous pouvez ensuite démarrer le proxy local de destination et de source pour vous connecter au tunnel sécurisé et commencer à utiliser l'application de serveur Web distant.

Note

Pour que le tunnelingAWS IoT sécurisé utilise des connexions TCP simultanées, vous devez effectuer une mise à niveau vers la dernière version du proxy local. Cette fonctionnalité n'est pas disponible si vous configurez le proxy local à l'aide du AWS IoT Device Client.

```
// Start the destination local proxy
```

```
./localproxy -r us-east-1 -d HTTP=80 -t destination_client_access_token  
// Start the source local proxy  
./localproxy -r us-east-1 -s HTTP=5555 -t source_client_access_token
```

Pour de plus amples informations sur la configuration et l'utilisation du proxy local, veuillez consulter [Comment utiliser le proxy local \(p. 868\)](#).

Vous pouvez désormais utiliser le tunnel pour accéder à l'application du serveur Web. AWS IoT tunneling sécurisé configurera et gérera automatiquement les connexions TCP simultanées en cas de demandes multiples de la part du client.

Configuration d'un appareil distant et utilisation d'un 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 qu'un tunnel sécurisé puisse délivrer le jeton d'accès client à l'aide de MQTT. L'agent IoT doit s'abonner à la rubrique MQTT IoT réservée suivante :

Note

Si vous souhaitez transmettre le jeton d'accès client de destination au périphérique distant par des méthodes autres que l'abonnement à la rubrique MQTT réservée, vous aurez peut-être besoin d'un écouteur CAT (Destination Client Access Token) et d'un proxy local. L'écouteur CAT doit fonctionner avec le mécanisme de distribution de jetons d'accès client que vous avez choisi et être capable de démarrer un proxy local en mode destination.

Extrait de l'agent IoT

L'agent IoT doit s'abonner à la rubrique réservée suivante sur l'IoT MQTT pour pouvoir recevoir le message MQTT et démarrer le proxy local :

```
$aws/things/thing-name/tunnels/notify
```

Où sething-name trouve le nom de l'AWS IoT objet 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 [SDK AWS IoT Device](#) et [ProcessBuilder](#) la bibliothèque Java pour créer un agent IoT simple capable de fonctionner avec un 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);
        }
    }
}
```

Contrôle de l'accès aux tunnels

Secure Tunneling fournit des actions, des ressources et des clés de contexte de condition spécifiques au service en vue de leur utilisation dans les stratégies d'autorisation IAM.

Conditions préalables à l'accès au tunnel

- Découvrez comment sécuriser les AWS ressources à l'aide de [politiques IAM](#).
- Découvrez comment créer et évaluer des [conditions IAM](#).
- Découvrez comment sécuriser les AWS ressources à l'aide de [balises de ressources](#).

stratégies d'accès au tunnel

Vous devez appliquer les règles suivantes pour autoriser les autorisations d'utilisation de l'API de tunneling sécurisée. Pour de plus amples informations sur les groupes de sécurité AWS IoT, veuillez consulter [Gestion des identités et des accès pour AWS IoT \(p. 414\)](#).

IoT :OpenTunnel

L'action de stratégie `iot:OpenTunnel` accorde à un mandataire l'autorisation d'appeler [OpenTunnel](#).

Dans l'`Resource` élément de la déclaration de politique de l'IAM :

- Spécifiez l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

- Spécifiez l'ARN d'un objet pour gérer les `OpenTunnel` autorisations relatives à des objets IoT spécifiques :

`arn:aws:iot:aws-region:aws-account-id:thing/thing-name`

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 politique suivante vous permet d'ouvrir un tunnel vers l'objet si l'objet appartient à un groupe d'objets dont le nom commence par `TestGroup` et que 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. Pour obtenir des informations générales sur l'utilisation de balises, veuillez consulter [Balisage de vos ressources AWS IoT \(p. 310\)](#).

```
{
    "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 :RotateTunnelAccessToken

L'action de stratégie `iot:RotateTunnelAccessToken` accorde à un mandataire l'autorisation d'appeler [RotateTunnelAccessToken](#).

Dans l'`Resource` élément de la déclaration de politique de l'IAM :

- Spécifiez un ARN de tunnel entièrement qualifié :

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

Vous pouvez également utiliser l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

- Spécifiez l'ARN d'un objet pour gérer lesRotateTunnelAccessToken autorisations relatives à des objets IoT spécifiques :

```
arn:aws:iot:aws-region:aws-account-id:thing/thing-name
```

Par exemple, la déclaration de politique suivante vous permet de faire alterner le jeton d'accès source d'un tunnel ou le jeton d'accès de destination d'un client pour l'objet IoT nomméTestDevice.

```
{
    "Effect": "Allow",
    "Action": "iot:RotateTunnelAccessToken",
    "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:RotateTunnelAccessToken` prend en charge les clés de condition suivantes :

- `iot:ThingGroupArn`
- `iot:TunnelDestinationService`
- `iot:ClientMode`
- `aws:SecureTransport`

La déclaration de politique suivante vous permet de faire pivoter le jeton d'accès de destination vers l'objet si l'objet appartient à un groupe d'objets dont le nom commence par**TestGroup**, si le service de destination configuré sur le tunnel est SSH et si le client est en**DESTINATION** mode.

```
{
    "Effect": "Allow",
    "Action": "iot:RotateTunnelAccessToken",
    "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"
            ],
            "iot:ClientMode": "DESTINATION"
        }
    }
}
```

IoT :DescribeTunnel

L'action de stratégie `iot:DescribeTunnel` accorde à un mandataire l'autorisation d'appeler [DescribeTunnel](#).

Dans l'`Resource` élément de la déclaration de politique IAM, spécifiez un ARN de tunnel entièrement qualifié :

```
arn:aws:iot:aws-region:aws-account-id:tunnel/tunnel-id
```

Vous pouvez également utiliser l'ARN générique :

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

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](#).

Dans l'`Resource` élément de la déclaration de politique de l'IAM :

- Spécifiez l'ARN du tunnel générique :

```
arn:aws:iot:aws-region:aws-account-id:tunnel/*
```

- Spécifiez un ARN d'objet pour gérer les `ListTunnels` autorisations sur des objets IoT sélectionnés :

```
arn:aws:iot:aws-region:aws-account-id:thing/thing-name
```

L'action `iot>ListTunnels` politique soutient la clé de condition `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](#).

Dans l'`Resource` élément de la déclaration de politique IAM, spécifiez un ARN de tunnel entièrement qualifié :

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

Vous pouvez également utiliser l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

L'action `iot>ListTagsForResource` politique soutient la clé de condition `aws:SecureTransport`.

IoT :CloseTunnel

L'action de stratégie `iot:CloseTunnel` accorde à un mandataire l'autorisation d'appeler [CloseTunnel](#).

Dans l'`Resource` élément de la déclaration de politique IAM, spécifiez un ARN de tunnel entièrement qualifié :

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

Vous pouvez également utiliser l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

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 politique suivante vous permet d'appeler `CloseTunnel` si le `Delete` paramètre de la demande est `false` et si la demande est marquée avec la 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`.

Dans l'`Resource` élément de la déclaration de politique IAM, spécifiez un ARN de tunnel entièrement qualifié :

`arn:aws:iot:aws-region: aws-account-id:tunnel/tunnel-id`

Vous pouvez également utiliser l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

L'action `iot:TagResource` politique soutient la clé de condition `aws:SecureTransport`.

IoT :UntagResource

L'action de stratégie `iot:UntagResource` accorde à un mandataire l'autorisation d'appeler `UntagResource`.

Dans l'`Resource` élément de la déclaration de politique IAM, spécifiez un ARN de tunnel entièrement qualifié :

`arn:aws:iot:aws-region:aws-account-id:tunnel/tunnel-id`

Vous pouvez également utiliser l'ARN du tunnel générique :

`arn:aws:iot:aws-region:aws-account-id:tunnel/*`

L'action `iot:UntagResource` politique soutient la clé de condition `aws:SecureTransport`.

Résolution des problèmes de connectivité par tunneling AWS IoT sécurisé en alternant les jetons d'accès client

Lorsque vous utilisez le tunneling AWS IoT sécurisé, vous pouvez rencontrer des problèmes de connectivité même si le tunnel est ouvert. Les sections suivantes présentent certains problèmes possibles et la manière dont vous pouvez les résoudre en alternant les jetons d'accès client. Pour faire pivoter le jeton d'accès client (CAT), utilisez l'[RotateTunnelAccessToken API](#) ou le [rotate-tunnel-access-token AWS CLI](#). Selon que vous rencontrez une erreur lors de l'utilisation du client en mode source ou destination, vous pouvez faire pivoter le CAT en mode source ou destination, ou les deux.

Note

- Si vous ne savez pas si le CAT doit être pivoté sur la source ou sur la destination, vous pouvez faire pivoter le CAT à la fois sur la source et sur la destination en le `ClientMode` réglant sur `ALL` lorsque vous utilisez l'`RotateTunnelAccessToken API`.
- La rotation du CAT ne prolonge pas la durée du tunnel. Supposons, par exemple, que la durée du tunnel soit de 12 heures et que le tunnel soit déjà ouvert depuis 4 heures. Lorsque vous alternez les jetons d'accès, les nouveaux jetons générés ne peuvent être utilisés que pendant les 8 heures restantes.

Rubriques

- [Erreur de jeton d'accès au client non valide \(p. 890\)](#)
- [Erreur de non-correspondance du jeton client \(p. 891\)](#)
- [Problèmes de connectivité de l'appareil à distance \(p. 892\)](#)

Erreur de jeton d'accès au client non valide

Lorsque vous utilisez le tunneling AWS IoT sécurisé, vous pouvez rencontrer une erreur de connexion lorsque vous utilisez le même jeton d'accès client (CAT) pour vous reconnecter au même tunnel. Dans ce cas, le proxy local ne peut pas se connecter au serveur proxy de tunneling sécurisé. Si vous utilisez un client en mode source, le message d'erreur suivant peut s'afficher :

Invalid access token: The access token was previously used and cannot be used again

L'erreur se produit car le jeton d'accès client (CAT) ne peut être utilisé qu'une seule fois par le proxy local, puis il devient invalide. Pour résoudre cette erreur, faites pivoter le jeton d'accès client dans le SOURCE mode de génération d'un nouveau CAT pour la source. Pour obtenir un exemple pratique illustrant la façon de faire tourner le CAT source, veuillez consulter [Exemple de CAT faisant pivoter la source \(p. 891\)](#).

Erreur de non-correspondance du jeton client

Note

L'utilisation de jetons clients pour réutiliser le CAT n'est pas recommandée. Nous vous recommandons plutôt d'utiliser l'`RotateTunnelAccessTokenAPI` pour alterner les jetons d'accès client afin de vous reconnecter au tunnel.

Si vous utilisez des jetons clients, vous pouvez réutiliser le CAT pour vous reconnecter au tunnel. Pour réutiliser le CAT, vous devez fournir le jeton client avec le CAT la première fois que vous vous connectez au tunnel sécurisé. Le tunneling sécurisé stocke le jeton client. Ainsi, pour les tentatives de connexion ultérieures utilisant le même jeton, le jeton client doit également être fourni. Pour plus d'informations sur l'utilisation des jetons clients, consultez [l'implémentation de référence du proxy local dans GitHub](#).

Lorsque vous utilisez des jetons client, si vous utilisez un client en mode source, le message d'erreur suivant peut s'afficher :

```
Invalid client token: The provided client token does not match the client token  
that was previously set.
```

L'erreur se produit car le jeton client fourni ne correspond pas au jeton client fourni avec le CAT lors de l'accès au tunnel. Pour résoudre cette erreur, faites pivoter le CAT dans le SOURCE mode afin de générer un nouveau CAT pour la source. Voici un exemple:

Exemple de CAT faisant pivoter la source

L'exemple suivant montre comment exécuter l'`RotateTunnelAccessTokenAPI` en SOURCE mode de génération d'un nouveau CAT pour la source :

```
aws iotsecuretunneling rotate-tunnel-access-token \  
--region <region> \  
--tunnel-id <tunnel-id> \  
--client-mode SOURCE
```

L'exécution de cette commande génère un nouveau jeton d'accès à la source et renvoie l'ARN de votre tunnel.

```
{  
    "sourceAccessToken": "<source-access-token>",  
    "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"  
}
```

Vous pouvez désormais utiliser le nouveau jeton source pour connecter le proxy local en mode source.

```
export AWSIOT_TUNNEL_ACCESS_TOKEN=<source-access-token>  
./localproxy -r <region> -s <port>
```

Vous pouvez voir ci-dessous un exemple de sortie de l'exécution du proxy local :

```
...
```

```
[info] Starting proxy in source mode
...
[info] Successfully established websocket connection with proxy server ...
[info] Listening for new connection on port <port>
...
```

Problèmes de connectivité de l'appareil à distance

Lorsque vous utilisez le tunneling AWS IoT sécurisé, le périphérique peut se déconnecter de manière inattendue, même si le tunnel est ouvert. Pour déterminer si un appareil est toujours connecté au tunnel, vous pouvez utiliser l'[DescribeTunnel](#) API ou le [tunnel descriptive](#) AWS CLI.

Un appareil peut être déconnecté pour plusieurs raisons. Pour résoudre le problème de connectivité, vous pouvez faire pivoter le CAT sur la destination si l'appareil a été déconnecté pour les raisons suivantes :

- Le CAT de destination n'est plus valide.
- Le jeton n'a pas été transmis à l'appareil via le sujet MQTT réservé au tunneling sécurisé :

```
$aws/things/<thing-name>/tunnels/notify
```

L'exemple suivant de indique comment résoudre ce problème :

Exemple de CAT pour faire pivoter la destination

Prenons l'exemple d'un appareil distant, `<RemoteThing1>`. Pour ouvrir un tunnel pour cet objet, vous pouvez utiliser la commande suivante :

```
aws iotsecuretunneling open-tunnel \
--region <region> \
--destination-config thingName=<RemoteThing1>,services=SSH
```

L'exécution de cette commande génère les détails du tunnel et le CAT pour votre source et votre destination.

```
{
  "sourceAccessToken": "<source-access-token>",
  "destinationAccessToken": "<destination-access-token>",
  "tunnelId": "<tunnel-id>",
  "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"
}
```

Toutefois, lorsque vous utilisez l'[DescribeTunnel](#) API, la sortie indique que l'appareil a été déconnecté, comme illustré ci-dessous :

```
aws iotsecuretunneling describe-tunnel \
--tunnel-id <tunnel-id> \
--region <region>
```

L'exécution de cette commande indique que l'appareil n'est toujours pas connecté.

```
{
  "tunnel": {
    ...
    "destinationConnectionState": {
      "status": "DISCONNECTED"
    }
  }
}
```

```
    },
    ...
}
```

Pour résoudre cette erreur, exécutez l'RotateTunnelAccessTokenAPI avec le client enDESTINATION mode et les configurations de la destination. L'exécution de cette commande révoque l'ancien jeton d'accès, génère un nouveau jeton et renvoie ce jeton à la rubrique MQTT :

```
$aws/things/<thing-name>/tunnels/notify
```

```
aws iotsecuretunneling rotate-tunnel-access-token \
--tunnel-id <tunnel-id> \
--client-mode DESTINATION \
--destination-config thingName=<RemoteThing1>,services=SSH \
--region <region>
```

L'exécution de cette commande génère le nouveau jeton d'accès comme indiqué ci-dessous. Le jeton est ensuite envoyé à l'appareil pour qu'il se connecte au tunnel, si l'agent du périphérique est correctement configuré.

```
{
    "destinationAccessToken": "destination-access-token",
    "tunnelArn": "arn:aws:iot:<region>:<account-id>:tunnel/<tunnel-id>"
}
```

Mise en service des appareils

AWS propose différentes manières de provisionner un appareil et d'y installer des certificats clients uniques. Cette section décrit chaque méthode et explique comment sélectionner celle qui convient le mieux à votre solution IoT. Ces options sont décrites en détail dans le livre blanc intitulé [Device Manufacturing and Provisioning with X.509 Certificates in AWS IoT Core](#)

Sélectionnez l'option qui correspond le mieux à votre situation

- Vous pouvez installer des certificats sur les appareils IoT avant qu'ils ne soient délivrés

Si vous pouvez installer en toute sécurité des certificats clients uniques sur vos appareils IoT avant qu'ils ne soient fournis à l'utilisateur final, vous devez utiliser le [just-in-time provisionnement \(JITP\) \(p. 903\)](#) ou l'[just-in-time enregistrement \(JITR\) \(p. 334\)](#).

À l'aide de JITP et JITR, l'autorité de certification (CA) utilisée pour signer le certificat de l'appareil est enregistrée AWS IoT et reconnue AWS IoT dès la première connexion de l'appareil. L'appareil est provisionné AWS IoT lors de sa première connexion à l'aide des détails de son modèle de provisionnement.

Pour plus d'informations sur un seul objet, le JITP, le JITR et le provisionnement en bloc d'appareils dotés de certificats uniques, consultez. [the section called “Mise en service d'appareils disposant de certificats d'appareils” \(p. 902\)](#)

- Les utilisateurs finaux ou les installateurs peuvent utiliser une application pour installer des certificats sur leurs appareils IoT

Si vous ne pouvez pas installer de manière sécurisée des certificats clients uniques sur votre appareil IoT avant qu'ils ne soient fournis à l'utilisateur final, mais que l'utilisateur final ou un installateur peut utiliser une application pour enregistrer les appareils et installer les certificats d'appareils uniques, vous souhaitez utiliser le processus de [provisionnement par utilisateur de confiance \(p. 898\)](#).

Le recours à un utilisateur de confiance, tel qu'un utilisateur final ou un installateur disposant d'un compte connu, peut simplifier le processus de fabrication des appareils. Au lieu d'un certificat client unique, les appareils disposent d'un certificat temporaire qui leur permet de se connecter AWS IoT pendant 5 minutes seulement. Au cours de cette fenêtre de 5 minutes, l'utilisateur de confiance 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 compromission du certificat.

Pour plus d'informations, veuillez consulter [the section called “Allocation par utilisateur approuvé” \(p. 898\)](#).

- 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, le processus de [provisionnement par réclamation \(p. 896\)](#) est une option. Grâce à ce processus, vos appareils IoT disposent d'un certificat de réclamation qui est partagé par les autres appareils du parc. La première fois qu'un appareil se connecte à l'aide d'un certificat de réclamation, l'AWS IoT enregistre à l'aide de son modèle de provisionnement et lui délivre son certificat client unique pour un accès ultérieur. AWS IoT

Cette option permet le provisionnement automatique d'un appareil lorsqu'il se connecte AWS IoT, mais elle peut présenter un risque plus important en cas de compromission du certificat de sinistre. Si un certificat de réclamation est compromis, vous pouvez le désactiver. La désactivation du certificat de réclamation empêche tous les appareils dotés de ce certificat de réclamation d'être enregistrés future. Toutefois, la désactivation du certificat de réclamation ne bloque pas les appareils déjà approvisionnés.

Pour plus d'informations, veuillez consulter [the section called “Allocation par revendication” \(p. 896\)](#).

Provisionner des appareils dans AWS 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 regroupés en groupes d'objets. Pour plus d'informations, veuillez consulter [Gestion des appareils avec AWS IoT \(p. 287\)](#).

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, veuillez consulter [Indexation de la flotte \(p. 929\)](#).

Note

Pour que Fleet Hub indexe les données d'état de connectivité de votre objet, provisionnez votre objet et configuez-le de manière à ce que le nom de l'objet corresponde à l'ID client utilisé dans la demande Connect.

- 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, veuillez consulter [Authentification \(p. 316\)](#).

- Une stratégie IoT.

Les politiques de l'IoT définissent les opérations qu'un appareil peut effectuer 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, veuillez consulter [Autorisation \(p. 355\)](#). 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 réclamation temporaire est transmise à l'appareil lors de la configuration du 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” \(p. 923\)](#)
 - [the section called “CreateKeysAndCertificate” \(p. 925\)](#)
 - [the section called “RegisterThing” \(p. 926\)](#)

Mise en service d'appareils qui ne disposent pas de certificats d'appareils à l'aide de la mise en service de flotte

En utilisant le provisionnement du AWS IoT parc, AWS IoT vous pouvez générer et délivrer en toute sécurité des certificats d'appareils et des clés privées à vos appareils lorsqu'ils se connectent AWS IoT pour la première fois. AWS IoT fournit des certificats clients signés par l'autorité de certification Amazon Root (CA).

Il existe deux façons d'utiliser la mise en service de flotte :

- [Approvisionnement par réclamation \(p. 896\)](#)
- [Approvisionnement par un utilisateur de confiance \(p. 898\)](#)

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 provisionnement. Cette API renvoie un ARN de modèle. Pour plus d'informations, veuillez consulter [API MQTT de mise en service des appareils \(p. 923\)](#).

Vous pouvez également créer un modèle d'allocation de parc dans la console AWS IoT.

- a. Dans le volet de navigation, choisissez Connect, puis choisissez Fleet Provisioning templates.

- b. Choisissez Créer un modèle et suivez les instructions.
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'utiliser des métriques et des journaux CloudWatch AWS IoT pour surveiller les utilisations abusives. Si vous détectez une utilisation abusive, désactivez le certificat de demande de provisionnement afin qu'il ne puisse pas être utilisé pour le provisionnement des appareils.

Pour initialiser l'appareil à utiliser

1. L'appareil utilise le [AWS IoT SDK pour appareils, kits SDK mobiles et client deAWS IoT l'appareil \(p. 1494\)](#) 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, les `certificateOwnershipToken` sont renvoyés [CreateCertificateFromCsr \(p. 923\)](#) et [CreateKeysAndCertificate \(p. 925\)](#) expirent au bout d'une heure. [RegisterThing \(p. 926\)](#) doit être appelé avant l'`certificateOwnershipToken` expiration. Si le certificat créé par [CreateCertificateFromCsr \(p. 923\)](#) ou n'[CreateKeysAndCertificate \(p. 925\)](#) a pas été activé et n'a pas été associé à une politique ou à un objet au moment de l'expiration du jeton, le certificat est supprimé. Si le jeton expire, l'appareil peut appeler [CreateCertificateFromCsr \(p. 923\)](#) ou à [CreateKeysAndCertificate \(p. 925\)](#) nouveau 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 \(p. 925\)](#) pour créer un nouveau certificat et une nouvelle clé privée à l'aide de l'autorité de AWS certification.

Ou
 - b. Appelez [CreateCertificateFromCsr \(p. 923\)](#) pour générer un certificat à partir d'une demande de signature de certificat dont la clé privée est sécurisée.
3. À partir de l'appareil,appelez [RegisterThing \(p. 926\)](#) pour enregistrer ce dernier auprès d'AWS IoT et créer des ressources cloud.

Le service Fleet Provisioning utilise un modèle de provisionnement pour définir et créer des ressources cloud telles que des IoT. Le modèle peut spécifier les attributs et les groupes auxquels appartient l'objet. Les groupes d'objets doivent exister pour que le nouvel objet puisse y être ajouté.

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. Pour ce faire, vous pouvez notamment fournir et gérer un compte pour l'utilisateur de confiance qui l'authentifie et lui donne accès aux AWS IoT fonctionnalités et aux opérations d'API requises pour effectuer cette procédure.

Avant de livrer l'appareil

1. *Appelez [CreateProvisioningTemplate](#) pour créer un modèle de provisionnement et renvoyez ses modèles `TemplateArn` et `TemplateName`.*
2. Créez un rôle IAM utilisé par un utilisateur de confiance pour lancer le processus de provisionnement. Le modèle de mise en service permet uniquement à cet utilisateur de mettre en service un appareil. Par exemple :

```
{
```

```
"Effect": "Allow",
"Action": [
    "iot:CreateProvisioningClaim"
],
"Resource": [
    "arn:aws:iot: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 ce faire, vous associez la politique AWSIoTThingsRegistration gérée à un rôle IAM (appelé rôle de provisionnement) qui fait confiance au principal de AWS IoT service.
4. Fournissez les moyens d'identifier vos utilisateurs de confiance, par exemple en leur fournissant un compte capable de les authentifier et d'autoriser leurs interactions avec les opérations d'AWSAPI 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 demande de provisionnement temporaire auprès de AWS IoT

Note

Pour des raisons de sécurité, le certificat de demande d'approvisionnement temporaire qui est `CreateProvisioningClaim` renvoyé expire au bout de 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 IoT SDK pour appareils, kits SDK mobiles et client de AWS IoT l'appareil \(p. 1494\)](#).
 5. L'appareil obtient un certificat permanent et une clé privée en utilisant l'une de ces options dans les cinq minutes suivant la connexion au AWS IoT certificat de demande de provisionnement temporaire. L'appareil utilisera le certificat et la clé que ces options renverront pour toute authentification future AWS IoT.
 - a. Appelez [CreateKeysAndCertificate \(p. 925\)](#) pour créer un nouveau certificat et une nouvelle clé privée à l'aide de l'autorité de AWS certification.
- Ou
- b. Appelez [CreateCertificateFromCsr \(p. 923\)](#) pour générer un certificat à partir d'une demande de certificat dont la clé privée est sécurisée.

Note

N'oubliez pas [CreateKeysAndCertificate \(p. 925\)](#) ou [CreateCertificateFromCsr \(p. 923\)](#) devez renvoyer un certificat valide dans les cinq minutes suivant la connexion AWS IoT avec le certificat de demande de provisionnement temporaire.

6. L'appareil appelle [RegisterThing \(p. 926\)](#) pour enregistrer l'appareil auprès d'AWS IoT et créer des ressources cloud.

Le service Fleet Provisioning utilise un modèle de provisionnement pour définir et créer des ressources cloud telles que des IoT. Le modèle peut spécifier les attributs et les groupes auxquels appartient l'objet. Les groupes d'objets doivent exister pour que le nouvel objet puisse y être ajouté.

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 avec une entrée et une sortie définies. Les fonctions Lambda sont hautement personnalisables. `allowProvisioning` et `parameterOverrides` sont nécessaires pour créer des hooks de pré-provisionnement. Pour plus d'informations sur la création de fonctions Lambda, consultez la section [Utilisation AWS Lambda avec l'interface de ligne de commande](#).

Voici un exemple de sortie de fonction Lambda :

```
{  
    "allowProvisioning": True,  
    "parameterOverrides": {  
        "incomingKey0": "incomingValue0",  
        "incomingKey1": "incomingValue1"  
    }  
}
```

2. AWS IoT utilise des politiques basées sur les ressources pour appeler Lambda. Vous devez donc AWS IoT autoriser l'appel de votre fonction Lambda.

Important

Veillez à inclure les clés contextuelles `source-arn` ou `source-account` dans les conditions globales des politiques associées à votre action Lambda afin d'éviter toute manipulation des autorisations. Pour de plus amples informations à ce sujet, veuillez consulter [Prévention du député confus entre services \(p. 368\)](#).

Voici un exemple d'utilisation de [add-permission](#) pour donner une 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 pré-provisionnement à un modèle à l'aide de la commande [create-provisioning-template](#) ou [update-provisioning-template](#).

L'exemple de CLI suivant utilise la [create-provisioning-template](#) pour créer un modèle de provisionnement doté de crochets de pré-provisionnement :

```
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 d'informations, veuillez consulter [Chargement de AWS CLI paramètres 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.
- Just-in-time provisioning (JITP) avec un modèle qui approvisionne 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 \(p. 903\)](#)
- [ust-in-timeApprovisionnement en J \(p. 903\)](#)
- [Enregistrement en bloc \(p. 907\)](#)

Mise en service d'un seul objet

Pour provisionner un objet, utilisez l'[RegisterThing](#) API ou la commande `register-thing` CLI. 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](#)"}).

Consultez [Mise en service des modèles \(p. 908\)](#).

[RegisterThing](#) ou `register-thing` renvoie les ARN des ressources et le texte du certificat qu'il a créé :

```
{  
    "certificatePem": "certificate-text",  
    "resourceArns": {  
        "PolicyLogicalName": "arn:aws:iot:us-west-2:123456789012:policy/2A6577675B7CD1823E271C7AAD8184F44630FFD7",  
        "certificate": "arn:aws:iot:us-west-2:123456789012:cert/cd82bb924d4c6ccbb14986dcba4f30d892cc6b3ce7ad5008ed6542eea2b049",  
        "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.

Just-in-time Approvisionnement en J

Vous pouvez utiliser le just-in-time approvisionnement (JITP) pour approvisionner vos appareils lorsqu'ils tentent de se connecter pour la première fois. AWS IoT 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 réussites et les erreurs de provisionnement sont enregistrées comme [Métriques de mise en service d'appareils \(p. 487\)](#) dans AmazonCloudWatch.

Rubriques

- [Présentation de JITP \(p. 903\)](#)
- [Enregistrer une autorité de certification à l'aide d'un modèle d'approvisionnement \(p. 906\)](#)
- [Enregistrer une autorité de certification en utilisant le nom du modèle de provisionnement \(p. 907\)](#)

Présentation de JITP

Lorsqu'un appareil tente de se connecter à AWS IoT l'aide d'un certificat signé par un certificat CA enregistré, AWS IoT charge le modèle à partir du certificat CA et l'utilise pour appeler [RegisterThing \(p. 926\)](#). Le flux de travail JITP enregistre d'abord un certificat avec une valeur d'état de PENDING_ACTIVATION. Lorsque le flux de provisionnement des appareils est terminé, le statut du certificat est remplacé par 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.

Note

Vous pouvez approvisionner des appareils à l'aide de la fonction de AWS IoT Core just-in-time provisionnement (JITP) sans avoir à envoyer l'intégralité de la chaîne de confiance lors de la première connexion d'un appareil à AWS IoT Core. La présentation du certificat CA est facultative, mais l'appareil doit envoyer l'extension [SNI \(Server Name Indication\)](#) lorsqu'il se connecte à AWS IoT Core.

Exemple de corps de modèle

Le fichier JSON suivant est un exemple de corps de modèle d'un modèle JITP complet.

```
{
  "Parameters": {
    "AWS::IoT::Certificate::CommonName": {
      "Type": "String"
    },
    "AWS::IoT::Certificate::SerialNumber": {
      "Type": "String"
    },
    "AWS::IoT::Certificate::Country": {
      "Type": "String"
    },
    "AWS::IoT::Certificate::Id": {
      "Type": "String"
    }
  },
  "Resources": {
    "thing": {
      "Type": "AWS::IoT::Thing",
      "Properties": {
        "ThingName": {
          "Ref": "AWS::IoT::Certificate::CommonName"
        },
        "AttributePayload": {
          "version": "v1",
          "serialNumber": {
            "Ref": "AWS::IoT::Certificate::SerialNumber"
          }
        }
      },
      "ThingTypeName": "lightBulb-versionA",
      "ThingGroups": [
        "v1-lightbulbs",
        "group1"
      ]
    }
  }
}
```

```
{
    "Ref":"AWS::IoT::Certificate::Country"
}
],
"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\"] }] }"
    }
}
}
```

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 de l'appareil échoue si le certificat ne possède pas toutes les propriétés mentionnées dans la Parameters section du templateBody. Par exemple, s'il AWS::IoT::Certificate::Country est inclus dans le modèle, mais que le certificat ne possède pas de Country propriété, le provisionnement de l'appareil échoue.

Vous pouvez également utiliser CloudTrail pour résoudre les problèmes liés à votre modèle JITP. Pour plus d'informations sur les métriques enregistrées dans AmazonCloudWatch, consultez [Métriques de mise en service d'appareils \(p. 487\)](#). Pour plus d'informations sur les modèles de provisionnement, consultez la section [Modèles de provisionnement \(p. 908\)](#).

Note

Au cours du processus de provisionnement, le just-in-time provisioning (JITP) appelle d'autres opérations de l'API du plan AWS IoT de contrôle. Ces appels peuvent dépasser les [quotas de AWS IoT limitation](#) définis pour votre compte et entraîner une limitation des appels. Contactez [AWS Support Client](#) pour augmenter vos quotas de limitation si nécessaire.

Enregistrer une autorité de certification à l'aide d'un modèle d'appvisionnement

Pour enregistrer une autorité de certification à l'aide d'un modèle de provisionnement complet, procédez comme suit :

1. Enregistrez votre modèle de provisionnement et les informations ARN du rôle comme dans l'exemple suivant sous forme de fichier JSON :

```
{
    "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        \"AWS::IoT::Certificate::ThingName\": {\r\n          \"Type\": \"AWS::IoT::Thing\", \"Properties\": {\r\n            \"ThingName\": {\r\n              \"Ref\": \"AWS::IoT::Certificate::CommonName\"\r\n            },\r\n            \"AttributePayload\": {\r\n              \"serialNumber\": {\r\n                \"Ref\": \"AWS::IoT::Certificate::SerialNumber\"\r\n              },\r\n              \"ThingType\": {\r\n                \"ThingType\": \"lightBulb-versionA\", \"ThingGroups\": [\r\n                  \"v1-lightbulbs\"\r\n                ],\r\n                \"Ref\": \"AWS::IoT::Certificate::Country\"\r\n              },\r\n              \"OverrideSettings\": {\r\n                \"AttributePayload\": \"MERGE\", \"ThingType\": \"REPLACE\", \"ThingGroups\": [\r\n                  \"DO NOTHING\"\r\n                ],\r\n                \"certificate\": {\r\n                  \"Type\": \"AWS::IoT::Certificate\", \"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                  \"policy\": {\r\n                    \"Type\": \"AWS::IoT::Policy\", \"Properties\": {\r\n                      \"PolicyDocument\": \"{\r\n                        \"Version\": \"2012-10-17\",\r\n                        \"Statement\": [\r\n                          {\r\n                            \"Effect\": \"Allow\",\r\n                            \"Action\": \"iot:Publish\",\r\n                            \"Resource\": \"arn:aws:iot:us-east-1:123456789012:topic/\\$topic\"\r\n                          }\r\n                        ]\r\n                      }\"\r\n                    }\r\n                  }\r\n                }\r\n              }\r\n            }\r\n          }\r\n        }\r\n      }\r\n    }\r\n  }\r\n}
```

Dans cet exemple, la valeur du `templateBody` champ doit être un objet JSON spécifié sous forme de chaîne échappée et ne peut utiliser que les valeurs de la [liste précédente \(p. 903\)](#). 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.

2. Enregistrez un certificat CA à l'aide de l'opération d'API [RegisterCACertificate ou de la commande CLI `register-ca-certificate`](#). Vous allez spécifier le répertoire du modèle de provisionnement et les informations ARN du rôle que vous avez enregistrées à l'étape précédente :

L'exemple suivant montre comment enregistrer un certificat CA en DEFAULT mode à l'aide de AWS CLI :

```
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
```

L'exemple suivant montre comment enregistrer un certificat CA en SNI_ONLY mode à l'aide de AWS CLI :

```
aws iot register-ca-certificate --ca-certificate file://your-ca-cert --certificate-mode SNI_ONLY
                                         --set-as-active --allow-auto-registration --registration-config
                                         file://your-template
```

Pour plus d'informations, consultez la section [Enregistrer vos certificats CA](#).

3. (Facultatif) Mettez à jour les paramètres d'un certificat CA à l'aide de l'opération d'API [UpdateCACertificate ou](#) de la commande CLI. [update-ca-certificate](#)

Voici un exemple de mise à jour d'un certificat CA à l'aide de AWS CLI :

```
aws iot update-ca-certificate --certificate-id caCertificateId
                               --new-auto-registration-status ENABLE --registration-config
                               file://your-template
```

Enregistrer une autorité de certification en utilisant le nom du modèle de provisionnement

Pour enregistrer une autorité de certification à l'aide d'un nom de modèle de provisionnement, procédez comme suit :

1. Enregistrez le corps de votre modèle de provisionnement sous forme de fichier JSON. Vous pouvez trouver un exemple de corps de modèle dans [Exemple de corps de modèle \(p. 904\)](#).
2. Pour créer un modèle de provisionnement, utilisez l'[CreateProvisioningTemplateAPI](#) ou la [create-provisioning-template](#) commande CLI :

```
aws iot create-provisioning-template --template-name your-template-name \
                                         --template-body file://your-template-body.json --type JITP \
                                         --provisioning-role-arn arn:aws:iam::123456789012:role/test
```

Note

Pour le just-in-time provisionnement (JITP), vous devez spécifier le type de modèle à utiliser JITP lors de la création du modèle de provisionnement. Pour plus d'informations sur le type de modèle, consultez [CreateProvisioningTemplate](#) la référence de l'AWS API.

3. Pour enregistrer une autorité de certification avec le nom du modèle, utilisez l'API [RegisterCACertificate](#) ou la commande CLI : [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
                                         templateName=your-template-name
```

Enregistrement en bloc

Vous pouvez utiliser la [start-thing-registration-task](#) commande pour enregistrer des éléments en bloc. Cette commande utilise un modèle de provisionnement, un nom de compartiment S3, un nom de clé et un ARN de rôle qui permet d'accéder 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. Par exemple :

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}  
{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

Les opérations d'API suivantes liées à l'enregistrement en masse peuvent être utiles :

- [ListThingRegistrationTasks](#): Répertorie les tâches de provisionnement d'objets en bloc en cours.
- [DescribeThingRegistrationTask](#): fournit des informations sur une tâche d'enregistrement d'objets en bloc spécifique.
- [StopThingRegistrationTask](#): Arrête une tâche d'enregistrement d'objets en bloc.
- [ListThingRegistrationTaskReports](#): Utilisé pour vérifier les résultats et les échecs d'une tâche d'enregistrement groupé d'objets.

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 opérations d'API du plan de AWS IoT contrôle. Ces appels peuvent dépasser les [quotas de AWS IoT limitation](#) de votre compte et provoquer des erreurs de limitation. Contactez [AWS support client](#) pour augmenter vos quotas de AWS IoT 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 de provisionnement 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 just-in-time provisionnement (JITP) et l'enregistrement groupé, et le second est utilisé pour le provisionnement de la flotte.

Rubriques

- [Section Parameters \(p. 908\)](#)
- [Section Resources \(p. 909\)](#)
- [Exemple de modèle pour l'enregistrement groupé \(p. 912\)](#)
- [Exemple de modèle pour le just-in-time provisionnement \(JITP\) \(p. 913\)](#)
- [Mise en service de flotte \(p. 915\)](#)

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 corps de modèle déclare quatre paramètres :`ThingName`, `SerialNumberLocation`, et `CSR`. Tous ces paramètres sont de type `String`. Le paramètre `Location` déclare la valeur par défaut "WA".

Section Resources

La Resources section du corps du modèle indique les ressources requises pour que votre appareil puisse communiquer AWS IoT : un objet, un certificat et une ou plusieurs politiques 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 indique 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.
- `BillingGroup` - Facultatif. Chaîne pour le nom d'un groupe de facturation associé.

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, veuillez consulter [Certificats client X.509 \(p. 320\)](#).

Pour plus d'informations, veuillez consulter [Présentation des certificats X.509 \(p. 317\)](#).

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 politique

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. Ils PolicyName peuvent uniquement faire référence à des AWS IoT politiques, mais pas à des politiques IAM. 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 appartéances 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 l'enregistrement groupé

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\"] }] }"  
            }  
        }  
    }  
}
```

Exemple de modèle pour le just-in-time provisionnement (JITP)

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":{  
        "AWS::IoT::Certificate::CommonName":{  
            "Type":"String"  
        },  
        "AWS::IoT::Certificate::SerialNumber":{  
            "Type":"String"  
        }  
    }  
}
```

```
        },
        "AWS::IoT::Certificate::Country":{
            "Type":"String"
        },
        "AWS::IoT::Certificate::Id":{
            "Type":"String"
        }
    },
    "Resources":{
        "thing":{
            "Type":"AWS::IoT::Thing",
            "Properties":{
                "ThingName":{},
                "Ref":"AWS::IoT::Certificate::CommonName"
            },
            "AttributePayload":{
                "version":"v1",
                "serialNumber":{},
                "Ref":"AWS::IoT::Certificate::SerialNumber"
            }
        },
        "ThingTypeName":"lightBulb-versionA",
        "ThingGroups":[
            "v1-lightbulbs",
            {
                "Ref":"AWS::IoT::Certificate::Country"
            }
        ]
    },
    "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\"] }] }"
    }
}
}
```

Important

Vous devez l'utiliser CertificateId dans un modèle utilisé pour le provisionnement JIT.

Pour plus d'informations sur le type de modèle de provisionnement, consultez la référence [CreateProvisioningTemplate](#) de l'AWS API.

Pour plus d'informations sur l'utilisation de ce modèle pour le just-in-time provisionnement, voir : [Just-in-time Provisionnement J.](#)

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, veuillez consulter [Mise en service des modèles \(p. 908\)](#). Les modèles de mise en service de flotte peuvent contenir une section `Mapping` et une section `DeviceConfiguration`. Vous pouvez utiliser des fonctions intrinsèques au sein d'un modèle de provisionnement 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-2:123456788:provisioningtemplate/templateName`).

Mappages

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 une AWS région, vous pouvez créer un mappage qui utilise le Région AWS nom comme clé et qui 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 de configuration des appareils contient des données arbitraires que vous souhaitez envoyer à vos appareils lors du provisionnement. Par exemple :

```
{  
  "DeviceConfiguration": {  
    "Foo": "Bar"  
  }  
}
```

Si vous envoyez des messages à vos appareils à l'aide du format de charge utile JSON (JavaScript Object Notation), AWS IoT Core formate ces données au format JSON. Si vous utilisez le format de charge utile CBOR (Concise Binary Object Representation), AWS IoT Core formate ces données au format CBOR. La `DeviceConfiguration` section 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, les valeurs sont concaténées avec aucun délimiteur.

Important

`Fn::Join` n'est pas pris en charge pour [the section called “Ressources de politique” \(p. 910\)](#).

`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 l'endroit où la chaîne est divisée (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 pour le provisionnement d'une 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 \(p. 917\)](#).

Hooks de mise en service en amont

AWS recommande d'utiliser les fonctions d'accroche de pré-provisionnement lors de la création de modèles de provisionnement afin de mieux contrôler les appareils intégrés à votre compte et leur nombre. Les hooks de mise en service en amont sont des fonctions Lambda qui valident les paramètres transmis par l'appareil avant d'autoriser la mise en service du périphérique. Cette fonction Lambda doit exister dans votre compte avant de provisionner un appareil, car elle est appelée chaque fois qu'un appareil envoie une demande. [the section called “RegisterThing” \(p. 926\)](#)

Important

Veuillez à inclure les clés contextuelles `source-arn` ou `source-account` dans les conditions globales des politiques associées à votre action Lambda afin d'éviter toute manipulation des autorisations. Pour de plus amples informations à ce sujet, veuillez consulter [Prévention du député confus entre services \(p. 368\)](#).

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. Le provisionnement n'a lieu que si la fonction Lambda renvoie un objet avec. `"allowProvisioning": True`

Entrée du hook de pré-provisionnement

AWS IoT envoie cet objet à la fonction Lambda lorsqu'un appareil s'enregistre avec AWS 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",  
        ...  
    }  
}
```

L'objet `parameters` transmis à la fonction Lambda contient les propriétés de l'argument `parameters` transmis dans la charge utile de la [section appelée "RegisterThing" \(p. 926\)](#) requête.

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 des propriétés à remplacer.

Voici un exemple de réponse réussie de la fonction de pré-provisionnement.

```
{  
    "allowProvisioning": true,  
    "parameterOverrides" : {  
        "Key": "newCustomValue",  
        ...  
    }  
}
```

"parameterOverrides" des valeurs seront ajoutées au "parameters" paramètre de la charge utile de la [section appelée "RegisterThing" \(p. 926\)](#) requête.

Note

- Si la fonction Lambda échoue, la demande de provisionnement échoue ACCESS_DENIED et une erreur est enregistrée dans Logs CloudWatch.
- Si la fonction Lambda ne revient pas "allowProvisioning": "true" dans la réponse, la demande de provisionnement échoue avec ACCESS_DENIED.
- La fonction Lambda doit terminer son exécution et revenir dans les 5 secondes, sinon la demande de provisionnement échoue.

Exemple de crochet de pré-provisionnement 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 de 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 la 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 la 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;
}
```

JavaScript

Exemple de hook de mise en service en amont dans Lambda. JavaScript

```
exports.handler = function(event, context, callback) {
    console.log(JSON.stringify(event, null, 2));
    var reply = {
        allowProvisioning: true,
        parameterOverrides: {
            DeviceLocation: 'Seattle'
        }
    };
    callback(null, reply);
}
```

Création de politiques et de rôles IAM pour un utilisateur installant un appareil

Note

Ces procédures ne doivent être utilisées que si la AWS IoT console le demande.
Pour accéder à cette page depuis la console, ouvrez [Créer un nouveau modèle de provisionnement](#).

Pourquoi cela ne peut-il pas être fait dans la AWS IoT console ?

Pour une expérience optimale, les actions IAM sont effectuées dans la console IAM. Les procédures décrites dans cette section vous expliquent comment créer les rôles et les politiques IAM nécessaires à l'utilisation du modèle de provisionnement.

Création d'une politique IAM pour l'utilisateur qui installera un appareil

Cette procédure explique comment créer une politique IAM qui autorise un utilisateur à installer un appareil à l'aide d'un modèle de provisioning.

Au cours de cette procédure, vous allez basculer entre la console IAM et la AWS IoT console. Nous vous recommandons d'ouvrir les deux consoles en même temps pendant que vous effectuez cette procédure.

Pour créer une politique IAM pour l'utilisateur qui installera un appareil

1. Ouvrez le [hub de stratégies dans la console IAM](#).
2. Choisissez Create Policy (Créer une politique).
3. Sur la page Créer une politique, choisissez l'onglet JSON.
4. Accédez à la page de la AWS IoT console où vous avez choisi Configurer la politique et le rôle des utilisateurs.
5. Dans l'exemple de politique de provisionnement, choisissez Copier.
6. Revenez à la console IAM.
7. Dans l'éditeur JSON, collez la politique que vous avez copiée depuis la AWS IoT console. Cette politique est spécifique au modèle que vous créez dans la AWS IoT console.
8. Pour continuer, choisissez Suivant : Balises.
9. Sur la page Ajouter des balises (facultatif), choisissez Ajouter une étiquette pour chaque balise que vous souhaitez ajouter à cette politique. Vous pouvez ignorer cette étape si vous n'avez aucun tag à ajouter.
10. Choisissez Suivant : Vérification pour continuer.
11. Sur la page Examiner une stratégie, procédez comme suit :
 - a. Dans Nom*, entrez un nom pour la politique qui vous aidera à vous souvenir de l'objectif de la politique.

Notez le nom que vous donnez à cette politique car vous l'utiliserez dans la procédure suivante.
 - b. Vous pouvez choisir de saisir une description facultative pour la politique que vous créez.
 - c. Passez en revue le reste de cette politique et ses balises.
12. Pour terminer la création de la nouvelle politique, choisissez Create policy.

Après avoir créé votre nouvelle politique, continuez [the section called “Création d'un rôle IAM pour l'utilisateur qui installera un appareil” \(p. 921\)](#) à créer l'entrée de rôle de l'utilisateur à laquelle vous associerez cette politique.

Création d'un rôle IAM pour l'utilisateur qui installera un appareil

Ces étapes décrivent comment créer un rôle IAM qui authentifie l'utilisateur qui installera un appareil à l'aide d'un modèle de provisioning.

Pour créer une politique IAM pour l'utilisateur qui installera un appareil

1. Ouvrez le [hub de rôles dans la console IAM](#).
2. Sélectionnez Create role (Créer un rôle).
3. Dans Sélectionner une entité sécurisée, choisissez le type d'entité sécurisée à laquelle vous souhaitez donner accès au modèle que vous créez.
4. Choisissez ou entrez l'identification de l'entité de confiance à laquelle vous souhaitez accorder l'accès, puis cliquez sur Suivant.
5. Sur la page Ajouter des autorisations, dans Politiques d'autorisation, dans le champ de recherche, entrez le nom de la politique que vous avez créée lors de la [procédure précédente \(p. 920\)](#).
6. Pour la liste des stratégies, choisissez la stratégie que vous avez créée lors de la procédure précédente, puis choisissez Suivant.
7. Dans la section Nom, vérifier et créer, procédez comme suit :
 - a. Dans Nom du rôle, entrez un nom de rôle qui vous aidera à vous souvenir de l'objectif du rôle.

- b. Pour Description, vous pouvez saisir une description facultative du rôle. L'opération n'est pas nécessaire pour continuer.
- c. Vérifiez les valeurs des étapes 1 et 2.
- d. Pour Ajouter des balises (facultatif), vous pouvez choisir d'ajouter des balises à ce rôle. L'opération n'est pas nécessaire pour continuer.
- e. Vérifiez que les informations de cette page sont complètes et correctes, puis choisissez Créeer un rôle.

Après avoir créé le nouveau rôle, revenez à la AWS IoT console pour continuer à créer le modèle.

Mettre à jour une politique existante pour autoriser un nouveau modèle

Les étapes suivantes décrivent comment ajouter un nouveau modèle à une politique IAM qui autorise un utilisateur à installer un appareil à l'aide d'un modèle de provisioning.

Pour ajouter un nouveau modèle à une stratégie IAM existante

1. Ouvrez le [hub de stratégies dans la console IAM](#).
2. Dans la zone de recherche, saisissez le nom de la stratégie à mettre à jour.
3. Dans la liste située sous le champ de recherche, recherchez la politique que vous souhaitez mettre à jour et choisissez son nom.
4. Pour le résumé de la politique, choisissez l'onglet JSON, si ce panneau n'est pas déjà visible.
5. Pour modifier le document de politique, choisissez Modifier la politique.
6. Dans l'éditeur, choisissez l'onglet JSON, si ce panneau n'est pas déjà visible.
7. Dans le document de stratégie, recherchez la déclaration de politique qui contient `iot:CreateProvisioningClaimaction`.

Si le document de stratégie ne contient pas de déclaration de stratégie avec `iot:CreateProvisioningClaimaction`, copiez l'extrait de déclaration suivant et collez-le en tant qu'entrée supplémentaire dans le Statement tableau du document de stratégie.

Note

Cet extrait doit être placé avant le] caractère de fermeture du Statement tableau. Il se peut que vous deviez ajouter une virgule avant ou après cet extrait de code pour corriger d'éventuelles erreurs de syntaxe.

```
{  
    "Effect": "Allow",  
    "Action": [  
        "iot:CreateProvisioningClaim"  
    ],  
    "Resource": [  
        "--PUT YOUR NEW TEMPLATE ARN HERE--"  
    ]  
}
```

8. Accédez à la page de la AWS IoT console dans laquelle vous avez choisi Modifier les autorisations du rôle utilisateur.
9. Trouvez l'ARN de ressource du modèle et choisissez Copier.
10. Revenez à la console IAM.
11. Collez le nom de ressource Amazon (ARN) copié en haut de la liste des modèles d'ARN du Statement tableau afin qu'il s'agisse de la première entrée.

- S'il s'agit du seul ARN du tableau, supprimez la virgule à la fin de la valeur que vous venez de coller.
12. Passez en revue la déclaration de politique mise à jour et corrigez les erreurs signalées par l'éditeur.
 13. Pour enregistrer le document de politique mis à jour, choisissez Revoir la politique.
 14. Passez en revue la politique, puis choisissez Enregistrer les modifications.
 15. Revenez à la console AWS IoT.

API MQTT de mise en service des appareils

Le service Fleet Provisioning prend en charge les opérations d'API MQTT suivantes :

- [the section called "CreateCertificateFromCsr" \(p. 923\)](#)
- [the section called "CreateKeysAndCertificate" \(p. 925\)](#)
- [the section called "RegisterThing" \(p. 926\)](#)

Cette API prend en charge les mémoires tampon de réponse au format CBOR (Concise Binary Object Representation) et au format JSON (JavaScriptObject Notation), en fonction du format de *charge utile du sujet*. 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	JavaScriptNotation d'objets (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 IoT fournit des certificats clients signés par l'autorité de certification Amazon Root (CA). 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.

Pour plus d'informations sur la création d'un certificat client à l'aide de votre certificat d'autorité de certification et d'une demande de signature de certificat, reportez-vous à [Création d'un certificat client à l'aide de votre certificat d'autorité de certification \(p. 330\)](#).

Note

Pour des raisons de sécurité, le [CreateCertificateFromCsr \(p. 923\)](#) délai de certificateOwnershipToken retour expire au bout d'une heure. [RegisterThing \(p. 926\)](#) doit être appelé avant l'certificateOwnershipToken expiration. Si le certificat créé par n'[CreateCertificateFromCsr \(p. 923\)](#) a pas été

activé et n'a pas été associé à une politique ou à un autre élément au moment de l'expiration du jeton, le certificat est supprimé. Si le jeton expire, l'appareil peut appeler [CreateCertificateFromCsr \(p. 923\)](#) pour générer un nouveau certificat.

Demande d'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.

CreateCertificateFromCsrla demande

```
{  
    "certificateSigningRequest": "string"  
}
```

certificateSigningRequest

La CSR, au format PEM.

CreateCertificateFromCsrréponse

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.

CreateCertificateFromCsrla réponse

```
{  
    "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.

CreateCertificateFromCsrErreur

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.

CreateCertificateFromCsrcharge utile d'erreur

```
{  
    "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 IoT fournit des certificats clients signés par l'autorité de certification Amazon Root (CA). 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 [CreateKeysAndCertificate \(p. 925\)](#) délai de certificateOwnershipToken retour expire au bout d'une heure.

[RegisterThing \(p. 926\)](#) doit être appelé avant l'certificateOwnershipTokenexpiration.

Si le certificat créé par n'[CreateKeysAndCertificate \(p. 925\)](#) a pas été activé et n'a pas été associé à une politique ou à un autre élément au moment de l'expiration du jeton, le certificat est supprimé. Si le jeton expire, l'appareil peut appeler [CreateKeysAndCertificate \(p. 925\)](#) pour générer un nouveau certificat.

Demande d'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.

CreateKeysAndCertificateréponse

S'abonner à \$aws/certificates/create/*payload-format*/accepted

payload-format

Format de charge utile du message en tant que cbor ou json.

CreateKeysAndCertificate réponse

```
{  
    "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.

CreateKeysAndCertificate Erreur

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.

CreateKeysAndCertificate charge utile d'erreur

```
{  
    "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 d'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.

RegisterThing la demande

```
{  
    "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 \(p. 917\)](#) pour évaluer la demande d'enregistrement.

RegisterThing réponse

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.

RegisterThing la réponse

```
{  
    "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.

RegisterThing la réponse d'erreur

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.

RegisterThing la réponse d'erreur

```
{  
    "statusCode": int,  
    "errorCode": "string",  
    "errorMessage": "string"  
}
```

statusCode

Le code de statut.

errorCode

Code de l'erreur.

errorMessage

Message d'erreur.

Indexation de la flotte

Vous pouvez utiliser l'indexation du parc pour indexer, rechercher et agréger les données de vos appareils à partir des sources suivantes : [AWS IoT registre \(p. 287\)](#), [AWS IoT Device Shadow \(p. 690\)](#), [AWS IoT connectivité \(p. 1268\)](#) et [AWS IoT Device Defender \(p. 975\)](#). Vous pouvez interroger un groupe d'appareils et agréger des statistiques sur les enregistrements des appareils en fonction de différentes combinaisons d'attributs des appareils, notamment l'état, la connectivité et les violations des appareils. L'indexation du parc vous permet d'organiser, d'analyser et de dépanner votre parc d'appareils.

L'indexation de la flotte fournit les fonctionnalités ci-dessous.

- Gestion des mises à jour des index

Vous pouvez configurer un index de parc pour indexer les mises à jour relatives à vos groupes d'objets, à vos registres d'objets, à la surveillance des appareils, à la connectivité des appareils et aux violations des appareils. Lorsque vous activez l'indexation du parc, AWS IoT crée un index pour vos objets ou groupes d'objets. `AWS_Thing` est l'index créé pour tous vos objets. `AWS_ThingGroup` est l'index qui contient tous vos groupes d'objets. Une fois que l'indexation du parc est active, vous pouvez exécuter des requêtes sur votre index, par exemple pour rechercher tous les appareils portables dont l'autonomie de batterie est supérieure à 70 %. AWS IoT maintient l'index continuellement à jour avec vos données les plus récentes. Pour de plus amples informations, veuillez consulter [Gestion de l'indexation de la flotte \(p. 930\)](#).

- Recherche dans différentes sources de données

Vous pouvez créer une chaîne de requête basée sur [un langage de requête \(p. 950\)](#) et l'utiliser pour effectuer des recherches dans les sources de données que vous configurez dans les paramètres d'indexation du parc. La chaîne de requête décrit les éléments que vous souhaitez trouver. Pour plus d'informations sur les sources de données qui prennent en charge l'indexation du parc, consultez la section [Gestion de l'indexation des objets \(p. 933\)](#).

- Recherche de données agrégées

Vous pouvez rechercher sur vos appareils des données agrégées et des statistiques de retour, un centile, une cardinalité ou une liste d'éléments à l'aide de requêtes de recherche portant sur des champs particuliers. Pour plus d'informations sur les requêtes d'agrégation, consultez la section [Requête de données agrégées \(p. 945\)](#).

- Surveillance des données agrégées à l'aide des mesures de la flotte

Vous pouvez utiliser les métriques du parc pour envoyer des données agrégées afin d'analyser CloudWatch automatiquement les tendances et de créer des alarmes afin de surveiller l'état global de votre flotte. Pour plus d'informations sur les métriques du parc, consultez la section [Métriques du parc \(p. 955\)](#).

Pour utiliser l'indexation du parc, configurez votre configuration d'indexation du parc. Pour configurer la configuration de l'indexation du parc, vous pouvez utiliser la [AWS IoT console](#), les AWS kits SDK ou le AWS Command Line Interface (AWS CLI).

Pour plus d'informations sur la tarification de ce service et d'autres services, consultez la section [Tarification de la gestion des AWS IoT appareils](#).

Gestion de l'indexation de la flotte

L'indexation de flotte gère deux types d'index pour vous : l'indexation des objets et l'indexation des groupes d'objets.

Indexation des objets

L'index créé pour toutes vos informations est AWS_Things. L'indexation des objets prend en charge les sources de données suivantes : données de [AWS IoT registre \(p. 287\)](#), données [AWS IoT Device Shadow \(p. 690\)](#), données de [AWS IoT connectivité \(p. 1268\)](#) et données de [AWS IoT Device Defender \(p. 975\)](#) violations. En ajoutant ces sources de données à la configuration d'indexation de votre flotte, vous pouvez rechercher des objets, rechercher des données agrégées et créer des groupes d'objets dynamiques et des indicateurs de flotte en fonction de vos requêtes de recherche.

Registre : AWS IoT fournit un registre qui vous aide à gérer les choses. Vous pouvez ajouter les données du registre à la configuration d'indexation de votre parc pour rechercher des appareils en fonction des noms des objets, des descriptions et d'autres attributs du registre. Pour de plus amples informations sur le registre, veuillez consulter [Comment gérer les choses avec le registre \(p. 287\)](#).

Shadow - Le service [AWS IoT Device Shadow \(p. 690\)](#) fournit des ombres qui vous aident à stocker les données d'état de vos appareils. L'indexation des objets prend en charge à la fois les ombres anonymes classiques et les ombres nommées. Pour indexer les ombres nommées, activez vos paramètres d'ombres nommées et spécifiez le nom de vos ombres dans la configuration d'indexation des objets. Par défaut, vous pouvez ajouter jusqu'à 10 noms d'ombre par Compte AWS. Pour savoir comment augmenter la limite du nombre de noms fictifs, consultez la section [AWS IoT Device Management Quotas](#) dans la référence AWS générale.

Pour ajouter des ombres nommées à des fins d'indexation :

- Si vous utilisez la [AWS IoT console](#), activez l'indexation des objets, choisissez Ajouter des ombres nommées, puis ajoutez le nom de vos ombres via la sélection d'ombres nommées.
- Si vous utilisez le AWS Command Line Interface (AWS CLI), définissez-le `namedShadowIndexingMode` comme étant ON et spécifiez les noms des ombres dans [IndexingFilter](#). Pour voir des exemples de commandes CLI, voir [Gérer l'indexation des objets \(p. 933\)](#).

Important

Le 20 juillet 2022 est la version de disponibilité générale de l'intégration de l'indexation de la flotte de AWS IoT Device Management aux ombres AWS IoT Core nommées et à la AWS IoT Device Defender détection des violations. Avec cette version GA, vous pouvez indexer des ombres nommées spécifiques en spécifiant des noms d'ombres. Si vous avez ajouté vos ombres nommées à des fins d'indexation pendant la période d'aperçu public de cette fonctionnalité, du 30 novembre 2021 au 19 juillet 2022, nous vous encourageons à reconfigurer les paramètres d'indexation de votre flotte et à choisir des noms d'ombres spécifiques afin de réduire les coûts d'indexation et d'optimiser les performances.

Pour plus d'informations sur les ombres, consultez le [service AWS IoT Device Shadow \(p. 690\)](#).

Connectivité - Les données de connectivité des appareils vous aident à identifier l'état de connexion de vos appareils. Ces données de connectivité sont déterminées par les [événements du cycle de vie \(p. 1268\)](#). Lorsqu'un client se connecte ou se déconnecte, AWS IoT publie les événements du cycle de vie avec des messages sur des sujets MQTT. Un message de connexion ou de déconnexion peut être une liste d'éléments JSON fournissant des détails sur l'état de la connexion. Pour plus d'informations sur la connectivité des appareils, consultez la section [Événements du cycle de vie \(p. 1268\)](#).

Violations de Device Defender : les données relatives aux AWS IoT Device Defender violations permettent d'identifier les comportements anormaux de vos appareils par rapport aux comportements normaux que vous définissez dans un profil de sécurité. Un profil de sécurité contient un ensemble de comportements attendus de l'appareil. Chaque comportement utilise une métrique qui indique le comportement normal de vos appareils. Pour plus d'informations sur les violations de Device Defender, consultez la section [AWS IoT Device DefenderDétecter \(p. 1090\)](#).

Pour de plus amples informations, veuillez consulter [Gestion de l'indexation des objets. \(p. 933\)](#)

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.

Pour de plus amples informations, veuillez consulter [Gestion de l'indexation des groupes d'objets. \(p. 943\)](#)

Champs gérés

Les champs gérés contiennent des données associées à des objets, à des groupes d'objets, à des ombres sur les appareils, à la connectivité des appareils et aux violations de Device Defender. AWS IoT définit le type de données dans les champs gérés. Vous spécifiez les valeurs de chaque champ géré lorsque vous créez un AWS IoT objet. Par exemple, les noms d'objets, les groupes d'objets et les descriptions d'objets sont tous des champs gérés. L'indexation du parc indexe les champs gérés en fonction du mode d'indexation que vous spécifiez. Les champs gérés ne peuvent pas être modifiés ni apparaître dans `customFields`. Pour de plus amples informations, veuillez consulter [Champs personnalisés \(p. 932\)](#).

La liste suivante répertorie les champs gérés pour l'indexation des objets :

- 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 ombres classiques sans nom

```
"managedFields" : [
    {name:shadow.version, type:Number},
    {name:shadow.hasDelta, type:Boolean}
]
```

- Champs gérés pour les ombres nommées

```
"managedFields" : [
    {name:shadow.name.shadowName.version, type:Number},
    {name:shadow.name.shadowName.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 Device Defender

```

"managedFields" : [
    {name:deviceDefender.violationCount, type:Number},
    {name:deviceDefender.securityprofile.behaviorname.metricName, type:String},
    {name:deviceDefender.securityprofile.behaviorname.lastViolationTime, type:Number},
    {name:deviceDefender.securityprofile.behaviorname.lastViolationValue, type:String},
    {name:deviceDefender.securityprofile.behaviorname.inViolation, type:Boolean}
]

```

- 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},
]

```

Le tableau suivant répertorie les champs gérés qui ne peuvent pas faire l'objet de recherches.

Source de données	Champ géré impossible à rechercher
Registre	registry.version
Ombres anonymes	shadow.version
Ombres nommées	shadow.name.*.version
Device Defender	deviceDefender.version
Groupes d'objets	version

Champs personnalisés

Vous pouvez agréger les attributs des objets, les données de Device Shadow et les données relatives aux violations de Device Defender en créant des champs personnalisés pour les indexer. L'customFieldsattribut est une liste de paires de noms de champs et de types de données. Vous pouvez effectuer des requêtes d'agrégation en fonction du type de données. Le mode d'indexation que vous choisissez affecte les champs peut être spécifié danscustomFields. Par exemple, si vous spécifiez le mode REGISTRY d'indexation, vous ne pouvez pas spécifier de champ personnalisé à partir d'une ombre d'objet. Vous pouvez utiliser la commande [update-indexing-configurationCLI](#) pour créer ou mettre à jour les champs personnalisés (voir un exemple de commande dans [Mise à jour des exemples de configuration d'indexation \(p. 935\)](#)).

- Noms de champs personnalisés

Les noms de champs personnalisés pour les attributs d'objet et de groupe d'objets commencent par`attributes.`, suivi du nom de l'attribut. Si l'indexation secondaire anonyme est activée, les noms

de champs personnalisés peuvent commencer par `shadow.desired`, suivis du nom de la valeur de données masquée anonyme. Si l'indexation masquée est activée, les champs personnalisés peuvent avoir des noms de champs commençant par `shadow.name.*.desired`, `oushadow.name.*.reported`, suivis de la valeur des données masquées nommées. Si l'indexation des violations de Device Defender est activée, les noms de champs personnalisés peuvent commencer par `deviceDefender.`, suivis de la valeur des données relatives aux violations de Device Defender.

Le nom de l'attribut ou de la valeur de données qui suit le préfixe peut uniquement contenir des caractères alphanumériques ou - (traits de soulignement) et - (traits de soulignement). Il ne peut pas y avoir d'espaces.

S'il existe une incohérence de type entre un champ personnalisé de votre configuration et la valeur indexée, l'indexation du parc ignore la valeur incohérente pour les requêtes d'agrégation. CloudWatchLes journaux sont utiles pour résoudre les problèmes liés aux requêtes d'agrégation. Pour plus d'informations, veuillez consulter [Dépannage des requêtes d'agrégation pour le service d'indexation de parc \(p. 1504\)](#).

- Types de champs personnalisés

Les types de champs personnalisés ont les valeurs prises en charge suivantes : `Number`, `String`, `Boolean`.

Gérer l'indexation des objets

L'index créé pour toutes vos informations est `AWS_Things`. Vous pouvez contrôler les éléments à indexer à partir des sources de données suivantes : données de [AWS IoT Registry \(p. 287\)](#), données de [AWS IoT Device Shadow \(p. 690\)](#), données de [AWS IoT Connectivity \(p. 1268\)](#) et données de [AWS IoT Device Defender \(p. 975\)](#) violations.

Activation de l'indexation d'objet

Vous utilisez la commande [update-indexing-configuration](#) CLI ou l'opération [UpdateIndexingConfiguration](#) d'API pour créer l'`AWS_Things` index et contrôler sa configuration. Le paramètre `--thing-indexing-configuration` (`thingIndexingConfiguration`) vous permet de contrôler le type de données (par exemple, le registre, l'ombre, les données de connectivité des appareils et les données relatives aux violations de Device Defender) qui sont indexées.

Le paramètre `--thing-indexing-configuration` prend une chaîne avec la structure suivante :

```
{  
    "thingIndexingMode": "OFF"|"REGISTRY"|"REGISTRY_AND_SHADOW",  
    "thingConnectivityIndexingMode": "OFF"|"STATUS",  
    "deviceDefenderIndexingMode": "OFF"|"VIOLATIONS",  
    "namedShadowIndexingMode": "OFF"|"ON",  
    "managedFields": [  
        {  
            "name": "string",  
            "type": "Number"|"String"|"Boolean"  
        },  
        ...  
    ],  
    "customFields": [  
        {  
            "name": "string",  
            "type": "Number"|"String"|"Boolean"  
        },  
        ...  
    ],  
    "filter": {  
        "namedShadowNames": [ "string" ]  
    }  
}
```

```
}
```

Mode d'indexation des objets

L'attribut `thingIndexingMode` contrôle le type de données indexées.

Important

Pour activer l'indexation des objets, l'`thingIndexingMode` attribut ne peut pas être défini sur OFF.

Attribut	Valeurs valides	Description
<code>thingIndexingMode</code>	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.

Attribut	Valeurs valides	Description
<code>thingConnectivityIndexingMode</code>	Non spécifié.	Le truc, c'est que les données de connectivité ne sont pas indexées.
	OFF	Le truc, c'est que les données de connectivité ne sont pas indexées.
	STATUS	Les données de connectivité de l'objet sont indexées.

L'`deviceDefenderIndexingMode` attribut indique si les données relatives aux violations de Device Defender sont indexées.

Attribut	Valeurs valides	Description
<code>deviceDefenderIndexingMode</code>	Non spécifié.	Les données relatives aux violations de Device Defender ne sont pas indexées.
	OFF	Les données relatives aux violations de Device Defender ne sont pas indexées.
	VIOLATIONS	Les données relatives aux violations de Device Defender sont indexées.

L'`namedShadowIndexingMode` attribut indique si les données d'ombre nommées sont indexées.

Attribut	Valeurs valides	Description
<code>namedShadowIndexingMode</code>	Non spécifié.	Les données masquées nommées ne sont pas indexées.
	OFF	Les données masquées nommées ne sont pas indexées.
	ON	Les données d'ombre nommées sont indexées.

Note

Pour sélectionner les ombres nommées à ajouter à la configuration d'indexation de votre flotte, `namedShadowIndexingMode` définissez-les comme tels ON et indiquez-les dans [filter](#).

Champs gérés et champs personnalisés

Champs gérés

Les champs gérés contiennent des données associées à des objets, à des groupes d'objets, à des ombres sur les appareils, à la connectivité des appareils et aux violations de Device Defender. AWS IoT définit le type de données dans les champs gérés. Vous spécifiez les valeurs de chaque champ géré lorsque vous créez un AWS IoT objet. Par exemple, les noms d'objets, les groupes d'objets et les descriptions d'objets sont tous des champs gérés. L'indexation du parc indexe les champs gérés en fonction du mode d'indexation que vous spécifiez. Les champs gérés ne peuvent pas être modifiés ni apparaître dans `customFields`.

Champs personnalisés

Vous pouvez agréger les attributs, les données de Device Shadow et les données relatives aux violations de Device Defender en créant des champs personnalisés pour les indexer. L'`customFields` attribut est une liste de paires de noms de champs et de types de données. Vous pouvez effectuer des requêtes d'agrégation en fonction du type de données. Le mode d'indexation que vous choisissez affecte les champs peut être spécifié dans `customFields`. Par exemple, si vous spécifiez le mode `REGISTRY` d'indexation, vous ne pouvez pas spécifier de champ personnalisé à partir d'une ombre d'objet. Vous pouvez utiliser la commande [update-indexing-configuration CLI](#) pour créer ou mettre à jour les champs personnalisés (voir un exemple de commande dans [Mise à jour des exemples de configuration d'indexation \(p. 935\)](#)). Pour de plus amples informations, veuillez consulter [Champs personnalisés \(p. 932\)](#).

Mise à jour des exemples de configuration d'indexation

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 `update-indexing-configuration`.

Syntaxe courte :

```
aws iot update-indexing-configuration --thing-indexing-configuration \
'thingIndexingMode=REGISTRY_AND_SHADOW,deviceDefenderIndexingMode=VIOLATIONS,namedShadowIndexingMode=0
{name= shadow.name.thing1shadow.desired.DefaultDesired,
type=String},{name=shadow.desired.power, type=Boolean},
{name=deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number,
type=Number}]'
```

Syntaxe JSON :

```
aws iot update-indexing-configuration --cli-input-json \\`
    "thingIndexingConfiguration": { "thingIndexingMode": "REGISTRY_AND_SHADOW",
    "thingConnectivityIndexingMode": "STATUS",
    "deviceDefenderIndexingMode": "VIOLATIONS",
    "namedShadowIndexingMode": "ON",
    "filter": { "namedShadowNames": ["thing1shadow"] },
    "customFields": [ { "name": "shadow.desired.power", "type": "Boolean" },
    { "name": "attributes.version", "type": "Number" },
    { "name": "shadow.name.thing1shadow.desired.DefaultDesired", "type": "String" },
    { "name": "deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number", "type": "Number" } ] } }
```

Cette commande ne produit aucune sortie.

Pour vérifier l'état de l'index des objets, exécutez la commande `describe-index` CLI :

```
aws iot describe-index --index-name "AWS_Things"
```

La sortie de la commande `describe-index` ressemble à ce qui suit :

```
{  
    "indexName": "AWS_Things",  
    "indexStatus": "ACTIVE",  
    "schema": "MULTI_INDEXING_MODE"  
}
```

Note

La mise à jour de l'index du parc peut prendre un certain temps. Nous vous recommandons d'attendre que les `indexStatus` émissions soient ACTIVES avant de l'utiliser. Vous pouvez avoir différentes valeurs dans le champ du schéma en fonction des sources de données que vous avez configurées. Pour de plus amples informations, veuillez consulter [Description d'un index d'objet \(p. 938\)](#).

Pour obtenir les détails de configuration de votre objet indexant, exécutez la `get-indexing-configuration` commande CLI :

```
aws iot get-indexing-configuration
```

La sortie de la commande `get-indexing-configuration` ressemble à ce qui suit :

```
{  
    "thingIndexingConfiguration": {  
        "thingIndexingMode": "REGISTRY_AND_SHADOW",  
        "thingConnectivityIndexingMode": "STATUS",  
        "deviceDefenderIndexingMode": "VIOLATIONS",  
        "namedShadowIndexingMode": "ON",  
        "managedFields": [  
            {  
                "name": "connectivity.disconnectReason",  
                "type": "String"  
            },  
            {  
                "name": "registry.version",  
                "type": "Number"  
            }  
        ]  
    }  
}
```

```

        },
        {
            "name": "thingName",
            "type": "String"
        },
        {
            "name": "deviceDefender.violationCount",
            "type": "Number"
        },
        {
            "name": "shadow.hasDelta",
            "type": "Boolean"
        },
        {
            "name": "shadow.name.*.version",
            "type": "Number"
        },
        {
            "name": "shadow.version",
            "type": "Number"
        },
        {
            "name": "connectivity.version",
            "type": "Number"
        },
        {
            "name": "connectivity.timestamp",
            "type": "Number"
        },
        {
            "name": "shadow.name.*.hasDelta",
            "type": "Boolean"
        },
        {
            "name": "registry.thingTypeName",
            "type": "String"
        },
        {
            "name": "thingId",
            "type": "String"
        },
        {
            "name": "connectivity.connected",
            "type": "Boolean"
        },
        {
            "name": "registry.thingGroupNames",
            "type": "String"
        }
    ],
    "customFields": [
        {
            "name": "shadow.name.thing1shadow.desired.DefaultDesired",
            "type": "String"
        },
        {
            "name":
"deviceDefender.securityProfile1.NUMBER_VALUE_BEHAVIOR.lastViolationValue.number",
            "type": "Number"
        },
        {
            "name": "shadow.desired.power",
            "type": "Boolean"
        }
    ]
}

```

```
        "name": "attributes.version",
        "type": "Number"
    }
],
"filter": {
    "namedShadowNames": [
        "thing1shadow"
    ]
}
},
"thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "OFF"
}
}
```

Pour mettre à jour les champs personnalisés, vous pouvez exécuter la `update-indexing-configuration` commande. Un exemple se présente comme suit :

```
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 reconstruit, vous pouvez utiliser une requête d'agrégation sur les champs récemment ajoutés, rechercher les données du registre, les données fictives et les données d'état de connectivité des objets.

Lorsque vous modifiez le mode d'indexation, assurez-vous que tous vos champs personnalisés sont valides en utilisant le nouveau mode d'indexation. Par exemple, si vous commencez à utiliser le REGISTRY_AND_SHADOW mode avec un champ personnalisé appelé `shadow.desired.temperature`, vous devez supprimer le champ `shadow.desired.temperature` personnalisé avant de remplacer le mode d'REGISTRYindexation par. 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"
```

La réponse de la commande peut ressembler à ce qui suit :

```
{
    "indexName": "AWS_Things",
    "indexStatus": "BUILDING",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

La première fois que vous indexez votre flotte, AWS IoT crée votre index. Lorsque BUILDING cet état indexStatus est activé, vous ne pouvez pas interroger l'index. Le schéma 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 des objets pendant sa reconstruction. 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.

Vous pouvez voir différentes valeurs dans le champ du schéma en fonction des sources de données que vous avez configurées. Le tableau suivant affiche les différentes valeurs de schéma et les descriptions correspondantes :

Schema	Description
OFF	Aucune source de données n'est configurée ni indexée.
REGISTRY	Les données du registre sont indexées.
REGISTRY_AND_SHADOW	Les données de registre et les données fictives non nommées (classiques) sont indexées.
REGISTRE ET CONNECTIVITÉ	Les données de registre et les données de connectivité sont indexées.
ÉTAT DU REGISTRE ET DE L'OMBRE ET DE LA CONNECTIVITÉ	Les données de registre, les données secondaires non nommées (classiques) et les données de connectivité sont indexées.
MODE D'INDEXATION MULTIPLE	Les données relatives aux ombres nommées ou aux violations de Device Defender sont indexées, en plus des données de registre, d'ombre anonyme (classique) ou de connectivité.

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,
        "disconnectReason": "CONNECTION_LOST"
    }
},
{
    "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":"AQFCuvk7zZ3D9p0YMbFCeHbdZ+h=G"
}

```

Dans la réponse JSON, "connectivity" (tel qu'activé par le `thingConnectivityIndexingMode=STATUS` paramètre) fournit une valeur booléenne, un

horodatage et un DisconnectReason qui indique si le périphérique est connecté à AWS IoT Core Le périphérique "mything1" s'est déconnecté (`false`) à l'heure POSIX 1556649874716 en raison de`CONNECTION_LOST`. Pour plus d'informations sur les raisons de déconnexion, consultez la section [Événements du cycle de vie \(p. 1268\)](#).

```
"connectivity": {  
    "connected":false,  
    "timestamp":1556649874716,  
    "disconnectReason": "CONNECTION_LOST"  
}
```

L'appareil "mything2" connecté (`true`) à l'heure POSIX 1556649855046 :

```
"connectivity": {  
    "connected":true,  
    "timestamp":1556649855046  
}
```

Les horodatages sont exprimés en millisecondes depuis l'époque, ce qui 1556649855046 représente 18 h 44 min 15 s 46 le mardi 30 avril 2019 (UTC).

Important

Si un appareil est déconnecté depuis environ une heure, la "timestamp" valeur et la "disconnectReason" valeur de l'état de connectivité peuvent être manquantes.

Restrictions et limitations

Les restrictions et limitations pour AWS_Things sont les suivantes.

Champs d'ombre avec des types complexes

Un champ d'ombre est indexé uniquement si la valeur du champ est un type simple, tel qu'un objet JSON qui ne contient pas de tableau ou un tableau entièrement composé de types simples. (Un type simple représente une chaîne, un nombre ou un des littéraux `true` ou `false`.) Par exemple, étant donné l'état d'ombre suivant, la valeur du champ "palette" n'est pas indexée car il s'agit d'un tableau contenant des éléments de 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 des champs d'ombre 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 enregistré sous la forme `desired.one.two.three`. Si vous avez également un document miroir, il est enregistré comme suit :

```
{  
  "state": {  
    "desired": {  
      "one.two.three": "v2"  
    }  
  }  
}
```

Les deux correspondent à une requête pour `shadow.desired.one.two.three:v2`. Il est recommandé de ne pas utiliser de points dans les noms des champs cachés.

Méta-données masquées

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 "palette" champ de la section des métadonnées de l'ombre n'est pas non plus indexé.)

Appareils non enregistrés

L'indexation du parc indexe l'état de connectivité d'un appareil dont la connexion `clientId` est la même que celle `thingName` d'un objet enregistré dans le registre.

Ombres non enregistrées

Si vous avez l'[UpdateThingShadow](#) habitude de créer une ombre en utilisant un nom d'objet qui n'a pas été enregistré dans votre AWS IoT compte, les champs de cette ombre ne sont pas indexés. Cela s'applique à la fois à l'ombre anonyme classique et à l'ombre nommée.

Valeurs 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 reconnue comme numérique, la valeur des données doit être un numéro JSON de type littéral valide. La valeur peut être un entier compris entre `-2^53... 2^53-1`, une virgule flottante à double précision avec notation exponentielle facultative, ou une partie d'un tableau contenant uniquement ces valeurs.

Valeurs nulles

Les valeurs nulles ne sont pas indexées.

Valeurs maximales

Le nombre maximum de champs personnalisés pour les requêtes d'agrégation est de 5.

Le nombre maximal de percentiles requis pour les requêtes d'agrégation est de 100.

Autorisation

Vous pouvez spécifier l'index des objets sous la forme d'un Amazon Resource Name (ARN) dans une action de AWS IoT politique, comme suit :

Action	Ressource
<code>iot:SearchIndex</code>	Un ARN d'index (par exemple, <code>arn:aws:iot:<i>your-aws-region</i>:<i>your-aws-account</i>:index/AWS_Things</code>).
<code>iot:DescribeIndex</code>	Un ARN d'index (par exemple, <code>arn:aws:iot:<i>your-aws-region</i>:index/AWS_Things</code>).

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.

Gérer l'indexation des 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 `thing-group-indexing-configuration` paramètre de l'[UpdateIndexingConfiguration](#) API pour créer l'`AWS_ThingGroups` index et contrôler sa configuration. Vous pouvez utiliser l'[GetIndexingConfiguration](#) API pour récupérer la configuration d'indexation actuelle.

Pour mettre à jour les configurations d'indexation des groupes d'objets, exécutez la `update-indexing-configuration` commande CLI :

```
aws iot update-indexing-configuration --thing-group-indexing-configuration  
    thingGroupIndexingMode=ON
```

Vous pouvez également mettre à jour les configurations pour l'indexation des objets et des groupes d'objets à l'aide d'une seule commande, comme suit :

```
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.

Pour récupérer les configurations actuelles d'indexation des objets et des groupes d'objets, exécutez la get-indexing-configuration commande CLI :

```
aws iot get-indexing-configuration
```

La réponse de la commande ressemble à ce qui suit :

```
{  
    "thingGroupIndexingConfiguration": {  
        "thingGroupIndexingMode": "ON"  
    }  
}
```

Description des index de groupes

Pour récupérer l'état actuel de l'AWS_ThingGroupsindex, utilisez la commande describe-index CLI :

```
aws iot describe-index --index-name "AWS_ThingGroups"
```

La réponse de la commande ressemble à ce qui suit :

```
{  
    "indexStatus": "ACTIVE",  
    "indexName": "AWS_ThingGroups",  
    "schema": "THING_GROUPS"  
}
```

AWS IoT crée votre index la première fois que vous l'indexez. Vous ne pouvez pas interroger l'index si le indexStatus est BUILDING.

Interrogation d'un index de groupes d'objets

Pour rechercher des données dans l'index, utilisez la commande search-index CLI :

```
aws iot search-index --index-name "AWS_ThingGroups" --query-string  
"thingGroupName:mythinggroup*"
```

Autorisation

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: <i>your-aws-region</i> :index/AWS_ThingGroups).

Action	Ressource
iot:DescribeIndex	Un ARN d'index (par exemple, <code>arn:aws:iot:<i>your-aws-region</i>:index/<i>AWS_ThingGroups</i></code>).

Interrogation des données agrégées

AWS IoT fournit quatre API (GetStatistics, GetCardinalityGetPercentiles, etGetBucketsAggregation) qui vous permettent de rechercher des données agrégées dans votre parc d'appareils.

Note

Pour les problèmes liés à des valeurs manquantes ou inattendues pour les API d'agrégation, consultez le guide de [résolution des problèmes liés à l'indexation de Fleet \(p. 1504\)](#).

GetStatistics

L'[GetStatistics](#) API et la commande get-statistics CLI renvoient le nombre, la moyenne, la somme, le minimum, le maximum, la somme des carrés, la variance et l'écart type pour le 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 éléments indexés dans votreCompte AWS.

aggregationField

(Facultatif) Le 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. Par exemple :

```
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 booléen :

- `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'éléments qui correspondent aux critères de la chaîne de requête et qui contiennent une valeur de champ d'agrégation valide.

GetStatistics avec des valeurs numériques

Lorsque vousappelez `GetStatistics` et spécifiez un champ d'agrégation de type `Number`, `GetStatistics` renvoie les valeurs suivantes :

count

Nombre d'éléments qui correspondent aux critères de la chaîne de requête et qui contiennent une valeur de champ d'agrégation valide.

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": 33.33333333333336,  
        "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'[GetCardinality](#) API et la commande get-cardinality CLI renvoient le nombre approximatif de valeurs uniques correspondant 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. Par exemple :

```
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 éléments indexés dans votreCompte AWS.

aggregationField

Champ à agréger.

query-version

Version de la requête à utiliser. La valeur par défaut est 2017-09-30.

GetPercentiles

L'[GetPercentiles](#) API et la commande get-percentiles CLI regroupent les valeurs agrégées qui correspondent à la requête en groupes 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 éléments indexés dans votreCompte 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 percentiles personnalisés.

GetBucketsAggregation

L'[GetBucketsAggregation](#)API et la commande get-buckets-aggregation CLI renvoient une liste de compartiments et le nombre total d'éléments qui correspondent aux critères de la chaîne de requête.

L'exemple suivant montre comment appeler la commande de l'interface de ligne de commande get-buckets-aggregation.

```
aws iot get-buckets-aggregation --query-string '*' --index-name AWS_Things --
aggregation-field 'shadow.reported.batteryLevelPercent' --buckets-aggregation-type
'termsAggregation={maxBuckets=5}'
```

Cette commande renvoie le résultat suivant :

```
{
    "totalCount": 20,
```

```
"buckets": [
    {
        "keyValue": "100",
        "count": 12
    },
    {
        "keyValue": "90",
        "count": 5
    },
    {
        "keyValue": "75",
        "count": 3
    }
]
```

La commande get-buckets-aggregation 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 éléments indexés dans votreCompte AWS.

aggregation-field

Champ à agréger.

buckets-aggregation-type

Contrôle de base de la forme de la réponse et du type d'agrégation de compartiments à effectuer.

Autorisation

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: <i>your-aws-region</i> :index/AWS_Things ou arn:aws:iot: <i>your-aws-region</i> :index/AWS_ThingGroups).

Syntaxe de requête

Dans l'indexation du parc, vous utilisez une syntaxe de requête pour spécifier des requêtes.

Fonctionnalités prises en charge

La syntaxe de requête prend en charge les fonctionnalités suivantes :

- Termes et expressions
- Champs de recherche
- Recherche de préfixe
- Recherche de plage
- Opérateurs booléens AND, ORNOT, et. – Le trait d'union est utilisé pour exclure un élément des résultats de recherche (par exemple, `thingName:(tv* AND -plasma)`).
- Regroupement
- Regroupement de champs
- Échapper aux caractères spéciaux (par exemple avec \)

Fonctions non prises en charge

La syntaxe de la requête ne prend pas en charge les fonctionnalités 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

Remarques

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 Registre, Device Shadow et Device Defender.
- Tous les noms de champs sont sensibles à la casse.
- La recherche est insensible à la casse. Les mots sont séparés par des espaces blancs tels que définis par Java. `Character.isWhitespace(int)`
- L'indexation des données Device Shadow (ombres anonymes et ombres nommées) inclut les sections signalées, souhaitées, delta et métadonnées.
- Les versions miroir et de registre de l'appareil ne peuvent pas être recherchées, mais elles sont présentes dans la réponse.
- Le nombre maximum de termes dans une requête est de douze.

Exemples de requêtes sur des objets

Spécifiez les requêtes dans une chaîne de requête à l'aide d'une syntaxe de requête. Les requêtes sont transmises à l'[SearchIndex API](#). Le tableau ci-après répertorie quelques exemples de chaînes de requête.

Chaîne de requête	Résultat
abc	Requêtes pour « abc » dans n'importe quel registre, ombre (ombre anonyme classique et ombre nommée) ou dans n'importe quel champ de violations de Device Defender.
thingName : myThingName	Requêtes pour un objet dont le nom est « myThingName ».
thingName:my*	Requêtes concernant les objets dont le nom commence par « my ».
thingName:ab?	Requêtes portant sur des éléments dont le nom comporte « ab » plus un caractère supplémentaire (par exemple, « aba », « abb », « abc », etc.)
thingTypeName:aa	Requêtes portant sur des éléments 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 portant sur des éléments dont l'attribut nommé « MyAttribute » possède une valeur comprise dans une plage numérique (75 à 80, inclus).
attributes.myAttribute:{75 TO 80}	Requêtes portant sur des éléments dont l'attribut nommé « MyAttribute » possède une valeur comprise dans la plage numérique (>75 et <=80).
attributes.serialNumber:["abcd" TO "abcf"]	Requêtes portant sur des éléments dont l'attribut nommé « SerialNumber » possède une valeur 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 portant sur des éléments comportant un attribut nommé « MyAttribute » dont la valeur est « i », suivie d'un nombre quelconque de caractères, suivi de « 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 des éléments dont l'attribut est nommé « attr1 » avec la valeur « abc », un attribut nommé « attr2 » inférieur à 5 et un attribut nommé « attr3 » dont la valeur n'est pas supérieure à 10.
shadow.hasDelta:true	Requêtes portant sur des objets comportant une ombre anonyme dotée d'un élément delta.
NOT attributes.model:legacy	Requêtes concernant les objets dont l'attribut nommé « model » n'est pas défini sur « legacy ».

Chaîne de requête	Résultat
shadow.reported.stats.battery:{70 TO 100} (v2 OR v3) NOT attributes.model:legacy	Requêtes concernant les objets possédant les caractéristiques suivantes : <ul style="list-style-type: none"> 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 : <ul style="list-style-type: none"> L'attribut location figure dans la section reported du shadow. L'attribut stats.battery n'existe pas dans la desired section de l'ombre.
nom de l'ombre. . hasDelta : <shadowName>vrai	Requêtes portant sur des éléments comportant une ombre portant le même nom et un élément delta.
nom de l'ombre. <shadowName>.filament souhaité : *	Requêtes portant sur des éléments comportant une ombre portant le nom donné et également une propriété de filament souhaitée.
nom de l'ombre. <shadowName>.lieu signalé : *	Requêtes portant sur des objets dont une ombre porte le nom donné et où l'locationattribut existe dans la section signalée de l'ombre nommée.
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 TO 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 TO 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.

Chaîne de requête	Résultat
Connectivité. Raison de la déconnexion : *	Requêtes pour tous les appareils dotés de la connectivité DisconnectReason.
Motif de connectivité et de déconnexion : Client_Initiated_Disconnect	Requêtes pour tous les appareils déconnectés en raison de CLIENT_INITIATED_DISCONNECT.
DeviceDefender.ViolationCount : [0 À 100]	Les requêtes portant sur des objets dont la valeur de décompte des violations de Device Defender se situe dans la plage numérique (0 à 100, inclus).
DeviceDefender. < appareil ->SecurityProfile. Comportement de déconnexion. InViolation:True	Requêtes portant sur des éléments qui ne respectent pas le comportement disconnectBehavior défini dans le profil de sécurité device-SecurityProfile. Notez que InViolation:False n'est pas une requête valide.
DeviceDefender. <device SecurityProfile ->. Comportement de déconnexion. lastViolationValue.nombre>2	Requêtes concernant les éléments qui enfreignent le comportement disconnectBehavior tel que défini dans le profil de sécurité de l'appareil, SecurityProfile avec une valeur d'événement de dernière violation supérieure à 2.
DeviceDefender. <device SecurityProfile ->. Comportement de déconnexion. lastViolationTime>1634227200000	Requêtes concernant les éléments qui enfreignent le comportement disconnectBehavior tel que défini dans le dispositif de profil de sécurité, SecurityProfile avec un dernier événement de violation après une époque spécifiée.

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: myGroupThing Nom	Requêtes pour un groupe d'objets dont le nom est « myGroupThing Nom ».
thingGroupNames:mon*	Requêtes concernant les groupes d'objets dont le nom commence par « my ».
thingGroupNames:ab ?	Requêtes portant sur des groupes d'objets dont le nom comporte « ab » plus 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 portant sur des groupes d'objets comportant un attribut nommé « MyAttribute » dont

Chaîne de requête	Résultat
	la valeur se situe dans une plage numérique (75 à 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 des groupes d'objets qui possèdent un attribut nommé « attr1 » avec une valeur « abc », un attribut nommé « attr2 » inférieur à 5 et un attribut nommé « attr3 » dont la valeur n'est pas supérieure à 10.
NOT attributes.myAttribute:cde	Requêtes concernant les groupes d'objets dont l'attribut nommé « myAttribute » n'est pas « cde ».
parentGroupNames:(myParentThingGroupName)	Requêtes relatives à des groupes d'objets dont le nom du groupe parent correspond à « myParentThing GroupName ».
parentGroupNames:(myParentThing GroupName OU myRootThingGroupName)	Requêtes relatives à des groupes d'objets dont le nom du groupe parent correspond à myParentThing GroupName « » ou « myRootThing GroupName ».
parentGroupNames:(myParentThingGroupNa*)	Requêtes relatives à des groupes d'objets dont le nom du groupe parent commence par « myParentThing GroupNa ».

Métriques du parc

Les métriques de flotte sont une fonctionnalité de l'[indexation du parc \(p. 929\)](#), un service géré qui vous permet d'indexer, de rechercher et d'agrégier les données de vos appareils. AWS IoT Vous pouvez utiliser les statistiques de votre parc pour surveiller l'état global des appareils de votre parc [CloudWatch](#) au fil du temps, notamment en examinant le taux de déconnexion de vos appareils ou l'évolution moyenne du niveau de batterie sur une période donnée.

À l'aide des métriques de flotte, vous pouvez créer des [requêtes d'agrégation \(p. 945\)](#) dont les résultats sont transmis en continu [CloudWatch](#) sous forme de métriques permettant d'analyser les tendances et de créer des alarmes. Pour vos tâches de surveillance, vous pouvez spécifier les requêtes d'agrégation de

différents types d'agrégation (statistiques, cardinalité et percentile). Vous pouvez enregistrer toutes vos requêtes d'agrégation pour créer des indicateurs de flotte à réutiliser future.

Didacticiel de démarrage

Dans ce didacticiel, vous allez créer une [métrique de flotte \(p. 955\)](#) pour surveiller la température de vos capteurs afin de détecter d'éventuelles anomalies. Lors de la création de la métrique du parc, vous définissez une [requête d'agrégation \(p. 945\)](#) qui détecte le nombre de capteurs dont la température dépasse 80 degrés Fahrenheit. Vous spécifiez la requête à exécuter toutes les 60 secondes et les résultats de la requête sont envoyés vers CloudWatch, où vous pouvez voir le nombre de capteurs présentant des risques potentiels de température élevée et définir des alarmes. Pour terminer ce tutoriel, vous utiliserez [AWS CLI](#).

Dans le cadre de ce tutoriel, vous allez apprendre à :

- [Mettre en place \(p. 956\)](#)
- [Créez des indicateurs de flotte \(p. 958\)](#)
- [Afficher les métriques dans CloudWatch \(p. 959\)](#)
- [Nettoyer les ressources \(p. 960\)](#)

Ce tutoriel prend environ 15 minutes.

Prérequis

- Installez la dernière version de [AWS CLI](#)
- Familiarisez-vous avec les [requêtes pour les données agrégées](#)
- Familiarisez-vous avec [l'utilisation CloudWatch des métriques Amazon](#)

Configuration

Pour utiliser les métriques du parc, activez l'indexation du parc. Pour activer l'indexation du parc pour vos objets ou groupes d'objets avec des sources de données spécifiques et des configurations associées, suivez les instructions des rubriques Gestion de l'indexation des objets [et Gestion de l'indexation \(p. 933\)](#) [des groupes d'objets \(p. 943\)](#).

Pour configurer

1. Exécutez la commande suivante pour activer l'indexation du parc et spécifier les sources de données à partir desquelles effectuer la recherche.

```
aws iot update-indexing-configuration \
--thing-indexing-configuration
"thingIndexingMode=REGISTRY_AND_SHADOW,customFields=[{name=attributes.temperature,type=Number},
{name=attributes.rackId,type=String},
{name=attributes.stateNormal,type=Boolean}],thingConnectivityIndexingMode=STATUS" \
```

L'exemple de commande CLI précédent permet d'indexer le parc afin de permettre la recherche de données de registre, de données fictives et d'état de connectivité des objets à l'aide de l'AWS_Thingsindex.

La modification de configuration peut prendre quelques minutes. Vérifiez que l'indexation de votre flotte est activée avant de créer des métriques de flotte.

Pour vérifier si l'indexation de la flotte a été activée, exécutez la commande d'interface de ligne de commande suivante :

```
aws --region us-east-1 iot describe-index --index-name "AWS_Things"
```

Pour plus d'informations, consultez [Activation de l'indexation d'objets. \(p. 933\)](#)

2. Exécutez le script bash suivant pour créer dix choses et les décrire.

```
# Bash script. Type `bash` before running in other shells.

Temperatures=(70 71 72 73 74 75 47 97 98 99)
Racks=(Rack1 Rack1 Rack2 Rack2 Rack3 Rack4 Rack5 Rack6 Rack6 Rack6)
IsNormal=(true true true true true false false false)

for ((i=0; i < 10; i++))
do
    thing=$(aws iot create-thing --thing-name "TempSensor$i" --attribute-payload
    attributes="{temperature=${Temperatures[@]:$i:1},rackId=${Racks[@]:$i:1},stateNormal=
    ${IsNormal[@]:$i:1}}")
    aws iot describe-thing --thing-name "TempSensor$i"
done
```

Ce script crée dix éléments représentant dix capteurs. Chaque objet possède les attributs de `temperaturerackId`, et `stateNormal` comme décrit dans le tableau suivant :

Attribut	Type de données	Description
<code>temperature</code>	Nombre	Valeur de température en degrés Fahrenheit
<code>rackId</code>	Chaîne	ID du rack de serveurs qui contient les capteurs
<code>stateNormal</code>	Booléen	Si la valeur de température du capteur est normale ou non

La sortie de ce script contient dix fichiers JSON. Un des fichiers JSON ressemble à ce qui suit :

```
{
    "version": 1,
    "thingName": "TempSensor0",
    "defaultClientId": "TempSensor0",
    "attributes": [
        {
            "rackId": "Rack1",
            "stateNormal": "true",
            "temperature": "70"
        }
    ],
    "thingArn": "arn:aws:iot:region:account:thing/TempSensor0",
    "thingId": "example-thing-id"
}
```

Pour plus d'informations, voir [Création d'un objet](#).

Créez des métriques de flotte

Pour créer une métrique de flotte

1. Pour créer une métrique de flotte nommée *High_Temp_FM*, exécutez la commande suivante. Vous créez la métrique du parc pour surveiller le nombre de capteurs dont la température dépasse 80 degrés Fahrenheit. CloudWatch

```
aws iot create-fleet-metric --metric-name "high_temp_FM" --query-string  
"thingName:TempSensor* AND attributes.temperature >80" --period 60 --aggregation-field  
"attributes.temperature" --aggregation-type name=Statistics,values=count
```

--nom de la métrique

Type de données : chaîne. Le **--metric-name** paramètre spécifie le nom d'une métrique de flotte. Dans cet exemple, vous créez une métrique de flotte nommée *High_Temp_FM*.

--chaîne de requête

Type de données : chaîne. Le **--query-string** paramètre spécifie la chaîne de requête. Dans cet exemple, la chaîne de requête signifie rechercher tous les éléments dont le nom commence par 80 degrés Fahrenheit TempSensoret dont la température est supérieure à 80 degrés Fahrenheit. Pour plus d'informations, consultez [Syntaxe de requête \(p. 950\)](#).

--période

Type de données : entier Le **--period** paramètre spécifie le temps nécessaire pour récupérer les données agrégées en secondes. Dans cet exemple, vous spécifiez que la métrique du parc que vous créez extrait les données agrégées toutes les 60 secondes.

--champ d'agrégation

Type de données : chaîne. Le **--aggregation-field** paramètre spécifie l'attribut à évaluer. Dans cet exemple, l'attribut de température doit être évalué.

--type d'agrégation

Le **--aggregation-type** paramètre spécifie le résumé statistique à afficher dans la métrique du parc. Pour vos tâches de surveillance, vous pouvez personnaliser les propriétés des requêtes d'agrégation pour les différents types d'agrégation (statistiques, cardinalité et percentile). Dans cet exemple, vous spécifiez le nombre pour le type d'agrégation et Statistics pour renvoyer le nombre d'appareils dont les attributs correspondent à la requête, en d'autres termes, pour renvoyer le nombre d'appareils dont le nom commence par 80 degrés Fahrenheit TempSensoret dont la température est supérieure à 80 degrés Fahrenheit. Pour plus d'informations, consultez [Requête de données agrégées \(p. 945\)](#).

La sortie de cette commande ressemble à ce qui suit :

```
{  
    "metricArn": "arn:aws:iot:region:111122223333:fleetmetric/high_temp_FM",  
    "metricName": "high_temp_FM"  
}
```

Note

L'affichage des points de données peut prendre un certain tempsCloudWatch.

Pour en savoir plus sur la création d'une métrique de flotte, consultez la section [Gestion des métriques de flotte \(p. 961\)](#).

Si vous ne pouvez pas créer de métrique de flotte, consultez [Résolution des problèmes liés aux métriques de flotte \(p. 1505\)](#).

2. (Facultatif) Exécutez la commande suivante pour décrire la métrique de flotte de votre parc nommée High_Temp_FM :

```
aws iot describe-fleet-metric --metric-name "high_temp_FM"
```

La sortie de cette commande ressemble à ce qui suit :

```
{  
    "queryVersion": "2017-09-30",  
    "lastModifiedDate": 1625249775.834,  
    "queryString": "*",  
    "period": 60,  
    "metricArn": "arn:aws:iot:region:111122223333:fleetmetric/high_temp_FM",  
    "aggregationField": "registry.version",  
    "version": 1,  
    "aggregationType": {  
        "values": [  
            "sum"  
        ],  
        "name": "Statistics"  
    },  
    "indexName": "AWS_Things",  
    "creationDate": 1625249775.834,  
    "metricName": "high_temp_FM"  
}
```

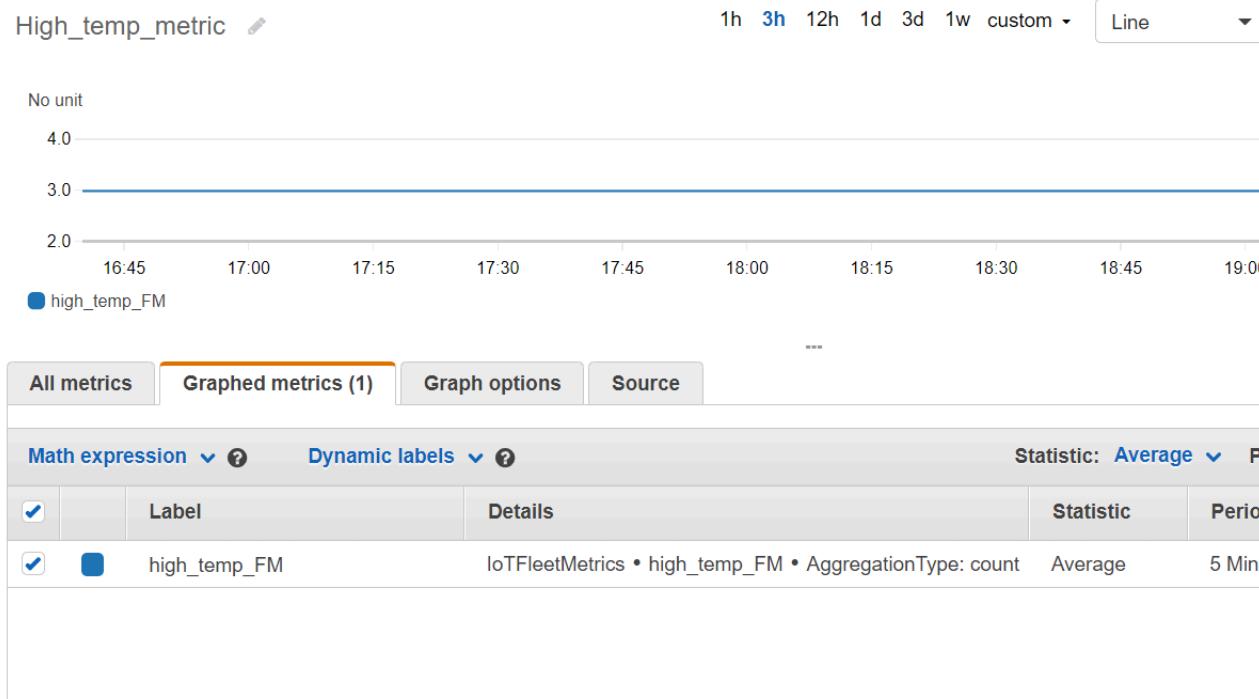
Afficher les statistiques de la flotte dans CloudWatch

Après avoir créé la métrique de flotte, vous pouvez afficher les données de métrique dans CloudWatch. Dans ce didacticiel, vous verrez la métrique qui indique le nombre de capteurs dont le nom commence par TempSensore et avec des températures supérieures à 80 degrés Fahrenheit.

Pour afficher des points de données dans CloudWatch

1. Ouvrez la console CloudWatch à l'adresse <https://console.aws.amazon.com/cloudwatch/>.
2. CloudWatch Dans le menu du panneau de gauche, choisissez Mesures pour développer le sous-menu, puis choisissez Toutes les mesures. Cela ouvre la page dont la moitié supérieure affiche le graphique et la moitié inférieure contient quatre sections à onglets.
3. La première section à onglets Toutes les mesures répertorie toutes les mesures que vous pouvez consulter par groupes, choisissez IoT FleetMetrics. Il contient tous les indicateurs de votre flotte.
4. Dans la section Type d'agrégation de l'onglet Toutes les mesures, choisissez Type d'agrégation pour afficher toutes les mesures de flotte que vous avez créées.
5. Choisissez la métrique du parc pour afficher le graphique à gauche de la section Type d'agrégation. La **somme** des valeurs s'affiche à gauche du nom de votre métrique. Il s'agit de la valeur du type d'agrégation que vous avez spécifié dans la section [Créer des métriques de flotte \(p. 958\)](#) de ce didacticiel.
6. Cliquez sur le deuxième onglet intitulé Statistiques graphiques à droite de l'onglet Toutes les mesures pour afficher les statistiques de flotte que vous avez choisies à l'étape précédente.

Vous devriez pouvoir voir un graphique qui affiche le nombre de capteurs dont la température est supérieure à 80 degrés Fahrenheit, comme suit :



Note

L'attribut Période est CloudWatch défini par défaut sur 5 minutes. Il s'agit de l'intervalle de temps entre les points de données affichés dans CloudWatch. Vous pouvez modifier le paramètre Période en fonction de vos besoins.

7. (Facultatif) Vous pouvez définir une alarme métrique.
 1. CloudWatch Dans le menu du panneau de gauche, choisissez Alarmes pour développer le sous-menu, puis choisissez Toutes les alarmes.
 2. Sur la page Alarmes, choisissez Créer une alarme dans le coin supérieur droit. Suivez les instructions de création d'alarme dans la console pour créer une alarme selon vos besoins. Pour plus d'informations, consultez [Utilisation des CloudWatch alarmes Amazon](#).

Pour en savoir plus, consultez [l'article Utiliser CloudWatch les métriques Amazon](#).

Si vous ne voyez pas les points de données CloudWatch, consultez [Résolution des problèmes liés aux métriques du parc \(p. 1505\)](#).

Nettoyage

Pour supprimer les indicateurs de flotte

Vous utilisez la commande `delete-fleet-metric` CLI pour supprimer les métriques du parc.

Pour supprimer la métrique de flotte nommée `High_Temp_FM`, exécutez la commande suivante.

```
aws iot delete-fleet-metric --metric-name "high_temp_FM"
```

Pour nettoyer les choses

Vous utilisez la commande delete-thing CLI pour supprimer des éléments.

Pour supprimer les dix éléments que vous avez créés, exécutez le script suivant :

```
# Bash script. Type `bash` before running in other shells.  
  
for ((i=0; i < 10; i++))  
do  
    thing=$(aws iot delete-thing --thing-name "TempSensor$i")  
done
```

Pour nettoyer les métriques dans CloudWatch

CloudWatch ne prend pas en charge la suppression des métriques. Les métriques expirent en fonction de leurs calendriers de conservation. Pour en savoir plus, consultez la [section Utilisation CloudWatch des métriques Amazon](#).

Gestion des métriques de flotte

Cette rubrique explique comment utiliser la AWS IoT console et comment AWS CLI gérer les indicateurs de votre flotte.

Gestion des indicateurs de flotte (console)

vous avez activé l'indexation du parc avec les sources de données et les configurations associées avant de créer des indicateurs du parc.

Activation de l'indexation de la flotte

Si vous avez déjà activé l'indexation du parc, ignorez cette section.

Si vous n'avez pas activé l'indexation du parc, suivez ces instructions.

1. Ouvrez votre AWS IoT console à l'[adresse https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/).
2. AWS IoT Dans le menu, choisissez Réglages.
3. Pour afficher les paramètres détaillés, sur la page Paramètres, faites défiler la page vers le bas jusqu'à la section Indexation de la flotte.
4. Pour mettre à jour les paramètres d'indexation de votre flotte, à droite de la section Indexation du parc, sélectionnez Gérer l'indexation.
5. Sur la page Gérer l'indexation du parc, mettez à jour les paramètres d'indexation de votre parc en fonction de vos besoins.
 - Configuration

Pour activer l'indexation des objets, activez l'indexation des objets, puis sélectionnez les sources de données à partir desquelles vous souhaitez indexer.

Pour activer l'indexation des groupes d'objets, activez-la.

- Champs personnalisés pour l'agrégation : facultatif

Les champs personnalisés sont une liste de paires de noms de champs et de types de champs.

Pour ajouter une paire de champs personnalisés, choisissez Ajouter un nouveau champ. Entrez un nom de champ personnalisé tel que `attributes.temperature`, puis sélectionnez un type de

champ dans le menu Type de champ. Notez que le nom d'un champ personnalisé commence par **attributes**. et sera enregistré en tant qu'attribut pour exécuter des [requêtes d'agrégation](#) d'objets.

Pour mettre à jour et enregistrer le paramètre, choisissez Mettre à jour.

Crée une métrique de flotte

1. Ouvrez votre AWS IoT console à l'[adresse https://console.aws.amazon.com/iot/](https://console.aws.amazon.com/iot/).
2. AWS IoT Dans le menu, choisissez Gérer, puis sélectionnez Indicateurs de flotte.
3. Sur la page Statistiques du parc, choisissez Créer un indicateur de parc et suivez les étapes de création.
4. À l'étape 1 Configurer les métriques du parc
 - Dans la section Requête, entrez une chaîne de requête pour spécifier les objets ou les groupes d'objets pour lesquels vous souhaitez effectuer la recherche agrégée. La chaîne de requête est composée d'un attribut et d'une valeur. Pour Propriétés, choisissez l'attribut de votre choix ou, s'il n'apparaît pas dans la liste, saisissez-le dans le champ. Saisissez la valeur après :. Un exemple de chaîne de requête peut être `thingName:TempSensor*`. Pour chaque chaîne de requête que vous saisissez, appuyez sur la touche Entrée de votre clavier. Si vous entrez plusieurs chaînes de requête, spécifiez leur relation en sélectionnant et, ou, et non, ou pas entre elles.
 - Dans Propriétés du rapport, choisissez le nom de l'index, le type d'agrégation et le champ d'agrégation dans leurs listes respectives. Sélectionnez ensuite les données que vous souhaitez agréger dans Sélectionner les données, où vous pouvez sélectionner plusieurs valeurs de données.
 - Choisissez Suivant.
5. À l'étape 2 : Spécifier les propriétés des métriques du parc
 - Dans le champ Nom de la métrique de flotte, entrez le nom de la métrique de flotte que vous créez.
 - Dans le champ Description - facultatif, entrez une description de la métrique du parc que vous créez. Ce champ est facultatif.
 - Dans les champs Heures et Minutes, entrez l'heure (à quelle fréquence) vous souhaitez que la métrique du parc émette des données CloudWatch.
 - Choisissez Suivant.
6. À l'étape 3 Révision et création
 - Vérifiez les paramètres des étapes 1 et 2. Pour modifier les paramètres, choisissez Edit.
 - Choisissez Créer une métrique de flotte.

Une fois la création réussie, la métrique de la flotte est répertoriée sur la page de la métrique de la flotte.

Met à activer une métrique de flotte

1. Sur la page des statistiques du parc, choisissez l'indicateur du parc que vous souhaitez mettre à jour.
2. Sur la page Détails des statistiques du parc, choisissez Modifier. Cela ouvre les étapes de création au cours desquelles vous pouvez mettre à jour les statistiques de votre flotte au cours de l'une des trois étapes.
3. Une fois que vous avez terminé de mettre à jour la métrique du parc, choisissez Mettre à jour l'indicateur du parc.

Supprimer une métrique de flotte

1. Sur la page des statistiques du parc, choisissez l'indicateur du parc que vous souhaitez supprimer.
2. Sur la page suivante qui affiche les détails des statistiques de votre flotte, choisissez Supprimer.
3. Dans la boîte de dialogue, tapez le nom de la métrique de flotte pour confirmer la suppression.
4. Choisissez Delete (Supprimer). Cette étape supprime définitivement les statistiques de votre flotte.

Gestion des métriques de flotte (CLI)

Les sections suivantes expliquent comment utiliser le AWS CLI pour gérer les indicateurs de votre flotte. Vous avez activé l'indexation du parc avec les sources de données et les configurations associées avant de créer des indicateurs de flotte. Pour activer l'indexation du parc pour vos objets ou groupes d'objets, suivez les instructions de la section [Gestion de l'indexation des objets](#) ou [Gestion de l'indexation \(p. 933\) des groupes d'objets \(p. 943\)](#).

Crée une métrique de flotte

Vous pouvez utiliser la commande `create-fleet-metric` CLI pour créer une métrique de flotte.

```
aws iot create-fleet-metric --metric-name "YourFleetMetricName" --query-string "*" --period 60 --aggregation-field "registry.version" --aggregation-type name=Statistics,values=sum
```

La sortie de cette commande contient le nom et l'Amazon Resource Name (ARN) de la métrique de flotte. Le résultat se présente comme suit :

```
{  
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",  
    "metricName": "YourFleetMetricName"  
}
```

Répertorie les métriques de flotte

Vous pouvez utiliser la commande `list-fleet-metric` CLI pour répertorier toutes les métriques de flotte de votre compte.

```
aws iot list-fleet-metrics
```

La sortie de cette commande contient toutes les métriques de flotte. Le résultat se présente comme suit :

```
{  
    "fleetMetrics": [  
        {  
            "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetric1",  
            "metricName": "YourFleetMetric1"  
        },  
        {  
            "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetric2",  
            "metricName": "YourFleetMetric2"  
        }  
    ]  
}
```

Décrire une métrique de flotte

Vous pouvez utiliser la commande `describe-fleet-metric` CLI pour afficher des informations plus détaillées sur une métrique de flotte.

```
aws iot describe-fleet-metric --metric-name "YourFleetMetricName"
```

La sortie de la commande contient les informations détaillées sur la métrique de flotte spécifiée. Le résultat se présente comme suit :

```
{  
    "queryVersion": "2017-09-30",  
    "lastModifiedDate": 1625790642.355,  
    "queryString": "*",  
    "period": 60,  
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",  
    "aggregationField": "registry.version",  
    "version": 1,  
    "aggregationType": {  
        "values": [  
            "sum"  
        ],  
        "name": "Statistics"  
    },  
    "indexName": "AWS_Things",  
    "creationDate": 1625790642.355,  
    "metricName": "YourFleetMetricName"  
}
```

Met à activer une métrique de flotte

Vous pouvez utiliser la commande update-fleet-metric CLI pour mettre à jour une métrique de flotte.

```
aws iot update-fleet-metric --metric-name "YourFleetMetricName" --query-string  
"*" --period 120 --aggregation-field "registry.version" --aggregation-type  
name=Statistics,values=sum,count --index-name AWS_Things
```

La update-fleet-metric commande ne produit aucune sortie. Vous pouvez utiliser la commande describe-fleet-metric CLI pour voir le résultat.

```
{  
    "queryVersion": "2017-09-30",  
    "lastModifiedDate": 1625792300.881,  
    "queryString": "*",  
    "period": 120,  
    "metricArn": "arn:aws:iot:us-east-1:111122223333:fleetmetric/YourFleetMetricName",  
    "aggregationField": "registry.version",  
    "version": 2,  
    "aggregationType": {  
        "values": [  
            "sum",  
            "count"  
        ],  
        "name": "Statistics"  
    },  
    "indexName": "AWS_Things",  
    "creationDate": 1625792300.881,  
    "metricName": "YourFleetMetricName"  
}
```

Supprimer une métrique de flotte

Utilisez la commande delete-fleet-metric CLI pour supprimer une métrique de flotte.

```
aws iot delete-fleet-metric --metric-name "YourFleetMetricName"
```

Cette commande ne produit aucun résultat si la suppression est réussie ou si vous spécifiez une métrique de flotte qui n'existe pas.

Pour plus d'informations, consultez [Résolution des métriques de flotte \(p. 1505\)](#).

Livraison de fichiers basée sur MQTT

L'une des options que vous pouvez utiliser pour gérer les fichiers et les transférer vers les AWS IoT appareils de votre parc est la livraison de fichiers basée sur le protocole MQTT. Cette fonctionnalité du AWS Cloud vous permet de créer un flux contenant plusieurs fichiers, de mettre à jour les données du flux (la liste et les descriptions des fichiers), d'obtenir les données du flux, etc. AWS IoT La livraison de fichiers basée sur le protocole MQTT permet de transférer des données par petits blocs vers vos appareils IoT, en utilisant le protocole MQTT qui prend en charge les messages de demande et de réponse au format JSON ou CBOR.

Pour plus d'informations sur les méthodes de transfert de données vers et depuis des appareils IoT à l'aide AWS IoT de [Connexion d'appareils à AWS IoT \(p. 85\)](#).

Rubriques

- [Qu'est-ce qu'un flux ? \(p. 965\)](#)
- [Gérer un flux dans le AWS cloud \(p. 966\)](#)
- [Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils \(p. 967\)](#)
- [Exemple de cas d'utilisation dans FreeRTOS OTA \(p. 974\)](#)

Qu'est-ce qu'un flux ?

Dans AWS IoT, un flux est une ressource publiquement adressable qui est une abstraction pour une liste de fichiers pouvant être transférés vers un appareil IoT. Un flux type 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`.
- Un identifiant de flux qui identifie votre flux et qui est utilisé (et généralement obligatoire) dans les commandes AWS Command Line Interface (AWS CLI) ou du SDK.
- Description du flux qui fournit une description de la ressource du flux.
- Version de flux qui identifie une version particulière du flux. Comme les données de flux peuvent être modifiées immédiatement avant que les appareils ne commencent le transfert de données, la version du flux peut être utilisée par les appareils pour appliquer un contrôle de cohérence.
- Liste des fichiers pouvant être transférés vers des appareils. Pour chaque fichier de la liste, le flux enregistre un identifiant 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.
- Rôle AWS Identity and Access Management (IAM) qui accorde à la transmission de fichiers AWS IoT basée sur MQTT l'autorisation de lire les fichiers de flux stockés dans le stockage de données.

AWS IoT La distribution de fichiers basée sur MQTT fournit les fonctionnalités suivantes afin que les appareils puissent transférer des données depuis le AWS cloud :

- Transfert de données à l'aide du protocole MQTT.
- Support pour les formats JSON ou CBOR.
- Possibilité de décrire un flux (`DescribeStreamAPI`) pour obtenir une liste de fichiers de flux, une version du flux et des informations connexes.
- Possibilité d'envoyer des données sous forme de petits blocs (`GetStreamAPI`) afin que les appareils soumis à des contraintes matérielles puissent recevoir les blocs.
- Support d'une taille de bloc dynamique par demande, afin de prendre en charge les appareils dotés de capacités de mémoire différentes.

- Optimisation des demandes de streaming simultanées lorsque plusieurs appareils demandent des blocs de données à partir du même fichier de flux.
- Amazon S3 en tant que stockage de données pour les fichiers de streaming.
- Support de la publication du journal de transfert de données depuis AWS IoT la livraison de fichiers MQTT vers CloudWatch.

Pour les quotas de livraison de fichiers basés sur MQTT, consultez la section [AWS IoT CoreService Quotas](#) dans le Références générales AWS.

Gérer un flux dans leAWS cloud

AWS IoT fournit unAWS SDK etAWS CLI des commandes que vous pouvez utiliser pour gérer un flux dans leAWS Cloud. Vous pouvez utiliser ces commandes pour effectuer les tâches suivantes :

- Créez un flux. [CLI/SDK](#)
- Décrivez un flux pour obtenir ses informations. [CLI/SDK](#)
- Répertoriez les streams dans votreCompte AWS. [CLI/SDK](#)
- Mettez à jour la liste des fichiers ou la description du flux dans un flux. [CLI/SDK](#)
- Supprimer un stream. [CLI/SDK](#)

Note

Pour le moment, les flux ne sont pas visibles dans leAWS Management Console. Vous devez utiliser leAWS SDKAWS CLI or pour gérer un flux dansAWS IoT.

Avant d'utiliser la transmission de fichiersAWS IoT basée sur MQTT 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 à la passerelle pourAWS IoT appareils.

Accordez des autorisations à vos appareils

Vous pouvez suivre les étapes de la [section Créer uneAWS IoT politique](#) pour créer une politique d'appareil ou utiliser une stratégie d'appareil 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 vos appareils à AWS IoT

Les appareils utilisant AWS IoT la transmission de fichiers basée sur MQTT doivent se connecter à AWS IoT. AWS IoT La distribution de fichiers basée sur MQTT s'intègre AWS IoT au AWS cloud, de sorte que vos appareils doivent se connecter directement [au point de terminaison du plan de AWS IoT données](#).

Note

Le point final du plan de AWS IoT données est spécifique à la régionCompte AWS et. Vous devez utiliser le point de terminaisonCompte AWS correspondant au et à la région dans laquelle vos appareils sont enregistrésAWS IoT.

Pour en savoir plus, consultez [Connexion à AWS IoT Core \(p. 76\)](#).

Utilisation de AWS IoT la livraison de fichiers basée sur MQTT sur les appareils

Pour lancer le processus de transfert de données, un appareil doit recevoir un ensemble de données initial, qui comprend au minimum un identifiant de flux. Vous pouvez utiliser un[Tâches \(p. 739\)](#) pour planifier des tâches de transfert de données pour vos appareils en incluant l'ensemble de données initial dans le document de travail. Lorsqu'un appareil reçoit l'ensemble de données initial, il doit alors commencer l'interaction avec la livraison de fichiersAWS IoT basée sur MQTT. Pour échanger des données avec uneAWS IoT transmission de fichiers basée sur MQTT, un appareil doit :

- Utilisez le protocole MQTT pour vous abonner au[Rubriques de livraison de fichiers basées sur MQTT \(p. 130\)](#).
- Envoyez des demandes, puis attendez de recevoir les réponses à l'aide de 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 identifiant de fichier de flux à un appareil peut simplifier la programmation du micrologiciel ou du logiciel de l'appareil, car il n'est plus nécessaire de faire uneDescribeStream demande depuis l'appareil pour obtenir cet identifiant. L'appareil peut spécifier la version du flux dans uneGetStream demande pour appliquer un contrôle de cohérence au cas où le flux aurait été mis à jour de manière inattendue.

DescribeStream À utiliser pour obtenir des données de flux

AWS IoTLa livraison de fichiers basée sur MQTT fournit l'DescribeStreamAPI permettant d'envoyer des données de flux à un appareil. Les données de flux renvoyées par cette API incluent l'identifiant du flux, la version du flux, la description du flux et une liste de fichiers de flux, chacun ayant un identifiant de fichier et la taille du fichier en octets. Avec ces informations, un appareil peut sélectionner des fichiers arbitraires pour lancer le processus de transfert de données.

Note

Vous n'avez pas besoin d'utiliser l'DescribeStreamAPI si votre appareil reçoit tous les identifiants de fichiers de flux requis dans l'ensemble de données initial.

Pour faire uneDescribeStream demande, procédez comme suit.

1. Abonnez-vous au filtre des sujets « acceptés »\$aws/things/*ThingName*/streams/*StreamId*/description/json.
2. Abonnez-vous au filtre des sujets « rejettés »\$aws/things/*ThingName*/streams/*StreamId*/rejected/json.
3. Publiez uneDescribeStream demande en envoyant un message à\$aws/things/*ThingName*/streams/*StreamId*/describe/json.
4. Si la demande a été acceptée, votre appareil reçoit uneDescribeStream réponse dans le filtre de sujet « accepté ».
5. Si la demande a été rejetée, votre appareil reçoit la réponse d'erreur dans le filtre thématique « rejetté ».

Note

Si vous remplacezjson parcbor dans les rubriques et les filtres de rubriques affichés, votre appareil reçoit des messages au format CBOR, qui est plus compact que le format JSON.

DescribeStream demande

UneDescribeStream demande type au format JSON ressemble à l'exemple ci-dessous.

```
{  
    "c": "ec944cfb-1e3c-49ac-97de-9dc4aaad0039"  
}
```

- (Facultatif) «c» est le champ du jeton client.

Le jeton client ne peut pas dépasser 64 octets. Un jeton client de plus de 64 octets provoque une réponse d'erreur et un messageInvalidRequest d'erreur.

DescribeStream réponse

UneDescribeStream réponse au format JSON ressemble à l'exemple ci-dessous.

```
{  
    "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 du jeton client. Il est renvoyé s'il a été indiqué dans la `DescribeStream` 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 un contrôle de cohérence avec vos `GetStream` demandes.
- «r» contient la liste des fichiers du flux.
 - «f» est l'ID du fichier de flux sous forme d'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 l'`GetStream` API pour qu'un appareil puisse recevoir des fichiers de flux sous forme de petits blocs de données, afin qu'elle puisse être utilisée par les appareils soumis à des contraintes liées au traitement de blocs de grande taille. Pour recevoir un fichier de données complet, 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.

GetStream demande

Pour faire une `GetStream` demande, procédez comme suit.

1. Abonnez-vous au filtre des sujets « acceptés » `$aws/things/ThingName/streams/StreamId/data/json`.
2. Abonnez-vous au filtre des sujets « rejettés » `$aws/things/ThingName/streams/StreamId/rejected/json`.
3. Publiez une `GetStream` demande sur le sujet `$aws/things/ThingName/streams/StreamId/get/json`.
4. Si la demande a été acceptée, votre appareil recevra une ou plusieurs `GetStream` réponses dans le filtre de sujet « accepté ». Chaque message de réponse contient des informations de base et une charge de données utiles 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 qu'il utilise plusieurs `GetStream` demandes afin de recevoir toutes les données demandées.
6. Si la demande a été rejetée, votre appareil recevra la réponse d'erreur dans le filtre thématique « rejetté ».

Note

- Si vous remplacez « json » par « cbor » dans les rubriques et les filtres de rubriques affichés, votre appareil recevra des messages au format CBOR, qui est plus compact que le format JSON.
- AWS IoT La livraison de fichiers basée sur MQTT limite la taille d'un bloc à 128 Ko. Si vous faites une demande pour un bloc de plus de 128 Ko, la demande échouera.
- Vous pouvez demander plusieurs blocs dont la taille totale est supérieure à 128 Ko (par exemple, si vous demandez 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 demandes afin de recevoir toutes les données demandées. Le service enverra des blocs supplémentaires au fur et à mesure que votre appareil effectuera des demandes supplémentaires. Nous vous recommandons de poursuivre une nouvelle demande uniquement après avoir reçu et traité correctement la réponse précédente.

- 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.

UneGetStream demande type au format JSON ressemble à l'exemple ci-dessous.

```
{  
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",  
    "s": 1,  
    "f": 0,  
    "l": 4096,  
    "o": 2,  
    "n": 100,  
    "b": "..."  
}
```

- [facultatif] «c» est le champ du jeton client.

La longueur du jeton client ne peut pas dépasser 64 octets. Un jeton client de plus de 64 octets provoque une réponse d'erreur et un messageInvalidRequest d'erreur.

- [facultatif] «s» est le champ de version du flux (un entier).

La livraison de fichiers basée sur MQTT applique un contrôle de cohérence basé sur cette version demandée et la dernière version du flux dans le cloud. Si la version du flux envoyée depuis un appareil dans le cadre d'uneGetStream demande ne correspond pas à la dernière version du flux dans le cloud, le service envoie une réponse d'erreur et un messageVersionMismatch d'erreur. Généralement, un appareil reçoit la version de flux attendue (la plus récente) dans l'ensemble de données initial ou dans la réponse àDescribeStream.

- «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 du SDKAWS CLI or. Si un appareil demande un fichier de flux avec un identifiant qui n'existe pas, le service envoie une réponse d'erreur et un messageResourceNotFound d'erreur.

- «l» est la taille du bloc de données en octets (un entier compris entre 256 et 131 072).

Reportez-vous à la section[Création d'un bitmap pour une GetStream demande \(p. 971\)](#) pour obtenir des instructions sur l'utilisation des champs bitmap afin de spécifier la partie du fichier de flux qui sera renvoyée dans laGetStream réponse. Si un appareil spécifie une taille de bloc hors de portée, le service envoie une réponse d'erreur et un messageBlockSizeOutOfBounds d'erreur.

- [facultatif] «o» est le décalage du bloc dans le fichier de flux (un entier compris entre 0 et 98 304).

Reportez-vous à la section[Création d'un bitmap pour une GetStream demande \(p. 971\)](#) pour obtenir des instructions sur l'utilisation des champs bitmap afin de spécifier la partie du fichier de flux qui sera renvoyée dans laGetStream réponse. La valeur maximale de 98 304 est basée sur une limite de taille de fichier de flux de 24 Mo et de 256 octets pour la taille de bloc minimale. La valeur par défaut est 0 si elle n'est pas spécifiée.

- [facultatif] «n» est le nombre de blocs demandés (un entier compris entre 0 et 98 304).

Le champ «n» indique soit (1) le nombre de blocs demandés, soit (2) lorsque le champ bitmap («b») est utilisé, la limite du nombre de blocs qui seront renvoyés par la demande bitmap. Cette seconde utilisation est facultative. S'il n'est pas défini, la valeur par défaut est 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 facilite la gestion des nouvelles tentatives suite à une erreur. Reportez-vous à la section[Création d'un bitmap pour une GetStream demande \(p. 971\)](#) pour obtenir des instructions sur l'utilisation des champs bitmap afin

de spécifier la partie du fichier de flux qui sera renvoyée dans laGetStream 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.

Important

Vous devez spécifier «b» ou «».n Si aucune d'entre elles n'est spécifiée, laGetStream demande risque de ne pas être valide lorsque la taille du fichier est inférieure à 131072 octets (128 Ko).

GetStream réponse

UneGetStream réponse au format JSON ressemble à cet exemple pour chaque bloc de données demandé.

```
{  
  "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380",  
  "f": 0,  
  "l": 4096,  
  "i": 2,  
  "p": "..."  
}
```

- «c» est le champ du jeton client. Il est renvoyé s'il a été indiqué dans laGetStream 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 actuel.
- «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 dans le codage [Base64](#).

Création d'un bitmap pour une GetStream demande

Vous pouvez utiliser le champ bitmap (b) dans uneGetStream requête pour obtenir des blocs non consécutifs à partir d'un fichier de flux. Cela permet aux appareils dont la capacité de RAM est limitée de faire face aux problèmes de mise en réseau. Un appareil ne peut demander que 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 à 1 dans le bitmap, un bloc correspondant du fichier de flux sera renvoyé.

Voici un exemple de spécification d'un bitmap et de ses champs de support dans uneGetStream demande. Par exemple, vous souhaitez recevoir un fichier de flux par blocs de 256 octets (la taille du bloc). Imaginez que chaque bloc de 256 octets possède un numéro qui indique sa position dans le fichier, à partir de 0. Le bloc 0 correspond donc au premier bloc de 256 octets du fichier, le bloc 1 au deuxième, et ainsi de suite. Vous souhaitez demander les blocs 20, 21, 24 et 43 à partir du fichier.

Décalage de bloc

Comme le premier bloc porte le numéro 20, spécifiez le décalage (champo) sur 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 en gérer avec des ressources mémoire limitées, vous pouvez spécifier le nombre maximum de blocs qui doivent être renvoyés dans chaque message envoyé par transmission de fichiers basée sur MQTT. Notez que

cette valeur n'est pas prise en compte si le bitmap lui-même spécifie un nombre inférieur à ce nombre de blocs, ou si cela rend la taille totale des messages de réponse envoyés par transmission de fichiers MQTT supérieure à la limite de service de 128 Ko par `GetStream` demande.

Mappage en mode bloc

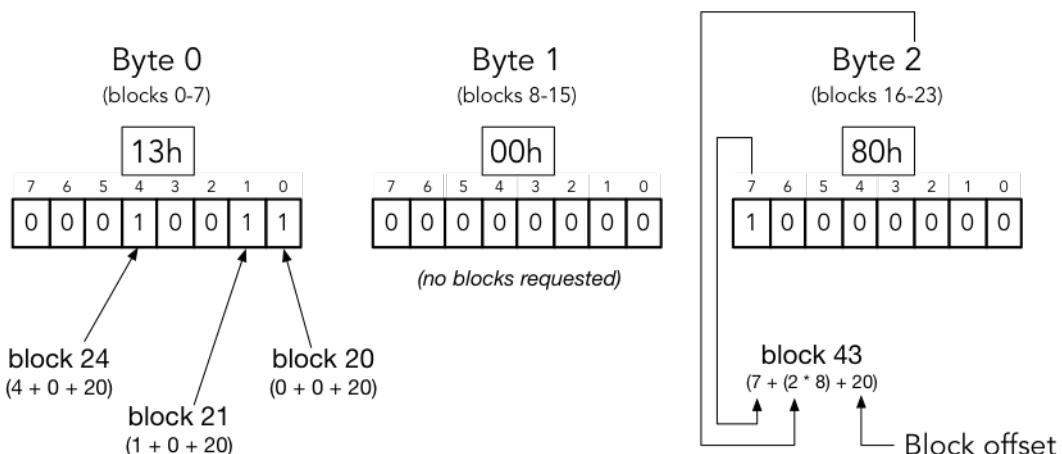
Le bitmap lui-même est un tableau d'octets non signés exprimé en notation hexadécimale et inclus dans laGetStream demande sous forme de chaîne de caractères du nombre. Mais pour construire cette chaîne, commençons par considérer le bitmap comme une longue séquence de bits (un nombre binaire). Si un bit de cette séquence est défini sur 1, le bloc correspondant du fichier de flux sera renvoyé à l'appareil. Dans 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. Ainsi, après avoir soustrait le décalage de chaque numéro de bloc, nous voulons définir les bits 0, 1, 4 et 23, comme dans l'exemple suivant.

En prenant un octet (8 bits) à la fois, cela s'écrit classiquement comme suit : « 0b00010011 », « 0b00000000 » 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 prêter à confusion à moins que vous ne connaissez les conventions. Le premier octet contient les bits 7 à 0 (dans cet ordre), le deuxième octet contient les bits 15 à 8, le troisième octet contient les bits 23 à 16, etc. En notation hexadécimale, cela se 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.

Block bitmap breakdown



block number = (bit position + (byte offset * 8) + base offset)

En mettant tout cela ensemble, le JSON de notre `getStream` demande ressemble à ce qui 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 liées à la livraison de fichiersAWS IoT basée sur MQTT

Une réponse d'erreur envoyée à un appareil pour lesGetStream APIDescribeStream et les API contient un jeton client, un code d'erreur et un message d'erreur. Une réponse d'erreur ressemble à l'exemple ci-dessous.

```
{
    "o": "BlockSizeOutOfBounds",
    "m": "The block size is out of bounds",
    "c": "1bb8aaa1-5c18-4d21-80c2-0b44fee10380"
}
```

- «**o**» est le code d'erreur qui indique la raison pour laquelle une erreur s'est produite. Reportez-vous aux codes d'erreur plus loin dans cette section pour plus de détails.
- «**m**» est le message d'erreur qui contient les détails de l'erreur.
- «**c**» est le champ du jeton client. Cela peut être retourné s'il a été indiqué dans laDescribeStream demande. Vous pouvez utiliser le jeton client pour associer la réponse à sa demande.

Le champ du jeton client n'est pas toujours inclus dans une réponse d'erreur. Lorsque le jeton client indiqué dans la demande n'est pas valide ou est mal formé, il n'est pas renvoyé dans la réponse d'erreur.

Note

Pour des raisons de rétrocompatibilité, les champs de la réponse d'erreur peuvent être sous une forme non abrégée. Par exemple, le code d'erreur peut être désigné par les champs «**code**» ou «**o**» et le champ du jeton client peut être désigné par les champs «**clientToken**» ou «**c**». Nous vous recommandons d'utiliser la forme d'abréviation ci-dessus.

InvalidTopic

La rubrique MQTT du message de flux n'est pas valide.

InvalidJson

La demande Stream n'est pas un document JSON valide.

InvalidCbor

La demande Stream 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, tel 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 MQTT](#)» dans [AWS IoT CoreService Quotas](#).

OffsetOutOfBounds

Le décalage est hors limites. Reportez-vous à la section « Livraison de fichiers basée sur MQTT » dans [AWS IoT CoreService Quotas](#).

BlockCountLimitExceeded

Le nombre de blocs de requêtes est hors limites. Reportez-vous à la section « Livraison de fichiers basée sur MQTT » dans [AWS IoT CoreService Quotas](#).

BlockBitmapLimitExceeded

La taille du bitmap de demande est hors limites. Reportez-vous à la section « Livraison de fichiers basée sur MQTT » dans [AWS IoT CoreService Quotas](#).

ResourceNotFound

Le flux, les fichiers, les versions de fichiers ou les blocs demandés n'ont pas été trouvés. Reportez-vous au message d'erreur pour plus de détails.

VersionMismatch

La version du flux dans la demande ne correspond pas à la version du flux dans la fonctionnalité de livraison de fichiers basée sur MQTT. Cela indique que les données du flux ont été modifiées depuis la réception initiale de la version du flux par l'appareil.

ETagMismatch

L'ETag S3 du flux ne correspond pas à l'ETag de la dernière version de l'objet S3.

InternalError

Une erreur interne s'est produite lors de la livraison de fichiers basée sur MQTT.

Exemple de cas d'utilisation dans FreeRTOS OTA

L'agent FreeRTOS OTA (over-the-air) utilise la transmission de fichiers AWS IoT basée sur MQTT pour transférer les images du microprogramme FreeRTOS vers des appareils FreeRTOS. Pour envoyer l'ensemble de données initial à un appareil, il utilise le service AWS IoT Job pour planifier une tâche de mise à jour OTA sur les appareils FreeRTOS.

Pour une implémentation de référence d'un client de distribution de fichiers basé sur MQTT, consultez [les codes des agents OTA FreeRTOS](#) dans 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 Defender répond à ces défis en fournissant des outils permettant d'identifier les problèmes de sécurité et les écarts par rapport aux meilleures pratiques. AWS IoT Device Defender peut auditer les parcs d'appareils pour s'assurer qu'ils respectent les meilleures pratiques de sécurité et détecter les comportements anormaux sur les appareils.

Formation et certification AWS

Suivez le cours suivant pour démarrer avec AWS IoT Device Defender : [AWS IoT Device Defender Primer](#).

Démarrer avec AWS IoT Device Defender

Vous pouvez utiliser les didacticiels suivants pour travailler avec AWS IoT Device Defender.

Rubriques

- [Configuration \(p. 975\)](#)
- [Guide d'audit \(p. 976\)](#)
- [Guide de ML \(p. 989\)](#)
- [Personnalisez quand et comment vous affichez AWS IoT Device Defender Résultats de l'audit \(p. 1012\)](#)

Configuration

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

Rubriques

- [S'inscrire à un Compte AWS \(p. 19\)](#)
- [Création d'un utilisateur administratif \(p. 20\)](#)

S'inscrire à un Compte AWS

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

Pour s'inscrire à 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 saisissez un code de vérification en utilisant le clavier numérique du téléphone.

Lorsque vous souscrivez à un Compte AWS, un Utilisateur racine d'un compte AWS est créé. Par défaut, seul l'utilisateur root a accès à l'ensemble des Services AWS et des ressources de ce compte. En tant que bonne pratique de sécurité, [attribuer un accès administratif à un utilisateur administratif](#), et utilisez uniquement l'utilisateur root pour effectuer [tâches nécessitant un accès utilisateur root](#).

AWS vous envoie un e-mail de confirmation lorsque le processus d'inscription est terminé. Vous pouvez afficher l'activité en cours de votre compte et gérer votre compte à tout moment en accédant à <https://aws.amazon.com/> et en cliquant sur Mon compte.

Création d'un utilisateur administratif

Une fois que vous êtes inscrit à un Compte AWS, créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisation de votre Utilisateur racine d'un compte AWS

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

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

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

Pour obtenir des instructions, consultez [Activation d'un dispositif MFA virtuel pour l'utilisateur root de votre Compte AWS \(console\)](#) dans le Guide de l'utilisateur IAM.

Création d'un utilisateur administratif

- Pour vos tâches administratives quotidiennes, octroyez un accès administratif à un utilisateur administratif dans AWS IAM Identity Center (successor to AWS Single Sign-On).

Pour plus d'informations, consultez [Mise en route](#) dans le Guide de l'utilisateur AWS IAM Identity Center (successor to AWS Single Sign-On).

Connexion en tant qu'utilisateur administratif

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

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

Ces tâches créent unCompte AWS et un utilisateur avec des privilèges d'administrateur pour le compte.

Guide d'audit

Ce didacticiel fournit des instructions sur la configuration d'un audit récurrent, la configuration des alarmes, l'examen des résultats d'audit et la réduction des problèmes d'audit.

Rubriques

- [Prérequis \(p. 977\)](#)
- [Activer les contrôles d'audit \(p. 977\)](#)
- [Afficher les résultats de l'audit \(p. 980\)](#)
- [Création d'actions d'atténuation d'audit \(p. 981\)](#)
- [Appliquez des mesures d'atténuation aux résultats de vos audits \(p. 984\)](#)
- [Activer les notifications SNS \(facultatif\) \(p. 985\)](#)
- [Activer la journalisation \(facultatif\) \(p. 987\)](#)
- [Création d'uneAWS IoT Device DefenderRôle Audit IAM \(facultatif\) \(p. 988\)](#)

Prérequis

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

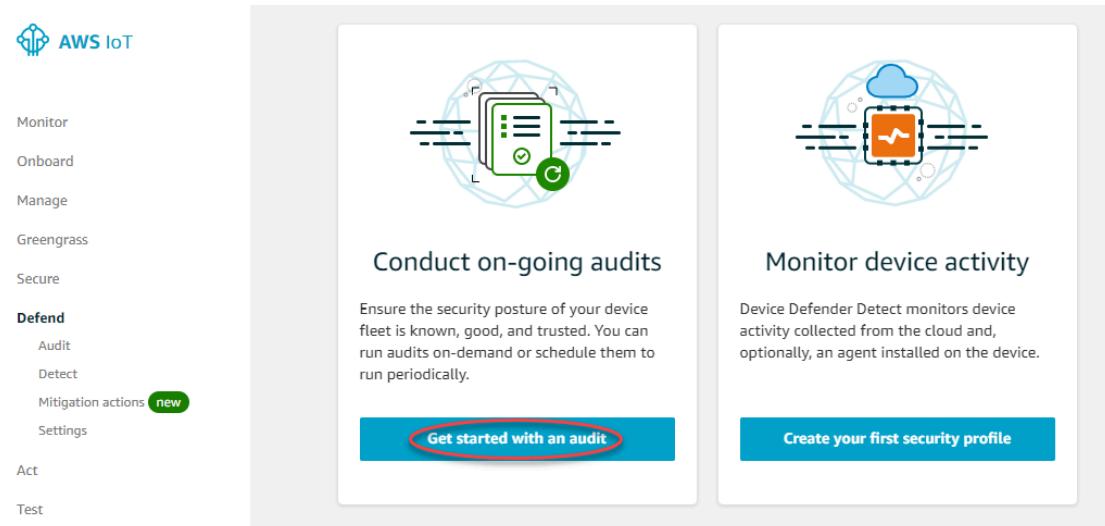
- Un Compte AWS. Si vous ne disposez pas de ce service, consultez.[Configuration d'.](#)

Activer les contrôles d'audit

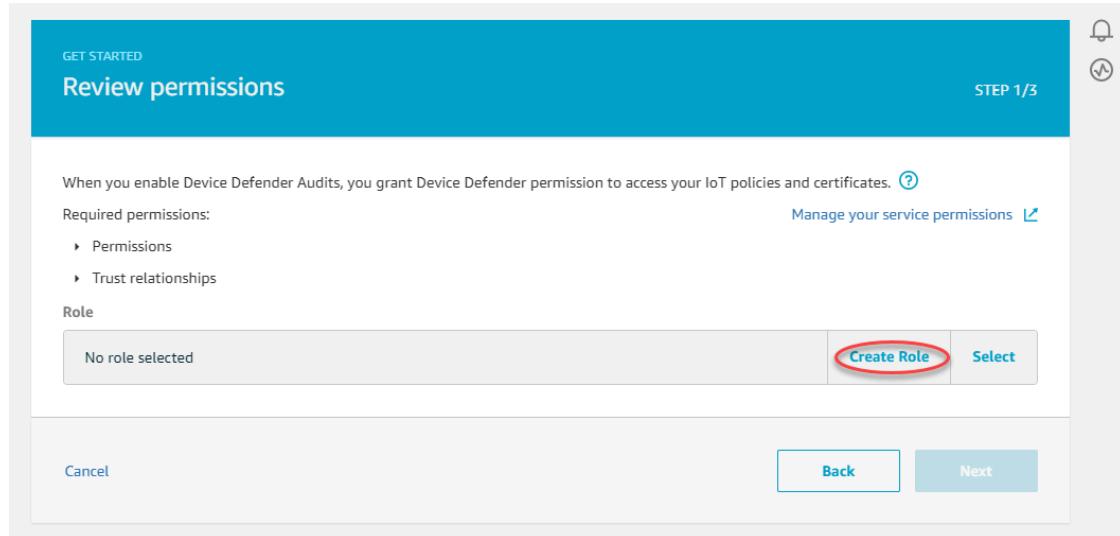
Dans la procédure suivante, vous activez les vérifications d'audit qui examinent les paramètres et les paramètres du compte et de l'appareil afin de vérifier que des mesures de sécurité sont en place. Dans ce tutoriel, nous vous demandons d'activer toutes les vérifications d'audit, mais vous pouvez sélectionner les vérifications de votre choix.

La tarification de l'audit est par nombre d'appareils et par mois (appareils de parc connectés àAWS IoT). Par conséquent, l'ajout ou la suppression de contrôles d'audit n'affecterait pas votre facture mensuelle lorsque vous utilisez cette fonctionnalité.

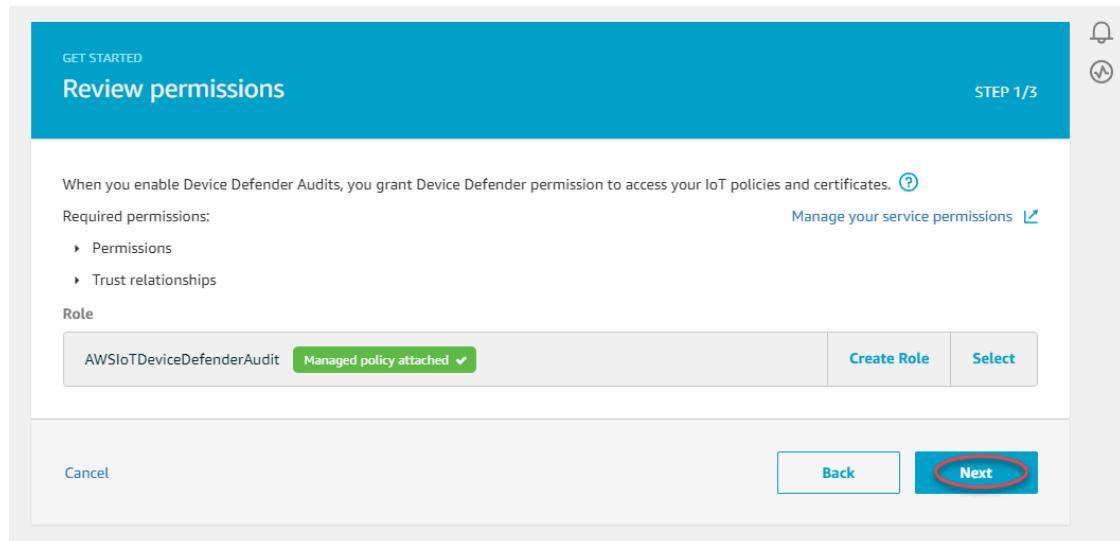
1. Dans[AWS IoTconsole](#), dans le volet de navigation, développezDéfendreet sélectionnezDémarrer avec un audit.



2. LeDémarrer avec Device Defender Auditdonne une vue d'ensemble des étapes requises pour activer les vérifications d'audit. Une fois que vous avez examiné l'écran, sélectionnez.Suivant.
3. Si vous avez déjà un rôle à utiliser, vous pouvez le sélectionner. Sinon, sélectionnezCréation d'un rôleet nommez-le[AWS IoT Device Defender Audit](#).



Les autorisations requises doivent être automatiquement attachées au rôle. Sélectionnez les triangles en regard de `AuthorisationsetRelations` d'approbation pour voir quelles autorisations sont accordées. Tâche de sélection suivante lorsque vous êtes prêt à aller de l'avant.



4. Dans la page Sélectionnez les chèques que 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 souhaitées. 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, consultez [Contrôles d'audit](#).

Tâche de sélection suivante une fois que vous avez sélectionné vos chèques.

GET STARTED

Select checks

STEP 2/3

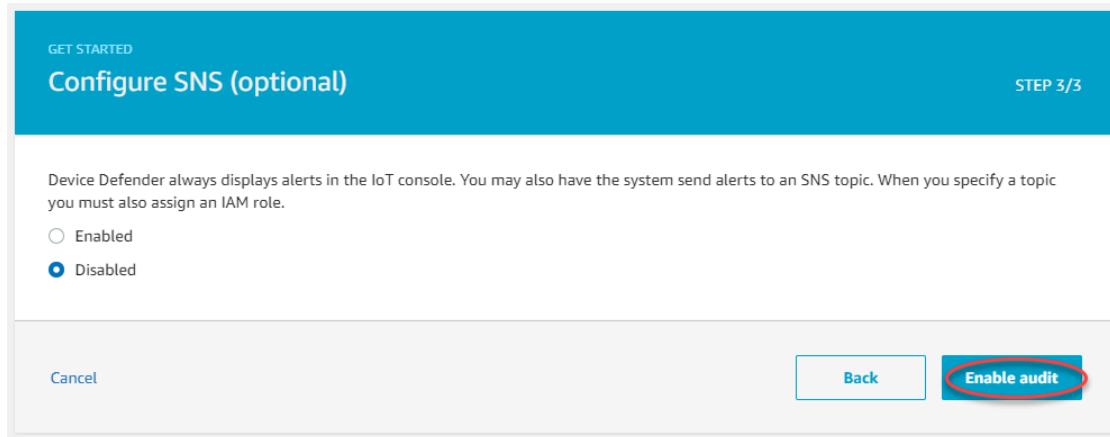
The checks you select here will be available when you set up audits. Data collection begins when a check has been enabled. All the free checks have been pre-selected for you. You can enable or disable checks at any time through the Device Defender Audit settings.

<input checked="" type="checkbox"/> Check name	Severity	Resource type
<input checked="" type="checkbox"/> Authenticated Cognito role overly permissive ?	Critical	Cognito pool
<input checked="" type="checkbox"/> CA certificate key quality ?	Critical	CA certificate
<input checked="" type="checkbox"/> CA certificate revoked but device certificates still active ?	Critical	CA certificate
<input checked="" type="checkbox"/> Device Certificate key quality ?	Critical	Device certificate
<input checked="" type="checkbox"/> Device certificate shared ?	Critical	Device certificate
<input checked="" type="checkbox"/> IoT policies overly permissive ?	Critical	Policy
<input checked="" type="checkbox"/> Role Alias overly permissive ?	Critical	Role alias
<input checked="" type="checkbox"/> Unauthenticated Cognito role overly permissive ?	Critical	Cognito pool
<input checked="" type="checkbox"/> Conflicting MQTT client IDs ?	High	Client ID
<input checked="" type="checkbox"/> CA certificate expiring ?	Medium	CA certificate
<input checked="" type="checkbox"/> Device certificate expiring ?	Medium	Device certificate
<input checked="" type="checkbox"/> Revoked device certificate still active ?	Medium	Device certificate
<input checked="" type="checkbox"/> Role Alias allows access to unused services ?	Medium	Role alias
<input checked="" type="checkbox"/> Logging disabled ?	Low	Account settings

[Cancel](#) [Back](#) [Next](#)

Vous pouvez toujours modifier vos vérifications d'audit configurées sous Paramètres.

5. Dans la page Configurer SNS (facultatif) écran, sélectionnez Activer l'audit. Si vous souhaitez activer les notifications SNS, consultez [Activer les notifications SNS \(facultatif\) \(p. 985\)](#).



6. Vous allez être redirigé vers Schedulessous Audit.

Afficher les résultats de l'audit

La procédure suivante vous montre comment afficher les résultats de vos audits. Dans ce didacticiel, vous pouvez voir les résultats de l'audit des contrôles d'audit configurés dans [Activer les contrôles d'audit \(p. 977\)](#) Didacticiel de

Pour afficher les résultats de l'audit

1. Dans [AWS IoT console](#), dans le volet de navigation, développez Défendre, sélectionnez Audit, et sélectionnez Résultats.
2. Le Récapitulatif vous indiquera si vous avez des contrôles non conformes.

The screenshot shows the AWS IoT Device Defender Audit Results page. On the left, there's a navigation sidebar with options like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend, Audit, and Results. Under Audit, there are sub-options for Schedules, Action executions, Finding suppressions, Detect, Mitigation actions (which is highlighted in blue), and Settings. The main content area is titled "Audit results (10+)" and shows a table with columns: Name, Date, Status, and Summary. The table lists 10 audit entries, all of which are marked as "Not compliant" (indicated by a red triangle icon). The "Status" column shows some entries as "Completed" (green circle) and others as "Not compliant". The "Summary" column indicates that each entry is "1 of 14 non-compliant".

3. SELECT la Nom de la vérification d'audit que vous souhaitez examiner.
`daily_audit_check - May 14, 2020 8:51:27 AM -0700`

The screenshot shows the "Audit findings" page for the audit task "daily_audit_check - May 14, 2020 8:51:27 AM -0700". At the top, it says "Audit task ID 302ce82f9e3377e1f6be784532ff04c0" and "Started at May 14, 2020 8:51:27 AM -0700". Below this, there are sections for "Non-compliant checks (2 of 14)", "Compliant checks (12 of 14)", and "Mitigation actions". The "Non-compliant checks" section lists two items: "IoT policies overly permissive" (Severity: Critical, Non-compliant: 1, % Resources: 50.0%, Mitigation: Use restrictive policies) and "Logging disabled" (Severity: Low, Non-compliant: 1, % Resources: 100.0%, Mitigation: Enable logging). The "Mitigation actions" section shows a table with columns: Created date, Task name, and Status. It notes that there are "0 of 0" mitigation actions.

4. Utilisez les points d'interrogation pour obtenir des conseils sur la façon de rendre vos contrôles non conformes. Par exemple, vous pouvez suivre [Activer la journalisation \(facultatif\) \(p. 987\)](#) pour rendre la vérification « Consignation désactivée » conforme.

Création d'actions d'atténuation d'audit

Dans la procédure suivante, vous allez créer un AWS IoT Device DefenderAction d'atténuation de l'audit pour activer AWS IoT Journalisation. Chaque vérification d'audit a cartographié les mesures d'atténuation qui affecteront le type d'action que vous choisissez. Pour plus d'informations, veuillez consulter [Actions 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*Activer l'action de journalisation des erreurs*.
5. DansType d'action, choisissezActiver la journalisation IoT.
6. DansRôle d'exécution d'action, sélectionnezCréation d'un rôle. PourNom, utilisez*Rôle de journalisation des erreurs de l'action d'atténuation de l'IoT*. Ensuite, choisissezCréation d'un rôle.
7. DansParamètres, sousRôle de journalisation, sélectionnezAWSIoTLoggingRole. PourNiveau de journalisation, choisissezError.

Create a new mitigation action

You can use AWS IoT Device Defender to take actions to mitigate issues that were found during an audit. AWS IoT Device Defender provides predefined actions for the different audit checks. You can configure those actions for your AWS account and then apply them to a set of findings. [Learn more](#)

Action name [?](#)

Action type [?](#)

Permissions

Please create or select a role with the following mitigation action type specific permission(s) and trust relationship.

Required permissions: [Manage your service permissions](#)

- ▶ Permissions
- ▶ Trust relationships

You can also attach an action specific managed policy to an existing role, or create a new role with the required managed policy attached.

Action execution role [?](#)
 Managed policy attached ✓ [Create Role](#) [Select](#)

Parameters

Role for logging [?](#)
 [Clear](#) [Select](#)

Log level [?](#)

Tags

Tag name Value [Clear](#)

[Add another](#)

[Cancel](#) [Save](#)

8. Choisissez Save (Enregistrer) pour enregistrer votre action d'atténuation pour votre compte AWS.
9. Une fois créé, l'écran suivant indique que votre action d'atténuation a été créée avec succès.

The screenshot shows a success message: "Successfully created mitigation action". Below it is a table titled "Mitigation actions (1)" with columns: Created date, Action name, and ARN. One row is listed: Jun 18, 2020 8:30:07 AM -0700, EnableErrorLoggingAction, arn:aws:iot:us-east-1:765219403047:mitigationaction/EnableErrorLoggingAction.

Appliquez des mesures d'atténuation aux résultats de vos audits

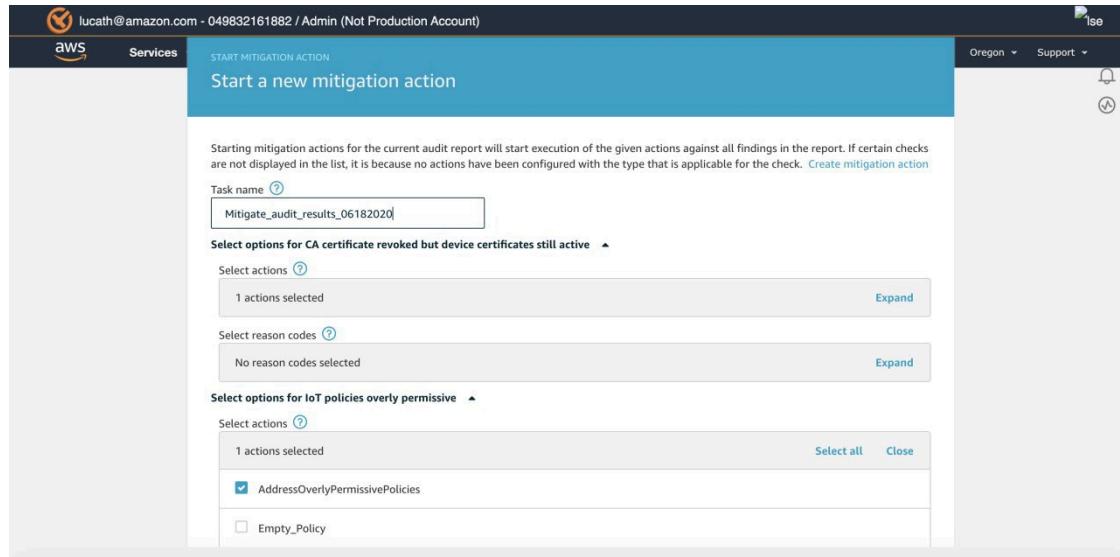
La procédure suivante vous montre comment appliquer des actions d'atténuation aux résultats de vos audits.

Pour atténuer les constatations d'audit non conformes

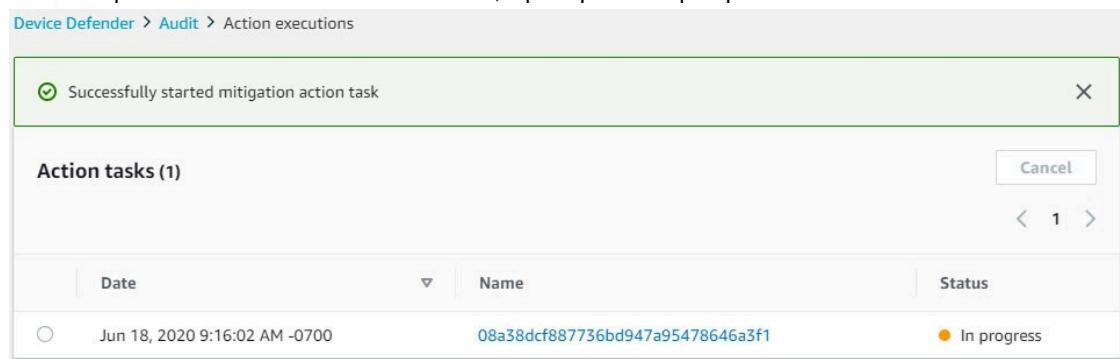
- Ouvrez la [console AWS IoT](#).
- Dans le volet de navigation de gauche, choisissez Audit, puis Résultats. Sélectionnez le nom de l'audit auquel vous souhaitez répondre.
- Vérifiez vos résultats. Notez que Logging disabled se trouve sous Contrôles non conformes.
- Tâche de sélection/Démarrer des actions d'atténuation.

The screenshot shows the Audit findings section for an audit task started on Jun 16, 2020 at 11:23:17 PM -0700. It lists two non-compliant checks: "IoT policies overly permissive" (Critical) and "Logging disabled" (Low). A red circle highlights the "Start mitigation actions" button. Below the findings, there's a section for mitigation actions with a note: "You don't have any Audit action tasks yet."

- UNDERSELECT actions, sélectionnez les actions appropriées pour chaque constatation non conforme afin de résoudre les problèmes.



6. Sélectionnez Confirm (Confirmer).
7. Une fois que l'action d'atténuation est lancée, il peut prendre quelques minutes.



Pour vérifier que l'action d'atténuation a fonctionné

1. Dans AWS dans le volet de navigation, sélectionnez Paramètres.
2. Confirmation que Journaux sont Enabled et l'Niveau de détail est Error.

Activer les notifications SNS (facultatif)

Dans la procédure suivante, vous activez les notifications SNS (Simple Notifications Service) pour vous alerter lorsque vos audits identifient des ressources non conformes. Dans ce didacticiel, vous allez configurer des notifications pour les contrôles d'audit activés dans le [Activer les contrôles d'audit \(p. 977\)](#) Didacticiel de

1. Dans un premier temps, vous devez créer une stratégie IAM qui donne accès à Amazon SNS via le service AWS Management Console. Pour ce faire, suivez la page [Création d'une AWS IoT Device Defender Rôle Audit IAM \(facultatif\) \(p. 988\)](#) processus, mais en sélectionnant AWS IoT Device Defender PublishFindingsToSNSSMitigationAction à l'étape 8.
2. Dans [AWS IoT console](#), dans le volet de navigation, développez Défendre et sélectionnez Paramètres.
3. UNDER Alertes SNS, sélectionnez Modifier.

The screenshot shows the AWS IoT Device Defender Audit interface. On the left, a sidebar lists 'Defend' (Audit, Detect, Mitigation actions), 'Settings' (Act, Test), and 'Audit'. The main area displays two audit checks:

- Role Alias allows access to unused services [?](#) Medium Role alias
- Logging disabled [?](#) Low Account settings

A large blue 'Disable' button is centered below the checks. Below this, a section titled 'Disabled Audit checks' states 'All Audit checks are currently enabled'. At the bottom, a section titled 'SNS alerts' shows 'SNS alerts are not configured' and features a blue 'Edit' button.

4. Dans la page **Modifier les alertes SNS**, sélectionnez **Activé**. UNDER Rubrique, sélectionnez **Créer**. Nommez la rubrique **Notifications_IoTDD** et sélectionnez **Créer**. UNDER Rôle, sélectionnez le rôle que vous venez de créer appelé **AWSIoTDeviceDefenderAudit**.

The screenshot shows the 'Edit SNS alerts' dialog. It includes the following fields:

- Enabled:** A radio button labeled 'Enabled' is selected, while 'Disabled' is unselected.
- Topic:** A dropdown menu shows 'No topic selected' with a 'Create' button (circled in red) and a 'Select' button.
- Role:** A dropdown menu shows 'No role selected' with a 'Select' button (circled in red).
- Buttons:** 'Cancel' and 'Update' buttons at the bottom right.

Tâche de sélection Mise à jour.

Edit SNS alerts

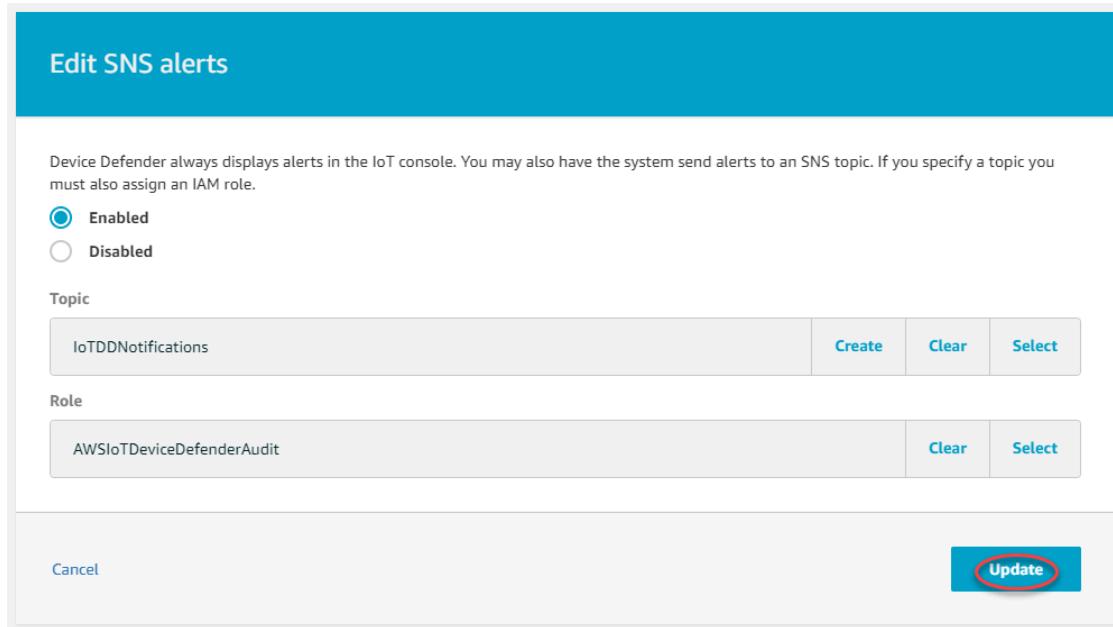
Device Defender always displays alerts in the IoT console. You may also have the system send alerts to an SNS topic. If you specify a topic you must also assign an IAM role.

Enabled
 Disabled

Topic
IoTDDNotifications Create Clear Select

Role
AWSIoTDeviceDefenderAudit Clear Select

Cancel Update



Si vous souhaitez recevoir des e-mails ou des SMS sur vos plates-formes Ops via SNS, consultez la section [Utilisation d'Amazon SNS pour les notifications utilisateur](#).

Activer la journalisation (facultatif)

Cette procédure explique comment activer AWS IoT pour consigner des informations dans CloudWatch Logs. Cela vous permettra d'afficher les résultats de votre audit. L'activation de la journalisation peut entraîner des frais.

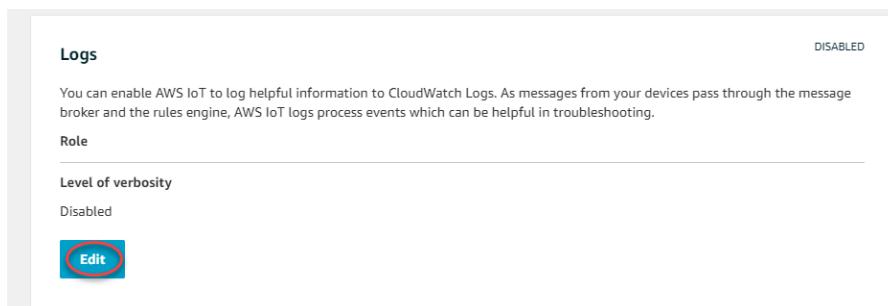
Pour activer la journalisation

1. Dans AWS dans le volet de navigation, sélectionnez Paramètres.
2. UNDER Journaux, sélectionnez Modifier.

Software Logs DISABLED

Settings Edit Learn

Logs
You can enable AWS IoT to log helpful information to CloudWatch Logs. As messages from your devices pass through the message broker and the rules engine, AWS IoT logs process events which can be helpful in troubleshooting.
Role _____
Level of verbosity
Disabled



3. UNDER Niveau de détail, sélectionnez Débogage (le plus verbeux).
4. UNDER Définir le rôle, sélectionnez Créeation d'un rôle et nommez le rôle **Rôle de journalisation AWS IOT**. Une stratégie sera automatiquement attachée.

Configure role setting

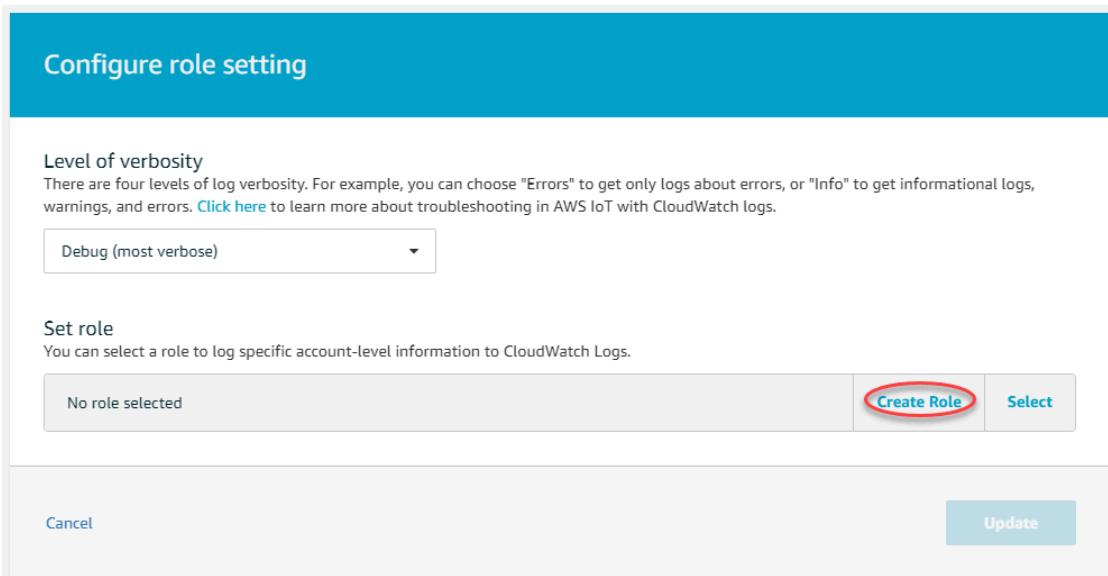
Level of verbosity
There are four levels of log verbosity. For example, you can choose "Errors" to get only logs about errors, or "Info" to get informational logs, warnings, and errors. [Click here](#) to learn more about troubleshooting in AWS IoT with CloudWatch logs.

Debug (most verbose) ▾

Set role
You can select a role to log specific account-level information to CloudWatch Logs.

No role selected [Create Role](#) [Select](#)

[Cancel](#) [Update](#)



Tâche de sélection Mise à jour.

Configure role setting

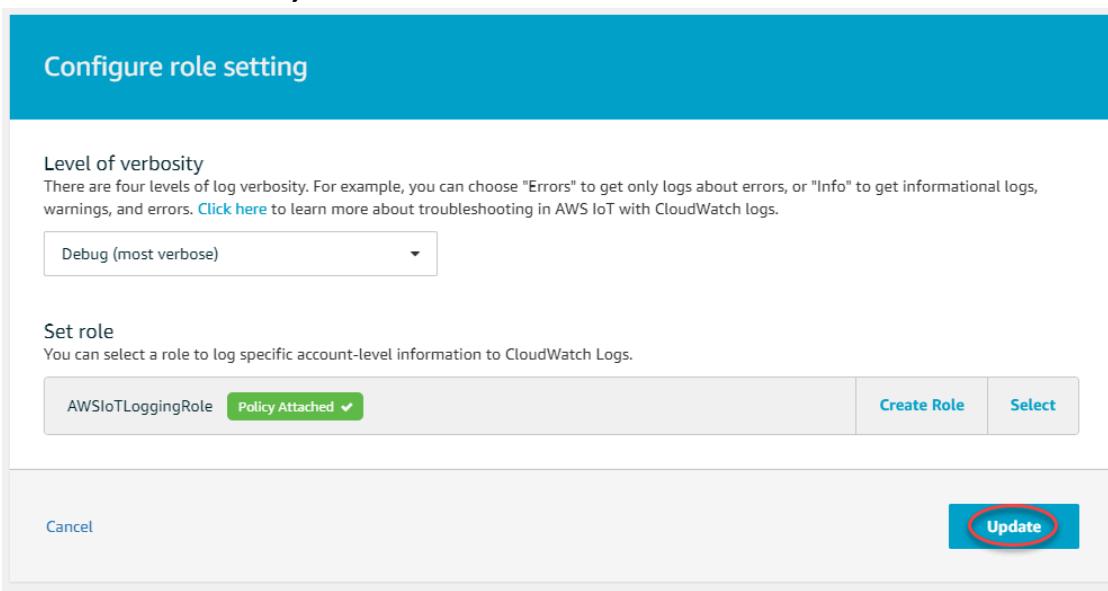
Level of verbosity
There are four levels of log verbosity. For example, you can choose "Errors" to get only logs about errors, or "Info" to get informational logs, warnings, and errors. [Click here](#) to learn more about troubleshooting in AWS IoT with CloudWatch logs.

Debug (most verbose) ▾

Set role
You can select a role to log specific account-level information to CloudWatch Logs.

AWSIoTLoggingRole Policy Attached ▾ [Create Role](#) [Select](#)

[Cancel](#) [Update](#)



Création d'uneAWS IoT Device DefenderRôle Audit IAM (facultatif)

Dans la procédure suivante, vous créez unAWS IoT Device DefenderRôle d'audit IAM qui fournitAWS IoT Device DefenderAccès en lecture àAWS IoT.

1. Accédez à la console IAM à l'adresse suivante :<https://console.aws.amazon.com/iam/>
2. Dans le volet de navigation, choisissez.Userspuis, choisissezAjouter un utilisateur.
3. Dans Nom d'utilisateur, saisissez **Administrator**.
4. Activez la case à cocher près deAWS Accès à Management Console.. Ensuite, sélectionnez Mot de passe personnalisé, puis saisissez 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. Désactivez la case en regard de 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 deAWSIoTDeviceDefenderAudit.
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 Crée un utilisateur.

Guide de ML

Dans ce guide de démarrage, vous allez créer un profil de sécurité ML Detect qui utilise l'apprentissage automatique (ML) pour créer des modèles de comportement attendu sur la base des données métriques historiques de vos appareils. Pendant que ML Detect crée le modèle de ML, vous pouvez suivre sa progression. Une fois le modèle de machine learning créé, vous pouvez visualiser et analyser les alertes de manière continue et atténuer les problèmes identifiés.

Pour plus d'informations sur ML Detect et ses commandes API et CLI, consultez[Déetectez ML \(p. 1100\)](#).

Ce chapitre contient les sections suivantes :

- [Prérequis \(p. 989\)](#)
- [Comment utiliser ML Detect dans la console \(p. 989\)](#)
- [Comment utiliser ML Detect avec l'interface de ligne de commande \(p. 1003\)](#)

Prérequis

- Un Compte AWS. Si vous ne disposez pas de ce mode de spécification, consultez [la section Configuration](#).

Comment utiliser ML Detect dans la console

Didacticiels

- [Activer ML Detect \(p. 989\)](#)
- [Surveillez l'état de votre modèle de machine learning \(p. 994\)](#)
- [Passez en revue vos alertes ML Detect \(p. 995\)](#)
- [Ajustez vos alertes de machine learning \(p. 997\)](#)
- [Marquez l'état de vérification de votre alerte \(p. 998\)](#)
- [Atténuer les problèmes identifiés sur les appareils \(p. 999\)](#)

Activer ML Detect

Les procédures suivantes expliquent comment configurer ML Detect dans la console.

1. Tout d'abord, assurez-vous que vos appareils créeront les points de données minimaux requis, tels que définis dans les [exigences minimales de ML Detect \(p. 1102\)](#) pour la formation continue et

l'actualisation du modèle. Pour que la collecte de données progresse, assurez-vous que votre profil de sécurité est associé à une cible, qui peut être un objet ou un groupe d'objets.

2. Dans la [AWS IoTconsole](#), dans le volet de navigation, développez Defend. Choisissez Déetecter, Profils de sécurité, Créer un profil de sécurité, puis Créer un profil de détection d'anomalies ML.
3. Dans la page Définir les configurations de base, procédez de la façon suivante.
 - Sous Target, choisissez vos groupes d'appareils cibles.
 - Sous Nom du profil de sécurité, entrez le nom de votre profil de sécurité.
 - (Facultatif) Sous Description, vous pouvez écrire une brève description du profil ML.
 - Sous Comportements de mesures sélectionnés dans le profil de sécurité, choisissez les mesures que vous souhaitez surveiller.

Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notifications
Authorization failures	Cloud-side	High	1	1	Suppressed
Connection attempts	Cloud-side	High	1	1	Suppressed
Disconnects	Cloud-side	High	1	1	Suppressed
Message size	Cloud-side	High	1	1	Suppressed
Messages received	Cloud-side	High	1	1	Suppressed
Messages sent	Cloud-side	High	1	1	Suppressed

Lorsque vous avez terminé, sélectionnez Next.

4. Sur la page Définir le SNS (facultatif), spécifiez une rubrique SNS pour les notifications d'alarme lorsqu'un appareil enfreint un comportement de votre profil. Choisissez un rôle IAM que vous utiliserez pour publier dans la rubrique SNS sélectionnée.

Si vous ne possédez pas encore de rôle SNS, procédez comme suit pour créer un rôle avec les autorisations et les relations de confiance requises.

- Accédez à la [console IAM](#). Dans le panneau de navigation, choisissez Roles (Rôles), puis Create role (Créer un rôle).
- Sous Sélectionner le type d'entité sécurisée, sélectionnez AWSService. Ensuite, sous Choisir un cas d'utilisation, choisissez IoT et sous Sélectionnez votre cas d'utilisation, choisissez IoT - Device Defender Mitigation Actions. Lorsque vous avez terminé, choisissez Next : Autorisations.
- Sous Politiques d'autorisations associées, assurez-vous que cette option AWSIoTDeviceDefenderPublishFindingsToSNSSMitigationActionest sélectionnée, puis choisissez Suivant : Tags.

Create role

1 2 3 4

Attached permissions policies

The type of role that you selected requires the following policy.

Filter policies		Search	Showing 6 results
Policy name	Used as	Description	
▶ AWSIoTDeviceDefenderAddThingsToThingGrou...	Permissions policy (1)	Provides write access to IoT thing groups and r...	
▶ AWSIoTDeviceDefenderEnableIoTLoggingMitig...	Permissions policy (2)	Provides access for enabling IoT logging for ex...	
▶ AWSIoTDeviceDefenderPublishFindingsToSNS...	None	Provides messages publish access to SNS topi...	
▶ AWSIoTDeviceDefenderReplaceDefaultPolicyMi...	None	Provides write access to IoT policies for execut...	
▶ AWSIoTDeviceDefenderUpdateCACertMitigatio...	None	Provides write access to IoT CA certificates for ...	
▶ AWSIoTDeviceDefenderUpdateDeviceCertMitig...	None	Provides write access to IoT certificates for exe...	

Set permissions boundary

* Required

Cancel

Previous

Next: Tags

- Sous Ajouter des balises (facultatif), vous pouvez ajouter toutes les balises que vous souhaitez associer à votre rôle. Lorsque vous avez terminé, sélectionnez Next: Review (Suivant : vérification).
- Sous Révision, attribuez un nom à votre rôle et assurez-vous qu'il AWSIoTDeviceDefenderPublishFindingsToSNSSMitigationActionest répertorié sous Autorisations et AWSservice : iot.amazonaws.com est répertorié sous Relations de confiance. Lorsque vous avez terminé, choisissez Create un rôle.

Identity and Access Management (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 publish SNS notifications | [Edit](#)

Instance Profile ARNs: [Edit](#)

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 **Add inline policy**

Policy name	Policy type
AWSIoTDeviceDefenderPublishFindingsToSNSSMitigationAction	AWS managed policy

▶ Permissions boundary (not set)

Identity and Access Management (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 publish SNS notifications | [Edit](#)

Instance Profile ARNs: [Edit](#)

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**

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

Edit trust relationship

Trusted entities
The following trusted entities can assume this role.

Conditions
The following conditions define how and when trusted entities can assume the role.

Trusted entities
The identity provider(s) [iot.amazonaws.com](#) ...
There are no conditions associated with this role.

- Sur la page Modifier le comportement de la métrique, vous pouvez personnaliser vos paramètres de comportement de machine learning.

The screenshot shows the 'Edit metric behaviors' step of a configuration wizard. The left sidebar indicates 'Step 1 Set basic configurations' is complete, 'Step 2 - optional Edit metric behaviors' is active, and 'Step 3 Review configuration' is next. The main area displays three behavior types:

- Authorization failures:** Behavior name: Authorization_failures_ML_behavior, Metric: Authorization failures. Trigger and clear thresholds: 1. Notifications: Suppressed. ML Detect confidence: High.
- Bytes in:** Behavior name: Bytes_in_ML_behavior, Metric: Bytes in. Trigger and clear thresholds: 1. Notifications: Suppressed. ML Detect confidence: High.
- Connection attempts:** Behavior name: Connection_attempts_ML_behavior, Metric: Connection attempts. Trigger and clear thresholds: 1. Notifications: Suppressed. ML Detect confidence: High.

6. Lorsque vous avez terminé, sélectionnez Next.
7. Sur la page Révision de la configuration, vérifiez les comportements que vous souhaitez que l'apprentissage automatique surveille, puis choisissez Suivant.

Review configuration

Security Profile basic configuration

Profile name	Target	Description
Smart_lights_ML_Detect_Security_Profile	All registered things	ML Detect security profile for monitoring smart lights

Selected metric behaviors in Security Profile

Behavior name	Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	No
Authorization_failures_ML_behavior	Authorization failures	Cloud-side	High	1	1	Sup
Bytes_out_ML_behavior	Bytes out	Device-side	High	1	1	Sup
Connection_attempts_ML_behavior	Connection attempts	Cloud-side	High	1	1	Sup
Disconnects_ML_behavior	Disconnects	Cloud-side	High	1	1	Sup

- Après avoir créé votre profil de sécurité, vous êtes redirigé vers la page Profils de sécurité, où le profil de sécurité nouvellement créé apparaît.

Note

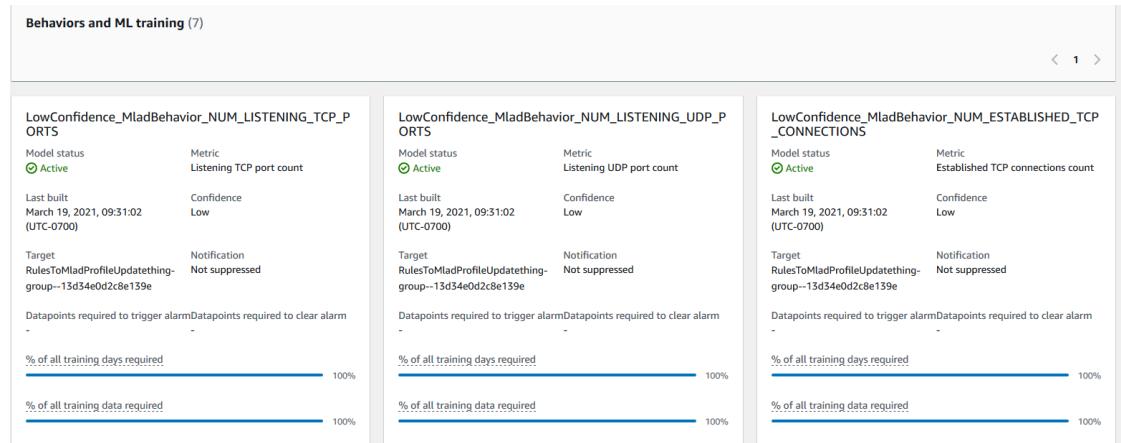
La formation initiale et la création du modèle de machine learning prennent 14 jours. Vous pouvez vous attendre à recevoir des alertes une fois l'opération terminée, en cas d'activité anormale sur vos appareils.

Surveillez l'état de votre modèle de machine learning

Pendant la période de formation initiale de vos modèles de machine learning, vous pouvez suivre leur progression à tout moment en suivant les étapes suivantes.

- Dans le volet de navigation de la [AWS IoT console](#), développez Défendre, puis choisissez Détecter, Profils de sécurité.
- Sur la page Profils de sécurité, choisissez le profil de sécurité que vous souhaitez consulter. Choisissez ensuite Comportements et formation en machine learning.
- Sur la page Comportements et formation en machine learning, vérifiez la progression de l'entraînement de vos modèles de machine learning.

Une fois que le statut de votre modèle est actif, il commence à prendre des décisions de détection en fonction de votre utilisation et met à jour le profil tous les jours.



Note

Si votre modèle ne progresse pas comme prévu, assurez-vous que vos appareils répondent aux exigences [Configuration requise \(p. 1102\)](#).

Passez en revue vos alarmes ML Detect

Une fois que vos modèles de machine learning sont créés et prêts pour l'inférence des données, vous pouvez régulièrement consulter et analyser les alarmes identifiées par les modèles.

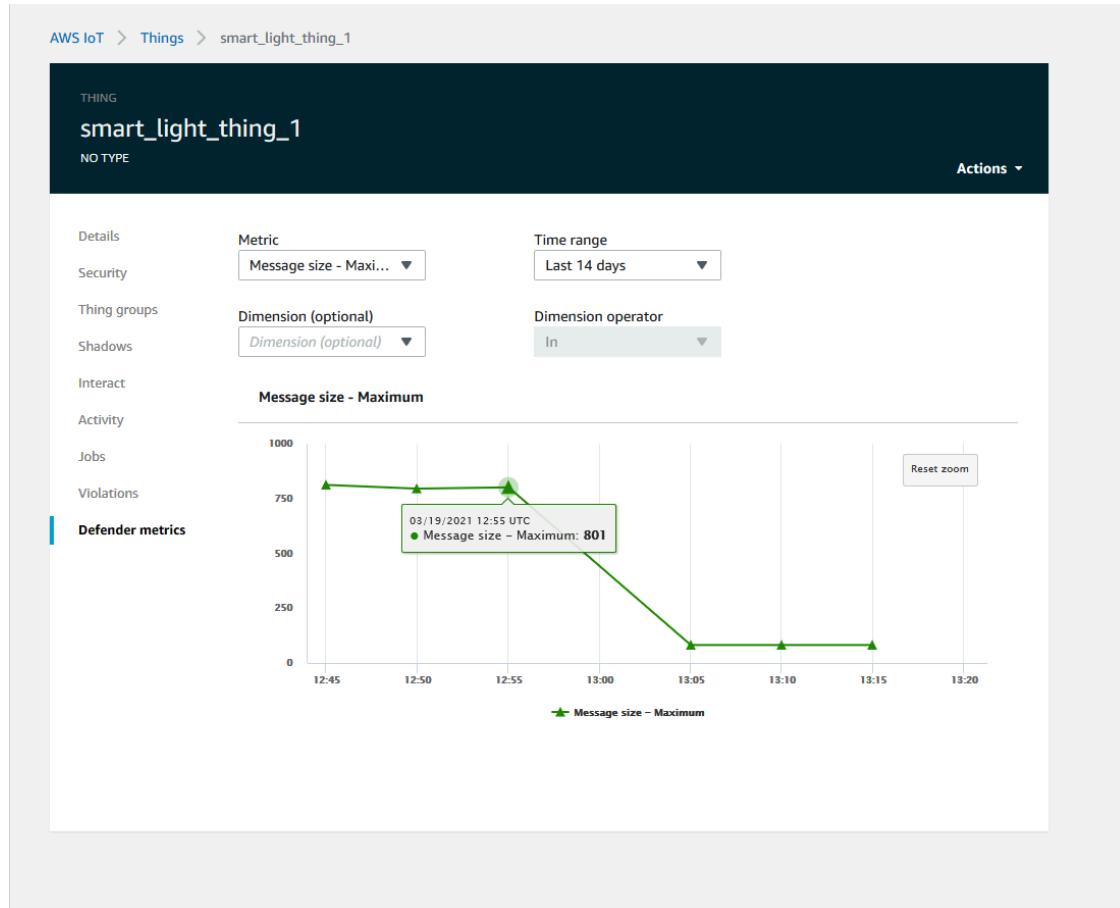
- Dans le volet de navigation de la [AWS IoTconsole](#), développez Defend, puis choisissez Detect, Alarms.

First event	Thing name	Security Profile	Behavior type	Behavior name	Last emitted	Verification state	Confidence
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-539d0ef0-3504-4a9c-a7a1-be53b472b850	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-81fc61d7-9362-4c87-ad6-333891ff7349	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-23e8ec9a-2162-456e-a8c2-302c3826f618	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-85e0278e-29aa-4554-b3b6-111b9063228f	fdsa	Rule-based	Authorization_failures_behavior (Notification: on)	Authorization failures: 0 failure(s)	Unknown	-

- Si vous accédez à l'onglet Historique, vous pouvez également consulter les informations relatives à vos appareils qui ne sont plus en mode d'alarme.



Pour plus d'informations, sous Gérer, choisissez Objets, choisissez l'objet pour lequel vous souhaitez obtenir plus de détails, puis accédez aux statistiques de Defender. Vous pouvez accéder au graphique des statistiques de Defender et effectuer votre enquête sur tout élément alarmant depuis l'onglet Actif. Dans ce cas, le graphique montre une augmentation de la taille du message, qui a déclenché l'alarme. Vous pouvez voir que l'alarme est ensuite effacée.



Ajustez vos alarmes de machine learning

Une fois que vos modèles de machine learning sont créés et prêts pour l'évaluation des données, vous pouvez mettre à jour les paramètres de comportement du machine learning de votre profil de sécurité pour modifier la configuration. La procédure suivante vous montre comment mettre à jour les paramètres de ML de votre profil de sécurité dans le AWS CLI.

1. Dans le volet de navigation de la [AWS IoT console](#), développez Défendre, puis choisissez Détecter, Profils de sécurité.
2. Dans la page des profils de sécurité, cochez la case en regard du profil de sécurité. Choisissez ensuite Actions, puis Modifier.

Security Profile	Threshold type	Behaviors	Metrics retained	Target	Creation date	Notifications
Smart_lights_ML_Detect_Security_Profile	ML	9	-	All registered things	March 17, 2021, 12:58:14 (UTC-0700)	Suppressed (9)
MyEmptyGroupSP	ML	6	-	EmptyGroup	March 16, 2021, 17:52:01 (UTC-0700)	Suppressed (6)

3. Sous Définir les configurations de base, vous pouvez ajuster les groupes d'objets cibles du profil de sécurité ou modifier les mesures que vous souhaitez surveiller.

Set basic configurations Info

Select target and metrics that you would like to configure for your ML Security Profile.

Security Profile basic configuration

Target
Choose target device group(s)
All registered things

Security Profile name
Smart_lights_ML_Detect_Security_Profile

Enter a unique name containing only letters, numbers, hyphens, colon, or underscores. A Security Profile name cannot contain any spaces.

Description - optional
ML Detect security profile for monitoring smart lights

Selected metric behaviors in Security Profile (6) Info

You can assess how your fleet of devices is operating across the following metric behaviors.

Delete	Add cloud-side metric	Add device-side metric				
<input type="checkbox"/>	Metric	Type	ML Detect confidence	Datapoints required to trigger alarm	Datapoints required to clear alarm	Notifications
<input type="checkbox"/>	Authorization failures	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Connection attempts	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Disconnects	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Message size	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Messages received	Cloud-side	High	1	1	Suppressed
<input type="checkbox"/>	Messages sent	Cloud-side	High	1	1	Suppressed

4. Vous pouvez mettre à jour l'un des éléments suivants en accédant à Modifier les comportements des métriques.
 - Les points de données de votre modèle de machine learning sont nécessaires pour déclencher l'alarme
 - Les points de données de votre modèle de machine learning sont nécessaires pour effacer l'alarme
 - Votre niveau de confiance dans ML Detect
 - Vos notifications ML Detect (par exemple, Non supprimées, Supprimées)

The screenshot shows the 'Edit metric behaviors - optional' page in the AWS IoT Device Defender. It displays three separate configuration sections:

- Authorization failures:**
 - Behavior name: Authorization_failures_ML_behavior
 - Metric: Authorization failures
 - Datapoints required to trigger alarm: 1
 - Datapoints required to clear alarm: 1
 - Notifications: Suppressed
 - ML Detect confidence: High
- Bytes out:**
 - Behavior name: Bytes_out_ML_behavior
 - Metric: Bytes out
 - Datapoints required to trigger alarm: 1
 - Datapoints required to clear alarm: 1
 - Notifications: Suppressed
 - ML Detect confidence: High
- Connection attempts:**
 - Behavior name: Connection_attempts_ML_behavior
 - Metric: Connection attempts
 - Datapoints required to trigger alarm: 1
 - Datapoints required to clear alarm: 1
 - Notifications: Suppressed
 - ML Detect confidence: High

Marquez l'état de vérification de votre alarme

Marquez vos alarmes en définissant l'état de vérification et en fournissant une description de cet état de vérification. Cela vous permet, à vous et à votre équipe, d'identifier les alarmes auxquelles vous n'avez pas à répondre.

1. Dans le volet de navigation de la [AWS IoTconsole](#), développez Defend, puis choisissez Detect, Alarms. Sélectionnez une alarme pour indiquer son état de vérification.

The screenshot shows the AWS IoT Device Defender Detect Alarms interface. At the top, there are tabs for Active and History. Below is a table titled 'All alarms (1/5) Info' with columns: First event, Thing name, Security Profile, Behavior type, Behavior name, Last emitted, Verification state, and Confidence. One row is selected, and its details are shown in a modal window.

First event	Thing name	Security Profile	Behavior type	Behavior name	Last emitted	Verification state	Confidence
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c	fdsa	Rule-based	Authorization_failures_behavior [Notification: on]	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-539d0ef0-3504-4a9c-a7a1-be53b472b50	fdsa	Rule-based	Authorization_failures_behavior [Notification: on]	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-81fc61d7-9362-4c87-ad46-333891f7349	fdsa	Rule-based	Authorization_failures_behavior [Notification: on]	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-23e8ec9a-2162-456e-a8c2-302c3826f18	fdsa	Rule-based	Authorization_failures_behavior [Notification: on]	Authorization failures: 0 failure(s)	Unknown	-
September 03, 2021, 15:50:00 (UTC-0700)	iotconsole-85e0278e-29aa-4554-b3b6-111b9063228f	fdsa	Rule-based	Authorization_failures_behavior [Notification: on]	Authorization failures: 0 failure(s)	Unknown	-

2. Choisissez Marquer l'état de vérification. Le modal d'état de vérification s'ouvre.
3. Choisissez l'état de vérification approprié, entrez une description de vérification (facultatif), puis choisissez Marquer. Cette action attribue un état de vérification et une description à l'alarme choisie.

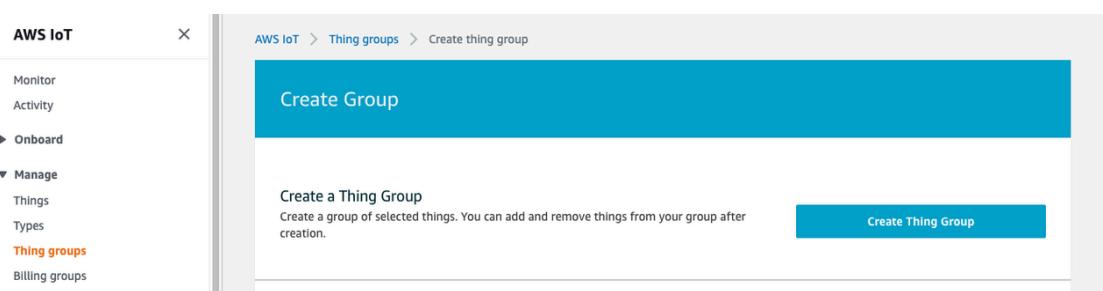
The screenshot shows a 'Mark verification state' modal window. It contains a heading 'Select verification state' and a note about providing AWS with information about alarm verification states. A dropdown menu lists five options: Unknown, True positive, False positive, Benign positive, and Unknown. The 'Unknown' option is currently selected. At the bottom right of the modal are 'Cancel' and 'Mark' buttons.

Atténuer les problèmes identifiés sur les appareils

1. (Facultatif) Avant de configurer les mesures d'atténuation de la quarantaine, configurons un groupe de quarantaine vers lequel nous déplacerons l'appareil en infraction. Vous pouvez également utiliser un groupe existant.
2. Accédez à Gérer, Groupes d'objets, puis Créez un groupe d'objets. Nommez votre groupe d'objets. Dans le cadre de ce didacticiel, nous nommons notre groupe d'objets Quarantine_group. Sous Groupe d'objets, Sécurité, appliquez la politique suivante au groupe d'objets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "iot:UpdateThingShadow"
      ...
    }
  ]
}
```

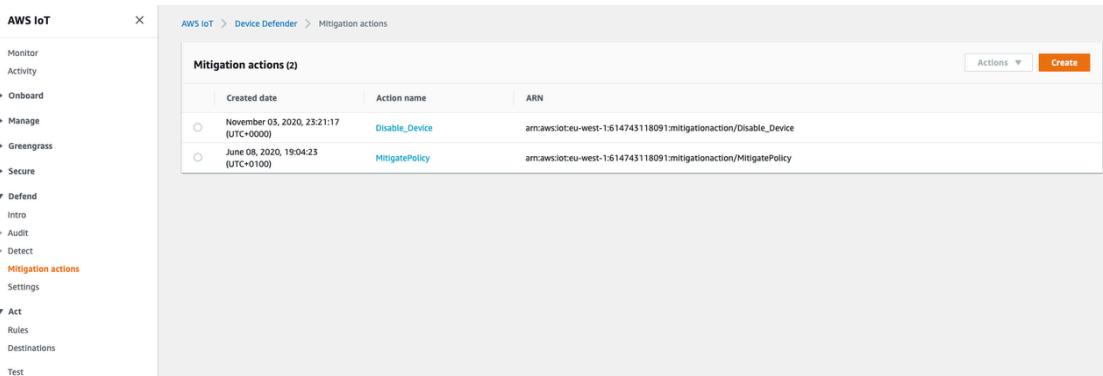
```
{
    "Effect": "Deny",
    "Action": "iot:/*",
    "Resource": "*",
}
}
```



Lorsque vous avez terminé, choisissez Create un groupe d'objets.

3. Maintenant que nous avons créé un groupe d'objets, créons une action d'atténuation qui déplace les appareils alarmés vers leQuarantine_group.

Sous Défendre, Actions d'atténuation, choisissez Créer.



4. Dans la page Create d'une nouvelle action d'atténuation, entrez les informations suivantes.

- Nom de l'action : donnez un nom à votre action d'atténuation, par exemple**Quarantine_action**.
- Type d'action : choisissez le type d'action. Nous allons choisir Ajouter des éléments au groupe d'objets (Audit ou Détecter l'atténuation).
- Rôle d'exécution de l'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 ongoing detect monitoring. There are predefined actions for the different audit checks and detect alarms to help you resolve issues quickly.

Action name [Info](#)

Action type [Info](#)

Permissions

Please create or select a role with the following mitigation action type specific permission(s) and trust relationship.

Required permissions: [Manage your service permissions ↗](#)

- ▶ Permissions
- ▶ Trust relationships

You can also attach an action specific managed policy to an existing role, or create a new role with the required managed policy attached.

Action execution role [Info](#)
 [Create Role](#) [Select](#)

Parameters

Thing groups [Info](#)

1 thing group(s) selected. [Close](#)

Thing groups	Summary
<input type="text" value="Quarantine_group"/>	

Lorsque vous avez terminé, sélectionnez Save. Vous disposez désormais d'une action d'atténuation qui déplace les appareils en état d'alarme vers un groupe d'objets en quarantaine, et d'une action d'atténuation qui permet d'isoler l'appareil pendant que vous enquêtez.

5. Accédez à Defender, Detect, Alarms. Vous pouvez voir quels appareils sont en état d'alarme sous Actif.

The screenshot shows the AWS IoT Device Defender Detect Alarms interface. At the top, there are tabs for Active and History. Below is a table titled "All alarms (5) Info" with columns: First event, Thing name, Security Profile, Behavior type, Behavior name, Last emitted, Verification state, and Confidence. The table lists five alarms, each with a timestamp, thing name (e.g., iotconsole-6f8379bc-c245-4ffe-8ef7-b2b52e78975c), security profile (fdsa), rule-based behavior, and authorization failure notifications. The last emitted time is September 03, 2021, at 15:50:00 UTC-0700. The verification state is Unknown for all, and confidence is also Unknown.

Sélectionnez l'appareil que vous souhaitez déplacer vers le groupe de quarantaine et choisissez Démarrer les actions d'atténuation.

6. Sous Démarrer les actions d'atténuation, Démarrer les actions, sélectionnez l'action d'atténuation que vous avez créée précédemment. Par exemple, nous allons choisir **Quarantine_action**, puis sélectionner Démarrer. La page des tâches d'action s'ouvre.

The screenshot shows the "Start mitigation actions" dialog box. It has a title bar "Start mitigation actions" and a close button. The main area contains the text "Select actions for mitigation." and "Things effected by the selected alarm(s) ddml7". Below this is a section titled "Select Actions" with the note "The sequence of action executions follows the order of selected action(s)". A dropdown menu "Choose actions(s) to execute" is open, showing the option "Quarantine_action" with a checked checkbox. At the bottom, there is a checkbox "I understand that the selected mitigation action(s) may not be reversible." followed by "Cancel" and "Start" buttons.

7. L'appareil est maintenant isolé **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 d'objets ou prendre d'autres mesures.

Action tasks (1)						
Date	Task ID	Action name	Action type	Action parameter (1)	Action parameter (2)	Action Executions
December 02, 2020, 14:19:57 (UTCZ)	73fad2ea-9bd8-48d0-af5a-3dbc120b91e7	Quarantine_action	Add things to thing group	Thing group(s): Quarantine_group	Override dynamic groups: false	Successful

Comment utiliser ML Detect avec l'interface de ligne de commande

Ce qui suit vous montre comment configurer ML Detect à l'aide de la CLI.

Didacticiels

- [Activer ML Detect \(p. 1003\)](#)
- [Surveillez l'état de votre modèle de machine learning \(p. 1005\)](#)
- [Passez en revue vos alarmes ML Detect \(p. 1006\)](#)
- [Ajustez vos alarmes de machine learning \(p. 1007\)](#)
- [Marquez l'état de vérification de votre alarme \(p. 1009\)](#)
- [Atténuer les problèmes identifiés sur les appareils \(p. 1009\)](#)

Activer ML Detect

La procédure suivante vous montre comment activer ML Detect dans AWS CLI.

1. Assurez-vous que vos appareils créeront les points de données minimaux requis, tels que définis dans les [exigences minimales de ML Detect \(p. 1102\)](#) pour la formation continue et l'actualisation du modèle. Pour que la collecte de données progresse, assurez-vous que vos objets se trouvent dans un groupe d'objets associé à un profil de sécurité.
2. Créez un profil de sécurité ML Detect à l'aide de la [create-security-profile](#) commande. L'exemple suivant crée un profil de sécurité nommé **security-profile-for-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 indique mlDetectionConfig que la métrique 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  
}],'
```

Sortie :

```
{  
    "securityProfileName": "security-profile-for-smart-lights",  
    "securityProfileArn": "arn:aws:iot:eu-west-1:123456789012:securityprofile/security-  
profile-for-smart-lights"  
}
```

3. Ensuite, associez votre profil de sécurité à un ou plusieurs groupes d'éléments. Utilisez la [attach-security-profile](#) commande pour associer un groupe d'objets à votre profil de sécurité. L'exemple suivant associe un groupe d'objets nommé **ML_Detect_beta_static_group** au profil de sécurité **security-profile-for-smart-lights**.

```
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
```

Sortie :

Aucun.

4. Une fois que vous avez créé votre profil de sécurité complet, le modèle de machine learning commence à s'entraîner. La formation initiale et la création du modèle de machine learning prennent 14 jours. Au bout de 14 jours, en cas d'activité anormale sur votre appareil, vous pouvez vous attendre à recevoir des alertes.

Surveillez l'état de votre modèle de machine learning

La procédure suivante vous montre comment suivre la formation en cours de ML.

- Utilisez la `get-behavior-model-training-summaries` commande pour visualiser la progression de votre modèle de machine learning. L'exemple suivant montre le résumé de la progression de la formation du modèle de machine learning pour le profil de sécurité `security-profile-for-smart-lights`. `modelStatus` vous indique si un modèle a terminé la formation ou s'il est toujours en attente de création pour un comportement particulier.

```
aws iot get-behavior-model-training-summaries \
--security-profile-name security-profile-for-smart-lights
```

Sortie :

```
{
  "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 répondent aux exigences [Configuration requise \(p. 1102\)](#).

Passez en revue vos alarmes ML Detect

Une fois que vos modèles de machine learning sont créés 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 le AWS CLI.

- Pour voir toutes les alarmes actives, utilisez la [list-active-violations](#) commande.

```
aws iot list-active-violations \  
--max-results 2
```

Sortie :

```
{  
    "activeViolations": []  
}
```

Vous pouvez également consulter toutes les violations découvertes au cours d'une période donnée à l'aide de la [list-violation-events](#) commande. L'exemple suivant répertorie les événements de violation survenus entre le 22 septembre 2020 à 5:42:13 GMT et le 26 octobre 2020 à 5:42:13 GMT.

```
aws iot list-violation-events \  
--start-time 1599500533 \  
--end-time 1600796533 \  
--max-results 2
```

Sortie :

```
{  
    "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": "Amo6XIUrsoohsojuIG6TuwSR3X9iUvH20CksBZg6bed2j21VSnD1uP1pf1xKX1+a3cvBRSosIB0xFv40kM6RYBknZ/vxabMe/ZW31Ps/WiZH1r9Wg7R7eEGli59IJ/U0iBQ1McP/ht0E2XA2TTIvYeMmKQQPsRj/eoV9j7P/wveu7sknGepU/mvpV002Ap7hnV5U+Prx/9+iJA/341va+pQww7jpUeHmJN9Hw4MqW0ysw0Ry3w38h0QWEpz2xwFWAxAARxeIxCxt5c37RK/1RZB1hYqoB+w2PZ74730h8pICGY4gktJxkwHyyRabpSM/G/f5DFrD905v8idkTZzBxW2jrbzSUIdafPtsZHL/yAMKr3HAKtaAbZ2nTs0BNre7X2d/jIjjarhon0Dh91+8I9Y5Ey+DIFBcqFTvhbKAafQt3gs6CUiqHdWiCenfJyb8whmDE2qxvdxE1GmRb+k6kuN5jrZxxw95gzfYDgRHv11iEn8h1qZLD0czkIFBpMppHj9cetHPvM+qffXGAzKi8tL6eQuCdMLXmVE3jbqcJck9ItnaYJi5zKDz9FVbrz9qZZPtZJFHp"
}
```

Ajustez vos alarmes de machine learning

Une fois que vos modèles de machine learning sont créés et prêts pour l'évaluation des données, vous pouvez mettre à jour les paramètres de comportement du machine learning de votre profil de sécurité pour modifier la configuration. La procédure suivante vous montre comment mettre à jour les paramètres de ML de votre profil de sécurité dans le AWS CLI.

- Pour modifier les paramètres de comportement du machine learning de votre profil de sécurité, utilisez la [update-security-profile](#) commande. L'exemple suivant met à jour les comportements du profil de sécurité **security-profile-for-smart-lights** en modifiant certains comportements et en annulant la suppression des notifications pour tous les comportements.confidenceLevel

```
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"
            }
        }
    ]
}'
```

```

        "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
}
]
'
```

Sortie :

```
{
    "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
}
```

Marquez l'état de vérification de votre alarme

Vous pouvez attribuer des états de vérification à vos alarmes afin de les classer et d'analyser les anomalies.

- Marquez vos alarmes avec un état de vérification et une description de cet état. Par exemple, pour définir l'état de vérification d'une alarme sur Faux positif, utilisez la commande suivante :

```
aws iot put-verification-state-on-violation --violation-id 12345 --verification-
state FALSE_POSITIVE --verification-state-description "This is dummy description" --
endpoint https://us-east-1.iot.amazonaws.com --region us-east-1
```

Sortie :

Aucun.

Atténuer les problèmes identifiés sur les appareils

1. Utilisez la [create-thing-group](#) commande pour créer un groupe d'objets pour l'action d'atténuation. Dans l'exemple suivant, nous créons un groupe d'objets appelé ThingGroupForDetectMitigationAction.

```
aws iot create-thing-group --thing-group-name ThingGroupForDetectMitigationAction
```

Sortie :

```
{
    "thingGroupName": "ThingGroupForDetectMitigationAction",
    "thingGroupArn": "arn:aws:iot:us-
east-1:123456789012:thinggroup/ThingGroupForDetectMitigationAction",
    "thingGroupId": "4139cd61-10fa-4c40-b867-0fc6209dca4d"
}
```

2. Ensuite, utilisez la [create-mitigation-action](#) commande pour créer une action d'atténuation. Dans l'exemple suivant, nous créons une action d'atténuation appelée detect_mitigation_action avec 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 objets vers le groupe d'objets que nous avons créé précédemment, appelé ThingGroupForDetectMitigationAction.

```
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  
    }  
}'
```

Sortie :

```
{  
    "actionArn": "arn:aws:iot:us-  
east-1:123456789012:mitigationaction/detect_mitigation_action",  
    "actionId": "5939e3a0-bf4c-44bb-a547-1ab59ffe67c3"  
}
```

- Utilisez la [start-detect-mitigation-actions-task](#) commande pour démarrer votre tâche de mesures d'atténuation. task-id,target et 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
```

Sortie :

```
{  
    "taskId": "taskIdForMitigationAction"  
}
```

- (Facultatif) Pour afficher les exécutions des actions d'atténuation incluses dans une tâche, utilisez la [list-detect-mitigation-actions-executions](#) commande.

```
aws iot list-detect-mitigation-actions-executions \  
    --task-id taskIdForMitigationAction \  
    --max-items 5 \  
    --page-size 4
```

Sortie :

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

- (Facultatif) Utilisez la [describe-detect-mitigation-actions-task](#) commande pour obtenir des informations sur une tâche d'action d'atténuation.

```
aws iot describe-detect-mitigation-actions-task \  
    --task-id taskIdForMitigationAction
```

Sortie :

```
{
    "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/MitigationActionValidRole",
                "actionParams": {
                    "addThingsToThingGroupParams": {
                        "thingGroupNames": [
                            "ThingGroupForDetectMitigationAction"
                        ],
                        "overrideDynamicGroups": false
                    }
                }
            }
        ],
        "taskStatistics": {
            "actionsExecuted": 0,
            "actionsSkipped": 0,
            "actionsFailed": 0
        }
    }
}
```

- (Facultatif) Pour obtenir la liste de vos tâches d'atténuation, utilisez la [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
```

Sortie :

```
{
    "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 la [cancel-detect-mitigation-actions-task](#) commande.

```
aws iot cancel-detect-mitigation-actions-task \
--task-id taskIdForMitigationAction
```

Sortie :

Aucun.

Personnalisez quand et comment vous affichez AWS IoT Device Defender Résultats de l'audit

AWS IoT Device Defender audit fournit des contrôles de sécurité périodiques pour confirmer que les appareils et les ressources respectent 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 résultant de problèmes connus répétés, la fonction de suppression de la 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 pendant une période pré-déterminée. Un résultat de vérification qui a été supprimé est classé comme résultat supprimé, distinct des catégories conformes et non conformes. Cette nouvelle catégorie ne déclenche pas d'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.

Commencer

Les sections suivantes expliquent comment utiliser les suppressions de recherche d'audit pour supprimer unDevice certificate expiringEnregistrez 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 qui arrivent à expiration pour que Device Defender puisse détecter.

Utilisez les éléments suivants pour créer vos certificats.

- [Création et enregistrement d'un certificat d'autorité de certification \(p. 324\)](#)
- [Création d'un certificat client à l'aide de votre certificat d'autorité de certification](#). À l'étape 3, définissez votredaysParamètre 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 résoudra le problème. Nous créons une suppression des résultats d'audit pour chaque certificat afin d'empêcher le résultat de l'audit d'être non conforme pour la semaine prochaine.

1. Nous allons d'abord effectuer un audit à la demande pour montrer que la vérification du certificat de l'appareil expirée n'est pas conforme.

De la[AWS IoT console](#), choisissezDéfendreà partir de la barre latérale gauche, puisAudit, puisRésultats. Dans la pageRésultats de l'audit, choisissezCréer. LeCréer un nouvel auditLa fenêtre s'ouvre. Sélectionnez Create (Créer).

▼ Defend

Intro

▼ Audit

Results

Schedules

Action executions

Finding suppressions

new

► Detect

Les résultats de l'audit à la demande montrent que « Certificat de périphérique expirant » n'est pas conforme pour deux ressources.

2. Maintenant, nous aimerais désactiver l'avertissement de vérification non conforme « Certificat de périphérique expirant », car nos développeurs testent de nouvelles fonctionnalités qui corrigeraient l'avertissement.

À partir de la barre latérale gauche sous Défendre, choisissez Audit, puis Trouver des suppressions. Dans la page Vérification de la recherche de suppressions, choisissez Créer.

POLICIES

CAs

Role Aliases

Authorizers

▼ Defend

Intro

▼ Audit

Results

Schedules

Action executions

Finding suppressions

3. Dans la pageCréation d'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 aimerais supprimer.
 - Identificateur de ressource : Nous saisissons l'ID du certificat de l'appareil de l'un des certificats pour lesquels nous souhaitons supprimer les résultats de l'audit.
 - Durée de suppression : Il est sélectionné :1 week, parce que c'est pendant combien de temps nous aimerais supprimer laDevice certificate expiring vérification d'audit pour.
 - Description (facultative) : Nous ajoutons une note qui explique pourquoi nous supprimons cette constatation d'audit.

Create an audit finding suppression

Suppressing an audit finding on a specified resource and its associated device. Supressing an audit finding on a specified resource and its associated device will make the device non-compliant.

Audit check

Device certificate expiring

Resource identifier

Device certificate id

b4490bd64c5cf85182f3182f1c03e70017e483f17b

Suppression duration

1 week

Description (optional)

Developer updates

Une fois que nous avons rempli les champs, choisissezCréer. Une bannière de succès s'affiche après la création de la suppression des résultats de l'audit.

4. Nous avons supprimé un résultat d'audit pour l'un des certificats et nous devons maintenant supprimer le résultat de l'audit pour le deuxième certificat. Nous pourrions utiliser la même méthode de suppression que celle utilisée à l'étape 3, mais nous utiliserons une méthode différente à des fins de démonstration.

À partir de la barre latérale gauche sousDéfendre, choisissezAudit, puisRésultats. Dans la pageRésultats de l'audit, choisissez l'audit avec la ressource non conforme. Sélectionnez ensuite la ressource sousContrôles non conformes. Dans notre cas, nous sélectionnons « Certificat d'appareil expirant ».

5. Dans la pageCertificat de l'appareil expirantpage, sousRègles non conformesChoisissez le bouton d'option en regard de la recherche qui doit être supprimée. Choisissez ensuite la caseActionsmenu déroulant, puis choisissez la durée pour laquelle vous souhaitez trouver une suppression. Dans notre cas, nous choisissons1 weekcomme nous l'avons fait pour l'autre certificat. Dans la pageVérifiez la suppressionFenêtre, choisissezActiver la suppression.

2 OT 195 device certificates non-compliant

Mitigation

Consult your security best practices for how to proceed.

1. Provision a new certificate and attach it to the device.
2. Verify that the new certificate is valid and the device can connect.
3. Mark the old certificate as "INACTIVE" in the AWS IoT Device Defender console.
4. Detach the old certificate from the device. (See [Detaching a certificate](#))

Non-compliant certificate (2)

Finding



28022a890964e991852c79a28a83eb89



dc9b109c705ed7e68588bc54eef86f1c

Une bannière de succès s'affiche après la création de la suppression des résultats de l'audit. Désormais, les deux résultats de l'audit ont été supprimés pendant une semaine, tandis que nos développeurs travaillent sur une solution pour répondre à l'avertissement.

Personnalisez vos résultats d'audit dans l'interface de ligne de commande

La procédure pas à pas suivante utilise un compte avec un certificat d'appareil 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 résoudra le problème. Nous créons une suppression des résultats d'audit pour le certificat afin d'empêcher le résultat de l'audit d'être non conforme pour la semaine prochaine.

Nous utilisons les commandes CLI suivantes.

- [suppression de création-audit](#)
- [Décrire-Audit Suppression](#)
- [suppression de mise à jour et d'audit](#)
- [suppression/suppression de l'audit](#)
- [suppressions d'audit de liste](#)

1. Utilisez la commande suivante pour activer l'audit.

```
aws iot update-account-audit-configuration \
--audit-check-configurations "{\"DEVICE_CERTIFICATE_EXPIRING_CHECK\":{\"enabled \
\":true}}"
```

Sortie :

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
```

Sortie :

```
{ \
  "taskId": "787ed873b69cb4d6cdbae6ddd06996c5"
}
```

3. Utilisation de l'[describe-account-audit configuration](#) pour 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
```

Sortie :

```
{
```

```
"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": {
    "AUTENTICATED_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
    },
    "UNAUTENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK": {
        "enabled": false
    }
}
}
```

DEVICE_CERTIFICATE_EXPIRING_CHECK doit avoir une valeur à true.

4. Utilisation de l'[tâche d'audit de liste](#) pour identifier les tâches d'audit terminées.

```
aws iot list-audit-tasks \
--task-status "COMPLETED" \
--start-time 2020-07-31 \
--end-time 2020-08-01
```

Sortie :

```
{  
  "tasks": [  
    {  
      "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",  
      "taskStatus": "COMPLETED",  
      "taskType": "SCHEDULED_AUDIT_TASK"  
    }  
  ]  
}
```

La taskID de l'audit que vous avez effectué à l'étape 1 devrait comporter un taskStatus de COMPLETED.

5. Utilisation de l'[Décrire-Audit Task](#) pour obtenir des détails sur l'audit terminé à l'aide de la commande taskID sortie de l'étape précédente. Cette commande répertorie les détails de votre audit.

```
aws iot describe-audit-task \  
  --task-id "787ed873b69cb4d6cdbae6ddd06996c5"
```

Sortie :

```
{  
  "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'[résultats de l'audit de la liste](#) pour rechercher l'ID de certificat non conforme afin que nous puissions suspendre les alertes d'audit de cette ressource.

```
aws iot list-audit-findings \  
  --start-time 2020-07-31 \  
  --end-time 2020-08-01
```

Sortie :

```
{  
  "findings": [  
    {  
      "findingId": "296cccd39f806bf9d8f8de20d0ceb33a1",  
      "taskId": "787ed873b69cb4d6cdbae6ddd06996c5",  
      "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",  
      "resourceArn": "arn:aws:iot:us-east-1:123456789012:cert/12345678901234567890123456789012",  
      "status": "PENDING",  
      "lastModified": 1596168096.157  
    }  
  ]  
}
```

```
"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": "787ed873b69cb4d6cd8ae6ddd06996c5",  
    "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'[suppression de création-audit](#) pour supprimer les notifications pour le DEVICE_CERTIFICATE_EXPIRING_CHECK vérification d'audit pour un certificat d'appareil avec l'identifiant **c7691e<shortened>** jusqu'à **2020-08-20**.

```
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 l'audit de liste](#) pour confirmer le paramètre de suppression de l'audit et obtenir des détails sur la suppression.

```
aws iot list-audit-suppressions
```

Sortie :

```
{  
    "suppressions": [  
        {  
            "checkName": "DEVICE_CERTIFICATE_EXPIRING_CHECK",  
            "resourceIdentifier": {  
                "deviceCertificateId": "c7691e<shortened>"
```

```
        },
        "expirationDate": 1597881600.0,
        "suppressIndefinitely": false
    }
}
```

9. Lessuppression de mise à jour et d'auditpeut être utilisée pour mettre à jour la suppression des résultats d'audit. L'exemple ci-dessous met à jour la `expiration-date`pour `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. Lessuppression/suppression de l'auditpeut 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 la casesuppressions d'audit de listecommande.

```
aws iot list-audit-suppressions
```

Sortie :

```
{
    "suppressions": []
}
```

Dans ce tutoriel, nous vous avons montré comment supprimer un `Device certificate expiring`Enregistrez la console et l'interface de ligne de commande. Pour de plus amples informations sur la recherche d'audit, consultez[Vérification de la recherche de suppressions \(p. 1081\)](#)

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 présentant cette 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.

Élevée

Les contrôles d'audit non conformes d'une telle gravité nécessitent une investigation urgente et une planification des mesures correctives une fois les problèmes critiques résolus. 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 présentant une telle gravité présentent des problèmes qui nécessitent une attention particulière dans le cadre de la maintenance 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 d'une telle gravité indiquent souvent que les meilleures 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 \(p. 1024\)](#). 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 :

- [Autorité de certification intermédiaire révoquée pour vérification des certificats d'appareils actifs \(p. 1025\)](#)
- [Le certificat CA révoqué est toujours actif \(p. 1026\)](#)
- [Certificat d'appareil partagé \(p. 1027\)](#)

- [Qualité clé du certificat de l'appareil \(p. 1028\)](#)
- [Certificat \(p. 1029\)](#)
- [Le rôle Cognito non authentifié est trop permissif \(p. 1030\)](#)
- [Le rôle Cognito authentifié est trop permissif \(p. 1035\)](#)
- [AWS IoT politiques trop permissives \(p. 1042\)](#)
- [AWS IoT politique potentiellement mal configurée \(p. 1046\)](#)
- [Alias de rôle trop permissif \(p. 1049\)](#)
- [L'alias de rôle permet d'accéder aux services inutilisés \(p. 1050\)](#)
- [Certificat \(p. 1051\)](#)
- [Identifiants de clients MQTT en conflit \(p. 1052\)](#)
- [Certificat de dispositif expir \(p. 1053\)](#)
- [Le certificat d'appareil révoqué est toujours actif \(p. 1053\)](#)
- [Journalisation désactivée \(p. 1054\)](#)

Autorité de certification intermédiaire révoquée pour vérification des certificats d'appareils actifs

Utilisez cette vérification pour identifier tous les certificats d'appareils associés qui sont toujours actifs malgré la révocation d'une autorité de certification intermédiaire.

Cette vérification apparaît comme `INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATES_CHECK` dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

Les codes de motif sont renvoyés lorsque ce contrôle trouve une non-conformité :

- `INTERMEDIATE_CA_REVOKED_BY_ISSUER`

Pourquoi est-ce important ?

L'autorité de certification intermédiaire révoquée pour les certificats de dispositif actifs vérifie l'identité et la confiance des appareils, en déterminant s'il existe des certificats de dispositif actifs dans AWS IoT Core lesquels les autorités de certification émettrices intermédiaires ont été révoquées dans la chaîne des autorités de certification.

Une autorité de certification intermédiaire révoquée ne doit plus être utilisée pour signer d'autres certificats d'autorité de certification ou d'appareil dans la chaîne d'autorités de certification. Les appareils récemment ajoutés avec des certificats signés à l'aide de ce certificat d'autorité de certification après la révocation de l'autorité de certification intermédiaire constitueront une menace de sécurité.

Comment réparer

Passez en revue l'activité d'enregistrement du certificat de l'appareil pour la période qui a suivi la révocation du certificat CA. Suivez vos meilleures pratiques en matière de sécurité pour atténuer la situation. Il se peut que vous souhaitiez :

1. Fournissez de nouveaux certificats, signés par une autorité de certification différente, pour les appareils concernés.

2. Vérifiez que les nouveaux certificats sont valides et que les appareils peuvent les utiliser pour se connecter.
3. Permet [UpdateCertificate](#) de marquer l'ancien certificat comme étant 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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.
 - Passez en revue l'activité d'enregistrement du certificat d'appareil pour la période suivant la révocation du certificat d'autorité de certification intermédiaire et envisagez de révoquer tous les certificats d'appareil qui auraient pu être émis pendant cette période. Vous pouvez l'utiliser [ListRelatedResourcesForAuditFinding](#) pour répertorier les certificats d'appareil signés par le certificat CA et [UpdateCertificate](#) pour révoquer un certificat d'appareil.
 - Détacher l'ancien certificat de l'appareil. (Consultez [DetachThingPrincipal](#).)

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Le certificat CA révoqué est toujours actif

Un certificat CA a été révoqué, mais demeure actif dans AWS IoT.

Cette vérification apparaît comme REVOKED_CA_CERTIFICATE_STILL_ACTIVE_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation pour implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

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. Vous pouvez utiliser [ListCertificatesByCA](#) pour répertorier les certificats d'appareil signés par le certificat CA et [UpdateCertificate](#) pour révoquer un certificat d'appareil.

Certificat d'appareil partagé

Plusieurs connexions simultanées utilisent le même certificat X.509 pour s'authentifier auprès d'AWS IoT.

Cette vérification apparaît commeDEVICE_CERTIFICATE_SHARED_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

Lorsqu'elle est effectuée dans le cadre d'un audit à la demande, cette vérification examine les certificats et les identifiants clients utilisés par les appareils pour se connecter au cours des 31 jours précédent le début de l'audit et jusqu'à 2 heures avant l'exécution du contrôle. Pour les audits planifiés, cette vérification examine les données entre 2 heures avant la dernière exécution de l'audit et 2 heures avant le début de cette instance d'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. Permet [UpdateCertificate](#) de marquer l'ancien certificat comme étant 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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

4. Détacher l'ancien certificat de chacun des appareils.

Qualité clé du certificat 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 clés 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 — Vérifiez si la clé a été générée à l'aide d'OpenSSL 0.9.8c-1 jusqu'aux versions antérieures à 0.9.8g-9 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 — Vérifiez si la clé a été générée par la bibliothèque Infineon RSA 1.02.013 du microprogramme du module TPM (Infineon Trusted Platform Module), par exemple les versions antérieures à 0000000000000422 — 4.34, antérieures à 000000000000062b — 6.43 et antérieures à 0000000000008521 — 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. Parmi les technologies concernées, citons BitLocker le TPM 1.2, la génération de clés PGP YubiKey 4 (avant la version 4.3.5) et la fonctionnalité de chiffrement des données utilisateur mises en cache de Chrome OS.

AWS IoT Device Defender déclare les certificats non conformes s'ils échouent à ces tests.

Cette vérification apparaît commeDEVICE_CERTIFICATE_KEY_QUALITY_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

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. Permet [UpdateCertificate](#) de marquer l'ancien certificat comme étant 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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

4. Détacher l'ancien certificat de chacun des appareils.

Certificat

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 clés 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 — Vérifiez si la clé a été générée à l'aide d'OpenSSL 0.9.8c-1 jusqu'aux versions antérieures à 0.9.8g-9 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 — Vérifiez si la clé a été générée par la bibliothèque Infineon RSA 1.02.013 du microprogramme du module TPM (Infineon Trusted Platform Module), par exemple les versions antérieures à 0000000000000422 — 4.34, antérieures à 000000000000062b — 6.43 et antérieures à 0000000000008521 — 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. Parmi les technologies concernées, citons BitLocker le TPM 1.2, la génération de clés PGP YubiKey 4 (avant la version 4.3.5) et la fonctionnalité de chiffrement des données utilisateur mises en cache de Chrome OS.

AWS IoT Device Defender déclare les certificats non conformes s'ils échouent à ces tests.

Cette vérification apparaît comme CA_CERTIFICATE_KEY_QUALITY_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

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.)

Le rôle Cognito non authentifié est trop permissif

Une politique attachée à un rôle de pool d'identités Amazon Cognito Amazon Cognito non authentifié AWS IoT

- 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 pool d'identités Amazon Cognito non authentifié ne doivent disposer que d'une autorisation limitée pour publier et s'abonner à des rubriques MQTT spécifiques ou utiliser les commandes de l'API pour lire et modifier des données spécifiques relatives aux données fictives ou à l'exécution de tâches.

Cette vérification apparaît comme UNAUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

Pour cette AWS IoT Device Defender vérification, audite tous les groupes d'identités Amazon Cognito qui ont été utilisés pour se connecter à l'agent de AWS IoT messages au cours des 31 jours précédent l'exécution de l'audit. Tous les pools d'identités Amazon Cognito à partir desquels une identité Amazon Cognito authentifiée ou non authentifiée est connectée sont inclus dans l'audit.

Les codes de raison suivants sont renvoyés lorsque cette vérification détecte un rôle de pool d'identités Amazon Cognito non conforme et non authentifié :

- `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 important 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 politique attachée à un rôle de pool d'identités Amazon Cognito Amazon Cognito non authentifié Nous vous recommandons la procédure suivante :

1. Créez un nouveau rôle conforme.
2. Créez un pool d'identités Amazon Cognito et associez-y le rôle conforme.
3. Vérifiez que vos identités peuvent accéder à AWS IoT à l'aide du nouveau groupe.
4. Une fois la vérification terminée, associez le rôle conforme au pool d'identités Amazon Cognito qui a été marqué comme non conforme.

Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquez l'action `PUBLISH_FINDINGS_TO_SNS` d'atténuation pour implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Gérer ou modifier des objets

Les actions d'API AWS IoT suivantes sont utilisées pour gérer ou modifier des objets. L'autorisation d'effectuer ces actions ne doit pas être accordée aux appareils qui se connectent via un pool d'identités Amazon Cognito non authentifié.

- `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 via un pool d'identités Amazon Cognito non authentifié 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 via un pool d'identités Amazon Cognito non authentifié ne doivent pas être autorisés à effectuer des actions d'AWS IoT API autres que celles décrites dans ces sections. Vous pouvez gérer votre compte à l'aide d'une application qui se connecte via un pool d'identités Amazon Cognito non authentifié en créant un pool d'identités distinct 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:topicfilter/$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.

Le rôle Cognito authentifié est trop permissif

Une politique attachée à un rôle de pool d'identités Amazon Cognito Amazon CognitoAWS IoT

- 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 pool d'identités Amazon Cognito authentifié ne doivent disposer que d'autorisations limitées pour lire des données administratives spécifiques, publier et s'abonner à des rubriques MQTT spécifiques à un objet, ou utiliser les commandes de l'API pour lire et modifier des données spécifiques relatives aux données fictives ou d'exécution de tâches.

Cette vérification apparaît comme AUTHENTICATED_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

Pour cette AWS IoT Device Defender vérification, audite tous les groupes d'identités Amazon Cognito Cognito Cognito Cognito qui ont été utilisés pour se connecter à l'agent de AWS IoT messages messages au cours des 31 jours précédent l'exécution de l'audit. Tous les pools d'identités Amazon Cognito à partir desquels une identité Amazon Cognito authentifiée ou non authentifiée est connectée sont inclus dans l'audit.

Les codes de raison suivants sont renvoyés lorsque cette vérification détecte un rôle de pool d'identités Amazon Cognito authentifié 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 politique attachée à un rôle de pool d'identités Amazon Cognito Nous vous recommandons la procédure suivante :

1. Créez un nouveau rôle conforme.
2. Créez un pool d'identités Amazon Cognito et associez-y le rôle conforme.
3. Vérifiez que vos identités peuvent accéder à AWS IoT à l'aide du nouveau groupe.
4. Une fois la vérification terminée, associez le rôle au pool d'identités Amazon Cognito qui a été marqué comme non conforme.

Vous pouvez également utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation pour implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Gérer ou modifier des objets

Les actions d'AWS IoT API suivantes sont utilisées pour gérer ou modifier des éléments. L'autorisation de les exécuter ne doit donc pas être accordée aux appareils se connectant via un pool d'identités Amazon Cognito authentifié :

- 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 via un pool d'identités Amazon Cognito authentifié ne doivent pas être autorisés à effectuer des actions d'AWS IoT API autres que celles décrites dans ces sections. Pour gérer votre compte à l'aide d'une application qui se connecte via un pool d'identités Amazon Cognito authentifié, créez un pool d'identités distinct non utilisé par les appareils.

Lire les données administratives d'objet.

Les actions d'AWS IoT API suivantes sont utilisées pour lire les données des objets. Les appareils qui se connectent via un pool d'identités Amazon Cognito authentifié doivent donc être autorisés à les exécuter uniquement sur un ensemble limité d'éléments :

- 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/*"
            ]
        }
    ]
}
```

```

        "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::client/*
```

Le caractère générique « * » permet à n'importe quel appareil de se connecter à AWS IoT.

```
arn:aws:iot::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:topicfilter/$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 IoT politiques trop permissives

Une stratégie AWS IoT accorde des autorisations qui sont trop larges ou illimitées. Il permet d'envoyer ou de recevoir des messages MQTT pour un large éventail de périphériques, ou d'accéder aux données d'exécution des tâches et d'accès à ces données pour un large éventail d'appareils ou de les modifier.

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 apparaît comme IOT_POLICY_OVERLY_PERMISSIVE_CHECK dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

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 soumis à une politique trop permissive peuvent, s'ils sont compromis, avoir un impact 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. [CreatePolicyVersion](#) À utiliser pour créer une nouvelle version conforme de la politique. Définissez l'indicateur `setAsDefault` sur true. (Cela rend cette nouvelle version opérationnelle pour toutes les entités qui utilisent la stratégie.)
2. Permet d'[ListTargetsForPolicy](#) obtenir la liste des cibles (certificats, groupes d'objets) auxquelles la politique est associée et de déterminer quels appareils font partie des 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 ne parvient pas à se connecter, utilisez [SetPolicyVersion](#) pour rétablir la politique par défaut à la version précédente, révisez la politique et réessayez.

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.
- Appliquez l'action `PUBLISH_FINDINGS_TO_SNS` d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Utilisez les [variables de stratégie AWS IoT \(p. 362\)](#) 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:topicfilter/$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.

AWS IoT politique potentiellement mal configurée

Une AWS IoT politique a été identifiée comme potentiellement mal configurée. Des politiques mal configurées, notamment des politiques trop permissives, peuvent provoquer des incidents de sécurité, tels que l'accès des appareils à des ressources involontaires.

La vérification AWS IoT de la politique potentiellement mal configurée est un avertissement vous demandant de vous assurer que seules les actions prévues sont autorisées avant de mettre à jour la politique.

Dans la CLI et l'API, cette vérification apparaît sous la forme IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK.

Gravité : Moyenne

Détails

AWS IoT renvoie le code de raison suivant lorsque cette vérification détecte une AWS IoT politique potentiellement mal configurée :

- POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT
- FILTRES_SUJETS DESTINÉS À REFUSER L'UTILISATION DE CARACTÈRES GÉNÉRIQUES

Pourquoi est-ce important ?

Des politiques mal configurées peuvent avoir des conséquences imprévues en fournissant aux appareils plus d'autorisations que nécessaire. Nous vous recommandons d'examiner attentivement la politique afin de limiter l'accès aux ressources et de prévenir les menaces de sécurité.

La politique contient des caractères génériques MQTT dans un exemple de déclaration de refus

La AWS IoT politique de vérification potentiellement mal configurée inspecte la présence de caractères génériques MQTT (+ou#) dans les instructions de refus. Les caractères génériques sont traités comme des chaînes littérales par AWS IoT les politiques et peuvent rendre la politique trop permissive.

L'exemple suivant vise à refuser l'abonnement à des rubriques liées à `building/control_room` l'utilisation du caractère générique MQTT# dans les politiques. Toutefois, les caractères génériques MQTT n'ont pas de caractère générique dans AWS IoT les politiques auxquelles les appareils peuvent s'abonner `building/control_room/data1`.

La vérification AWS IoT de la politique potentiellement mal configurée signalera cette politique avec un code de raison `POLICY_CONTAINS_MQTT_WILDCARDS_IN_DENY_STATEMENT`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/#"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    }
  ]
}
```

Voici un exemple de stratégie Les appareils ne sont pas autorisés à s'abonner aux sous-sujets `debuilding/control_room/` et ne sont pas autorisés à recevoir des messages provenant de sujets secondaires `debuilding/control_room/`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:region:account-id:topicfilter/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Subscribe",
      "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"
    },
    {
      "Effect": "Allow",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/*"
    },
    {
      "Effect": "Deny",
      "Action": "iot:Receive",
      "Resource": "arn:aws:iot:region:account-id:topic/building/control_room/*"
    }
  ]
}
```

Exemple de filtres de rubrique destinés à refuser l'autorisation à l'aide de caractères génériques

L'exemple de politique suivant vise à refuser l'abonnement à des rubriques associées `building/control_room` en refusant la ressource `building/control_room/*`. Toutefois, les appareils peuvent envoyer des demandes d'abonnement `building/#` et recevoir des messages provenant de tous les sujets liés à `building`, y compris `building/control_room/data1`.

La vérification AWS IoT de la politique potentiellement mal configurée signalera cette politique avec un code de raison `TOPIC_FILTERS_INTENDED_TO_DENY_ALLOWED_USING_WILDCARDS`.

L'exemple de politique suivant contient des autorisations pour recevoir des messages sur `building/control_room` topics :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iot:Receive",  
            "Resource": "arn:aws:iot:region:account-id:topic/building/*"  
        }  
    ]  
}
```

Voici un exemple de stratégie. Les appareils ne sont pas autorisés à s'abonner aux sous-sujets `debuilding/control_room/` et ne sont pas autorisés à recevoir des messages provenant de sujets secondaires `debuilding/control_room/`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:region:account-id:topicfilter/building/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "iot:Subscribe",  
            "Resource": "arn:aws:iot:region:account-id:topicfilter/building/control_room/*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": "iot:Receive",  
            "Resource": "arn:aws:iot:region:account-id:topic/building/*"  
        },  
        {  
            "Effect": "Deny",  
            "Action": "iot:Receive",  
            "Resource": "arn:aws:iot:region:account-id:topic/building/control_room/*"  
        }  
    ]  
}
```

}

Note

Cette vérification peut signaler des faux positifs. Nous vous recommandons d'évaluer toutes les politiques signalées et de signaler les ressources faussement positives à l'aide de suppressions d'audit.

Comment réparer

Cette vérification signale les politiques potentiellement mal configurées, de sorte qu'il peut y avoir des faux positifs. Marquez tous les faux positifs à l'aide de [suppressions d'audit \(p. 1081\)](#) afin qu'ils ne soient pas signalés à l'future.

Vous pouvez également suivre ces étapes pour corriger les politiques non conformes associées à des objets, à des groupes d'objets ou à d'autres entités :

1. [CreatePolicyVersion](#) À utiliser pour créer une nouvelle version conforme de la politique. Définissez l'indicateur `setAsDefault` sur true. (Cela rend cette nouvelle version opérationnelle pour toutes les entités qui utilisent la stratégie.)

Pour des exemples de création deAWS IoT politiques pour des cas d'utilisation courants, consultez les [exemples de politiques de publication/d'abonnement \(p. 376\)](#) dans le Guide duAWS IoT Core développeur.

2. Vérifiez que tous les appareils associés sont en mesure de se connecter à AWS IoT. Si un appareil ne parvient pas à se connecter, utilisez [SetPolicyVersion](#)pour rétablir la politique par défaut à la version précédente, révisez la politique et réessayez.

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.
- Appliquez l'action `PUBLISH_FINDINGS_TO_SNS` d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Utilisez les [variables de stratégie AWS IoT \(p. 362\)](#) pour faire référence de manière dynamique aux ressources AWS IoT de vos stratégies.

Alias de rôle trop permissif

AWS IoT's alias du rôle fournit un mécanisme permettant aux dispositifs connectés de s'authentifier auprès de àAWS IoT l'aide de certificats X.509, puis d'obtenir desAWS informations d'identification de durée limitée à partir d'un rôle IAM associé à un alias deAWS IoT 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 priviléges. 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 apparaît comme `IOT_ROLE_ALIAS_OVERLY_PERMISSIVE_CHECK` dans l'interface de ligne de commande et l'API.

Gravité : critique

Détails

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 d'appels directs vers des AWS services à l'aide du fournisseur AWS IoT Core d'informations d'identification \(p. 402\)](#) pour appliquer une stratégie plus restrictive à votre alias de rôle.

Vous pouvez utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquez l'action `PUBLISH_FINDINGS_TO_SNS` d'atténuation si vous souhaitez implémenter une action personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

L'alias de rôle permet d'accéder aux services inutilisés

AWS IoT alias du rôle fournit un mécanisme permettant aux dispositifs connectés de s'authentifier auprès de AWS IoT l'aide de certificats X.509, puis d'obtenir des AWS informations d'identification de durée limitée à partir d'un rôle IAM associé à un alias de AWS IoT 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 priviléges. 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, l'audit indique si vous avez un rôle IAM lié à l'alias de rôle qui n'a été utilisé AWS IoT que l'année dernière, mais que la politique associée au rôle accorde également des autorisations à "iam:getRole" et "dynamodb:PutItem".

Cette vérification apparaît comme `IOT_ROLE_ALIAS_ALLOWS_ACCESS_TO_UNUSED_SERVICES_CHECK` dans l'interface de ligne de commande et l'API.

Gravité : Moyenne

Détails

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 d'appels directs vers des AWS services à l'aide du fournisseur AWS IoT Core d'informations d'identification \(p. 402\)](#) pour appliquer une stratégie plus restrictive à votre alias de rôle.

Vous pouvez utiliser des actions d'atténuation pour effectuer les actions suivantes :

- Appliquez l'action `PUBLISH_FINDINGS_TO_SNS` d'atténuation si vous souhaitez implémenter une action personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Certificat

Un certificat CA expire sous 30 jours ou a expiré.

Cette vérification apparaît comme `CA_CERTIFICATE_EXPIRING_CHECK` dans l'interface de ligne de commande et l'API.

Gravité : Moyenne

Détails

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.
 - Appliquez l'action PUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Identifiants de clients MQTT en conflit

Plusieurs appareils se connectent en utilisant le même ID client.

Cette vérification apparaît comme CONFLICTING_CLIENT_IDS_CHECK dans l'interface de ligne de commande et l'API.

Gravité : Elevée

Détails

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 :

- Appliquez l'actionPUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

Certificat de dispositif expiré

Un certificat d'appareil expire sous 30 jours ou a expiré.

Cette vérification apparaît commeDEVICE_CERTIFICATE_EXPIRING_CHECK dans l'interface de ligne de commande et l'API.

Gravité : Moyenne

Détails

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. Permet [UpdateCertificate](#)de marquer l'ancien certificat comme INACTIF dansAWS 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.
 - Appliquez l'actionPUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

4. Détacher l'ancien certificat de l'appareil. (Consultez [DetachThingPrincipal](#).)

Le certificat d'appareil révoqué est toujours actif

Un certificat d'appareil révoqué est toujours actif.

Cette vérification apparaît commeREVOKEDED_DEVICE_CERTIFICATE_STILL_ACTIVE_CHECK dans l'interface de ligne de commande et l'API.

Gravité : Moyenne

Détails

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. Permet [UpdateCertificate](#)de marquer l'ancien certificat comme étant RÉVOQUÉ dansAWS 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.
 - Appliquez l'actionPUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

4. Détacher l'ancien certificat de l'appareil. (Consultez [DetachThingPrincipal](#).)

Journalisation désactivée

AWS IoT les journaux ne sont pas activés sur Amazon CloudWatch. Vérifie la journalisation V1 et V2.

Cette vérification apparaît commeLOGGING_DISABLED_CHECK dans l'interface de ligne de commande et l'API.

Gravité : Faible

Détails

Les codes de motif sont renvoyés lorsque ce contrôle trouve une non-conformité :

- LOGGING_DISABLED

Pourquoi est-ce important ?

AWS IoTLes connexions CloudWatch fournissent une visibilité sur les comportementsAWS IoT, notamment les échecs d'authentification et les connexions et déconnexions inattendues qui peuvent indiquer qu'un appareil a été compromis.

Comment réparer

AWS IoTActivez les connexions CloudWatch. Consultez [Outils de surveillance \(p. 453\)](#). 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.
- Appliquez l'actionPUBLISH_FINDINGS_TO_SNS d'atténuation si vous souhaitez implémenter une réponse personnalisée en réponse au message Amazon SNS.

Pour plus d'informations, veuillez consulter [Actions d'atténuation \(p. 1144\)](#).

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	chaîne 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	chaîne	L'ARN de la cible (rubrique SNS) à laquelle des notifications d'audit sont envoyées.
roleArn	chaîne longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde l'autorisation d'envoyer des notifications à la cible.
enabled	booléen	La valeur est true si les notifications vers la cible sont activées.
auditCheckConfigurations	map	<p>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.</p> <p>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.</p> <p>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.</p> <p>Dans le premier appel à UpdateAccountAuditConfiguration,</p>

Nom	Type	Description
		ce paramètre est obligatoire et doit spécifier au moins un contrôle activé.
enabled	booléen	La valeur est true si cette vérification d'audit est activée pour ce compte.

Sortie

Aucun

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	chaîne 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 <code>UpdateAccountAuditConfiguration</code> , ce paramètre est requis.
auditNotificationTargetConfigurationsMap		Des informations sur les cibles auxquelles les notifications d'audit sont envoyées pour ce compte.
targetArn	chaîne	L'ARN de la cible (rubrique SNS) à laquelle des notifications d'audit sont envoyées.
roleArn	chaîne longueur - max. : 2048. Min. : 20	L'ARN du rôle qui accorde l'autorisation d'envoyer des notifications à la cible.
enabled	booléen	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	booléen	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	booléen	Si la valeur est true, tous les audits planifiés sont supprimés.

Sortie

Aucun

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	chaîne	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 : QUOTIDIEN HEBDOMADAIRE BIHEBDOMADAIRE MENSUEL
dayOfMonth	chaîne Modèle : ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	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	chaîne	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 ».

Nom	Type	Description
		enum : DIM LUN MAR MER JEU VEN SAM
targetCheckNames	list membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit planifié. Les contrôles doivent être activés sur votre compte. (Utilisez <code>DescribeAccountAuditConfiguration</code> pour afficher la liste de tous les contrôles, y compris ceux activés, ou <code>UpdateAccountAuditConfiguration</code> pour sélectionner les contrôles activés.)
tags	list membre : Tag classe Java : java.util.List	Métadonnées qui peuvent être utilisées pour gérer l'audit planifié.
Clé	chaîne	Clé de la balise.
Valeur	chaîne	Valeur de la balise.
scheduledAuditName	chaîne 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	chaîne	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.

LimitExceeded**Exception**

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	chaîne	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	list membre : ScheduledAuditMetadata	Liste des audits planifiés.

Nom	Type	Description
	classe Java : java.util.List	
scheduledAuditName	chaîne longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié.
scheduledAuditArn	chaîne	L'ARN de l'audit planifié.
frequency	chaîne	À quelle fréquence se déroule l'audit planifiée. enum : QUOTIDIEN HEBDOMADAIRE BIHEBDOMADAIRE MENSUEL
dayOfMonth	chaîne Modèle : ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	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 ne compte pas autant de jours, l'audit se déroule le « LAST » (dernier) jour du mois.
dayOfWeek	chaîne	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	chaîne	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	chaîne 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	chaîne	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 : QUOTIDIEN HEBDOMADAIRE BIHEBDOMADAIRE MENSUEL
dayOfMonth	chaîne Modèle : ^ ([1-9] [12] [0-9] [01]) \$ ^LAST\$	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.

Nom	Type	Description
dayOfWeek	chaîne	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	list membre : AuditCheckName	Quels contrôles sont effectués pendant l'audit planifié. Les contrôles doivent être activés sur votre compte. (Utilisez <code>DescribeAccountAuditConfiguration</code> pour afficher la liste de tous les contrôles, y compris ceux activés, ou <code>UpdateAccountAuditConfiguration</code> pour sélectionner les contrôles activés.)
scheduledAuditName	chaîne longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié.
scheduledAuditArn	chaîne	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	chaîne	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 : QUOTIDIEN HEBDOMADAIRE BIHEBDOMADAIRE MENSUEL
dayOfMonth	chaîne Modèle : ^ ([1-9] [12] [0-9] 3 [01]) \$ ^LAST\$	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	chaîne	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	list	Quels contrôles sont effectués pendant l'audit planifié. Les

Nom	Type	Description
	membre : AuditCheckName	contrôles doivent être activés sur votre compte. (Utilisez <code>DescribeAccountAuditConfiguration</code> pour afficher la liste de tous les contrôles, y compris ceux activés, ou <code>UpdateAccountAuditConfiguration</code> pour sélectionner les contrôles activés.)
scheduledAuditName	chaîne 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	chaîne	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	chaîne longueur - max. : 128. Min. : 1 modèle : [a-zA-Z0-9_-]+	Le nom de l'audit planifié que vous souhaitez supprimer.

Sortie

Aucun

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	list 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 <code>DescribeAccountAuditConfiguration</code> pour afficher la liste de tous les contrôles, y compris ceux activés, ou <code>UpdateAccountAuditConfiguration</code> pour sélectionner les contrôles activés.

Sortie

```
{
  "taskId": "string"
}
```

Champs de sortie de l'interface de ligne de commande

Nom	Type	Description
taskId	chaîne 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	chaîne 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	chaîne	Le statut de l'audit : « IN_PROGRESS », « COMPLETED », « FAILED » ou « CANCELED ». enum : IN_PROGRESS TERMINÉ ÉCHOUÉ ANNULÉ
taskType	chaîne	Type d'audit : ON_DEMAND_AUDIT_TASK ou SCHEDULED_AUDIT_TASK. enum : ON_DEMAND_AUDIT_TASK SCHEDULED_AUDIT_TASK
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	chaîne 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	chaîne	Le statut de finalisation de ce contrôle : « IN_PROGRESS », « WAITING_FOR_DATA_COLLECTION », « CANCELED », « COMPLETED_COMPLIANT »,

Nom	Type	Description
		« COMPLETED_NON_COMPLIANT » ou « FAILED ». enum : IN_PROGRESS WAITING_FOR_DATA_COLLECTION ANNULÉ COMPLÉTED_CONFORME COMPLETED_NON_COMPLIANT ÉCHEC
checkCompliant	booléen	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.
nonCompliantResourcesCount	long	Le nombre de ressources non conformes trouvées par le contrôle.
errorCode	chaîne	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	chaîne	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	chaîne	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	chaîne	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	list membre : AuditTaskMetadata classe Java : java.util.List	Audits exécutés au cours de la période spécifiée.
taskId	chaîne longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9-]+	ID de cet audit.
taskStatus	chaîne	Statut de l'audit : « IN_PROGRESS », « COMPLETED », « FAILED » ou « CANCELED ». enum : IN_PROGRESS TERMINÉ ÉCHOUÉ ANNULÉ
taskType	chaîne	Type de l'audit : « ON_DEMAND_AUDIT_TASK » ou « SCHEDULED_AUDIT_TASK ». enum : ON_DEMAND_AUDIT_TASK SCHEDULED_AUDIT_TASK
nextToken	chaîne	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	chaîne 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

Aucun

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 \(p. 1144\)](#).

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	chaîne 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	chaîne	Filtre pour limiter les conclusions au contrôle d'audit spécifié.
resourceIdentifier	ResourceIdentifier	Informations qui identifient les ressources non conformes.

Nom	Type	Description
deviceCertificateId	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.
cognitoIdentityPoolId	chaîne	ID du groupe d'identités Amazon Cognito.
clientId	chaîne	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.
policyName	chaîne longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	chaîne Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.
roleAliasArn	chaîne	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	chaîne	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	list	Conclusions (résultats) de l'audit.

Nom	Type	Description
	membre : AuditFinding	
taskId	chaîne longueur - max. : 40. Min. : 1 modèle : [a-zA-Z0-9]+	ID de l'audit qui a généré ce résultat.
checkName	chaîne	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	chaîne	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	chaîne	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	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.
cognitoIdentityPoolId	chaîne	ID du groupe d'identités Amazon Cognito.
clientId	chaîne	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.

Nom	Type	Description
policyName	chaîne longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	chaîne Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.
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	list membre : RelatedResource	Liste des ressources associées.
type de ressource	chaîne	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	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat attaché à la ressource.
caCertificateId	chaîne longueur - max. : 64. Min. : 64 modèle : (0x)?[a-fA-F0-9]+	ID du certificat CA utilisé pour autoriser le certificat.
cognitoIdentityPoolId	chaîne	ID du groupe d'identités Amazon Cognito.
clientId	chaîne	ID client.
policyVersionIdentifier	PolicyVersionIdentifier	Version de la stratégie associée à la ressource.
iamRoleArn	chaîne longueur - max. : 2048. Min. : 20	ARN du rôle IAM ayant des actions trop permissives.

Nom	Type	Description
policyName	chaîne longueur - max. : 128. Min. : 1 modèle : [w+=,.@-]+	Nom de la stratégie.
policyVersionId	chaîne Modèle : [0-9] +	ID de la version de la stratégie associée à la ressource.
roleAliasArn	chaîne 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	chaîne	Raison pour laquelle la ressource était non conforme.
reasonForNonComplianceCode	chaîne	Code qui indique la raison pour laquelle la ressource était non conforme.
nextToken	chaîne	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.

Vérification de la recherche de suppressions

Lorsque vous exécutez un audit, il signale les résultats de toutes les ressources non conformes. Cela signifie que vos rapports d'audit incluent des résultats pour les ressources dans lesquelles vous travaillez pour atténuer les problèmes, ainsi que pour les ressources connues comme non conformes, telles que les appareils de test ou les appareils cassés. L'audit continue de rapporter les résultats des ressources qui restent non conformes lors d'audits successifs, 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 corrigée, ou indéfiniment pour une ressource associée à un appareil de test ou à un périphérique endommagé.

Note

Les mesures d'atténuation ne seront pas disponibles pour les résultats de l'audit supprimés. Pour plus d'informations sur les actions d'atténuation, consultez [Actions d'atténuation \(p. 1144\)](#).

Pour plus d'informations sur les quotas de suppression de résultats d'audit, consultez [AWS IoT Points de terminaison et quotas Device Defender](#).

Comment fonctionne la recherche des suppressions d'audit

Lorsque vous créez une suppression de recherche d'audit pour une ressource non conforme, vos rapports d'audit et vos notifications se comportent différemment.

Vos rapports d'audit incluront une nouvelle section qui répertorie toutes les conclusions supprimées associées au rapport. Les résultats supprimés ne seront pas pris en compte lorsque nous évaluons si une vérification d'audit est conforme ou non. Un nombre de ressources supprimées est également renvoyé pour chaque vérification d'audit lorsque vous utilisez le [Décrire-Audit Task](#) dans l'interface de ligne de commande (CLI).

Pour les notifications d'audit, les résultats supprimés ne sont pas pris en compte lorsque nous évaluons si une vérification d'audit est conforme ou non. Un nombre de ressources supprimées est également inclus dans chaque notification de vérification d'audit AWS IoT Device Defender publie sur Amazon CloudWatch et Amazon Simple Notification Service (Amazon SNS).

Comment utiliser les suppressions de recherche d'audit dans la console

Pour supprimer un constat d'un rapport d'audit

La procédure suivante vous montre comment créer une suppression de résultats d'audit dans le fichier de AWS IoT console

1. Dans [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis choisissez Audit, Résultats.
2. Sélectionnez un rapport d'audit que vous souhaitez consulter.

Name	Date	Status	Summary
On-demand	July 28, 2020, 14:14:18 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
On-demand	July 28, 2020, 11:55:43 (UTC-0700)	🟢 Compliant	14 of 14 completed
AWSIoTDeviceDefenderDailyAudit	July 28, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 27, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 26, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 25, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 24, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 23, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 22, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant
AWSIoTDeviceDefenderDailyAudit	July 21, 2020, 05:12:39 (UTC-0700)	⚠️ Not compliant	1 of 14 non-compliant

3. Dans Contrôles non conformes, sous Nom de la vérification, choisissez la vérification d'audit qui vous intéresse.

The screenshot shows the AWS IoT Device Defender Audit Report interface. At the top, a breadcrumb navigation path is visible: AWS IoT > Device Defender > Audit > Audit Results > Audit Report. The main title is "Audit Report" with the subtitle "On-demand - July 28, 2020, 14:14:18 (UTC-0700)".

Audit findings:

- Audit task ID: 40c1204d7be8bb0d33682ef35c144231
- Started at: July 28, 2020, 14:14:18 (UTC-0700)

Non-compliant checks (1 of 14):

Check name	Severity	Non-compliant resources	% Resources	Mitigation
Logging disabled	Low	1	100%	Logging disabled ⓘ

Compliant checks (13 of 14):

Check name	Severity	Scanned ⓘ
Authenticated Cognito role overly permissive	Critical	0
CA certificate key quality	Critical	0
CA certificate revoked but device certificates still active	Critical	0
Device certificate key quality	Critical	0
Device certificate shared	Critical	0
IoT policies overly permissive	Critical	0
Role alias overly permissive	Critical	0
Unauthenticated Cognito role overly permissive	Critical	0
Conflicting MQTT client IDs	High	0
CA certificate expiring	Medium	0
Device certificate expiring	Medium	0
Revoked device certificate still active	Medium	0
Role alias allows access to unused services	Medium	0

- Sur l'écran des détails de la vérification de l'audit, s'il y a des constatations que vous ne souhaitez pas voir, cliquez sur le bouton d'option en regard de la recherche. Ensuite, choisissez Actions, puis choisissez la durée pendant laquelle vous souhaitez que la suppression des résultats de votre audit persiste.

Note

Dans la console, vous pouvez sélectionner 1 semaine, 1 mois, 3 mois, 6 mois, ou indéfinimentement tant que dates d'expiration pour la suppression des résultats de votre 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.

The screenshot shows the AWS IoT Device Defender Audit Findings page. The navigation path is: AWS IoT > Device Defender > Audit > Audit Results > Audit Report > Audit Findings. The page title is "Audit Findings" and it displays the message "Logging disabled". A section titled "1 account non-compliant" lists a single finding: "Non-compliant account (1)". The table row for this finding includes columns for "Finding" (containing the ID 417b2f816eac7a2e40fdb0bc709b01a2), "Reason" (containing "Logging disabled on account."), and "Account settings" (containing the ID 765219403047). To the right of the table is a "Actions" dropdown menu with options: "Start mitigation actions", "Suppress Finding", and a submenu with "1 week", "1 month", "3 months", "6 months", and "Indefinitely".

5. Confirmez les détails de suppression, puis choisissez **Activer la suppression**.

The screenshot shows a modal dialog box titled "Confirm suppression". The text inside the dialog reads: "Please verify the details of the audit finding suppression". Below this, the suppression details are listed: "Check name" (Logging disabled), "Account settings" (765219403047), "Expiration period" (3 months), and "Expiration date" (2020-10-28T21:25:41.100Z). At the bottom of the dialog are two buttons: "Cancel" and a large orange "Enable suppression" button.

6. Une fois que vous avez créé la suppression des résultats d'audit, une bannière s'affiche confirmant que la suppression de la recherche d'audit a été créée.

The screenshot shows a success message at the top: "Audit finding suppression created successfully" and "The finding related to the resource is suppressed for audit check Logging disabled". Below this, the navigation path is: AWS IoT > Device Defender > Audit > Audit Results > Audit Report > Audit Findings. The main section is titled "Audit Findings" and "Logging disabled". It displays a summary: "1 account non-compliant". Under "Mitigation", it says "Enable CloudWatch Logs.". Below this, a table titled "Non-compliant account (1)" lists one finding: "417b2f816eac7a2e40fdb0bc709b01a2" with the reason "Logging disabled on account." and the account ID "765219403047".

Pour afficher vos résultats supprimés dans un rapport d'audit

1. Dans [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis choisissez Audit, Résultats.
2. Sélectionnez un rapport d'audit que vous souhaitez consulter.
3. Dans Résultats supprimés, permet d'afficher les résultats d'audit qui ont été supprimés pour le rapport d'audit que vous avez choisi.

The screenshot shows the AWS IoT Device Defender Audit Report interface. The left sidebar navigation includes sections like Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend (with Intro selected), Audit (with Results, Schedules, Action executions, Finding suppressions selected), Detect (with Mitigation actions), Settings, Act (with Test), Software, Settings, Learn, and Documentation. The main content area displays an Audit Report for an on-demand audit on July 28, 2020, at 11:55:43 (UTC-0700). It lists 14 compliant checks (Authenticating Cognito role overly permissive, CA certificate key quality, CA certificate revoked but device certificates still active, Device certificate key quality, Device certificate shared, IoT policies overly permissive, Role alias overly permissive, Unauthenticated Cognito role overly permissive, Conflicting MQTT client IDs, CA certificate expiring, Device certificate expiring, Revoked device certificate still active, Role alias allows access to unused services, Logging disabled) and 1 suppressed finding (Logging disabled). A search bar at the bottom allows filtering by check name.

Check name	Finding	Reason	Resource identifier
Logging disabled	755a27914fb2ca24a8b3d47ef3563726	Logging disabled on account.	765219403047

Pour répertorier vos suppressions de recherche d'audit

- Dans [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis choisissez Audit, Trouver des suppressions.

The screenshot shows the AWS IoT Device Defender Audit Finding Suppressions interface. On the left, a navigation sidebar lists various AWS IoT services: Monitor, Activity, Onboard, Manage, Greengrass, Secure, Defend (with Intro), Audit (selected), Results, Schedules, Action executions, Finding suppressions (highlighted in orange), Detect, Mitigation actions (new), and Settings. The main content area displays a table titled "Audit finding suppressions (1) Info". The table has columns: Resource identifier, Check name, Expiration date, and Description. A single row is shown: Resource identifier 765219403047, Check name Logging disabled, Expiration date October 28, 2020, 14:26:53 (UTC-0700), and Description -. There are "Actions" and "Create" buttons at the top right of the table.

Pour modifier la suppression des résultats de votre audit

1. Dans [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis choisissez Audit, Trouver des suppressions.
2. Sélectionnez le bouton d'option en regard de la suppression de résultats d'audit que vous souhaitez modifier. Ensuite, choisissez Actions, Modifier.
3. Dans la page Modifier la suppression des résultats d'audit, vous pouvez modifier la fenêtre Durée de suppression ou Description (facultative).

Edit audit finding suppression

Supressing an audit finding on a specified resource means that the finding related to the resource for the specified audit check will no longer be flagged as non-compliant.

Audit check

Logging disabled

Resource identifier

Account ID

765219403047

Suppression duration

The expiration date is October 28, 2020, 14:26:53 (UTC-0700). Select a different duration to change this.

6 months

Description (optional)

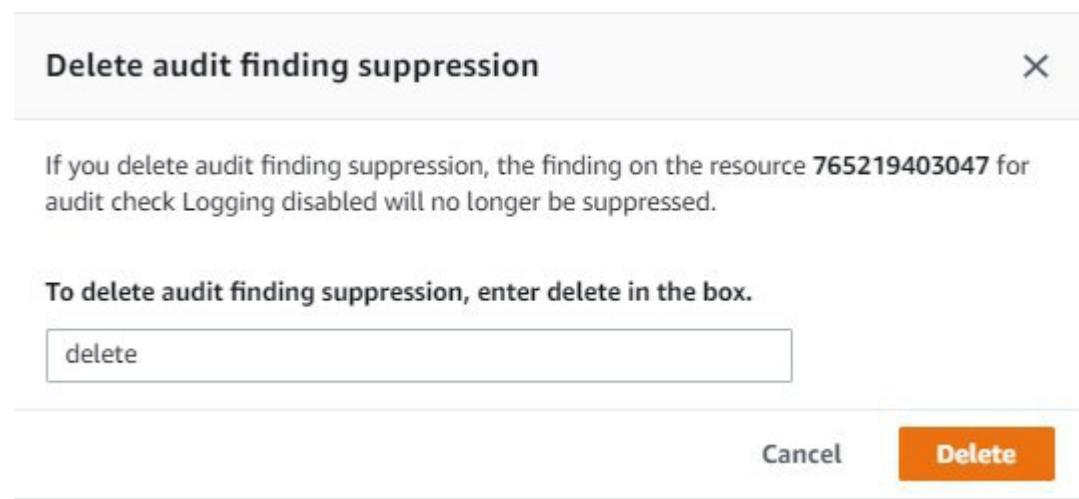
Suppresses "Logging disabled" check because I don't want to enable logging for now.

Cancel Save

4. Lorsque vous avez apporté vos modifications, choisissezEnregistrer. LeTrouver des suppressionsLa fenêtre s'ouvre.

Pour supprimer une suppression de recherche d'audit

1. DansAWS IoTconsole, dans le panneau de navigation, développezDéfendre, puis choisissezAudit,Trouver des suppressions.
2. Sélectionnez le bouton d'option en regard de la suppression de résultats d'audit à supprimer, puis choisissezActions,Supprimer.
3. Dans la pageSupprimer la suppression de la recherche d'auditfenêtre, saisissezdeletedans la zone de texte pour confirmer votre suppression, puis choisissezSupprimer. LeTrouver des suppressionsLa fenêtre s'ouvre.



Comment utiliser les suppressions de recherche d'audit dans l'interface de ligne de commande

Vous pouvez utiliser les commandes de ligne de commande suivantes pour créer et gérer des suppressions de résultats d'audit.

- [suppression de création-audit](#)
- [Décrire-Audit Suppression](#)
- [suppression de mise à jour et d'audit](#)
- [suppression/suppression de l'audit](#)
- [suppressions d'audit de liste](#)

Le `resource-identifier` votre entrée dépend de la `check-name` pour laquelle vous supprimez les résultats. Le tableau suivant détaille les vérifications qui exigent `resource-identifier` pour créer et modifier des suppressions.

Note

Les commandes de suppression n'indiquent pas la désactivation d'un audit. Les audits seront toujours exécutés sur votre AWS IoT appareils. Les suppressions ne s'appliquent qu'aux résultats de l'audit.

check-name	resource-identifier
AUTHENTICATE_COGNITO_ROLE_OVERLY_PERMISSIVE_CHECK	deviceIdentityPoolId
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

check-name	resource-identifier
DEVICE_CERTIFICATE_SHARED_CHECK	deviceCertificateId
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 des résultats d'audit

La procédure suivante vous montre comment créer une suppression de résultats d'audit dans le fichier deAWSCLI.

- Utilisation de l'`create-audit-suppression`pour créer une suppression de recherche d'audit. L'exemple suivant montre comment créer une suppression des résultats d'audit pourCompte AWS **123456789012**sur la base du contrôleLa journalisation est 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.

Vérification de la recherche de suppression des API

Les API suivantes peuvent être utilisées pour créer et gérer des suppressions de recherche d'audit.

- [Créer une suppression d'audit](#)
- [Décrire la suppression de l'audit](#)
- [Mise à jour de la suppression](#)
- [Supprimer la suppression de l'audit](#)
- [Liste des suppressions d'audit](#)

Pour filtrerpourdes résultats d'audit spécifiques, vous pouvez utiliser le[ListAuditFindingsAPI](#).

Détection

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 des comportements attendus des appareils, et vous les attribuez à un groupe d'appareils ou à tous les appareils de votre parc. AWS IoT Device Defender Detect utilise ces profils de sécurité pour détecter les anomalies et envoyer des alertes via CloudWatch Metrics Amazon 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. Une fois que AWS IoT Device Defender Detect vous a détecté la surface de l'attaque, vous pouvez la minimiser (en éliminant les services réseau inutilisés) ou exécuter des évaluations supplémentaires pour identifier les failles de sécurité (par exemple, telnet configuré avec des mots de passe courants, par défaut ou faibles).

Détecter les anomalies de comportement des appareils dont la cause possible est la sécurité

Vous pouvez utiliser AWS IoT Device Defender Detect pour vous alerter en cas de mesures comportementales inattendues de l'appareil (nombre de ports ouverts, nombre de connexions, port ouvert inattendu, connexions à des adresses IP inattendues) susceptibles d'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 rejettés 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 des

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 métriques 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 métrique.

Pour examiner une alarme d'anomalie, corrélez les détails de l'alarme avec d'autres informations contextuelles telles que les attributs de l'appareil, les tendances historiques des métriques de l'appareil, les tendances historiques des métriques du profil de sécurité, les mesures personnalisées et les journaux afin de déterminer la présence d'une menace de sécurité.

Cas d'utilisation côté cloud

Device Defender peut surveiller les cas d'utilisation suivants côté AWS IoT cloud.

Vol de propriété intellectuelle :

Le vol de propriété intellectuelle consiste à voler les propriétés intellectuelles d'une personne ou d'une entreprise, y compris des secrets commerciaux, du matériel ou des logiciels. Cela 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 dans le cloud peut se produire en raison de la présence de politiques qui autorisent un accès involontaire aux ressources de l'IoT. Vous devez revoir vos [politiques relatives à l'IoT](#) et activer l'option [Auditer les contrôles trop permissifs](#) afin d'identifier les politiques trop permissives.

Métriques associées :

Métrique	Justification
<u>IP Source</u>	Si l'appareil est volé, son adresse IP source se situera en dehors de la plage d'adresses IP normalement attendue pour les appareils circulant dans une chaîne d'approvisionnement normale.
<u>Nombre de messages reçus</u>	Étant donné qu'un attaquant peut utiliser un appareil pour voler des adresses IP dans le cloud, les statistiques relatives au nombre de messages ou à la taille des messages envoyés à l'appareil depuis le AWS IoT cloud peuvent augmenter, indiquant un éventuel problème de sécurité.
<u>Taille du message</u>	

Exfiltration de données basée sur le protocole MQTT :

L'exfiltration de données se produit lorsqu'un acteur malveillant effectue un transfert de données non autorisé depuis un déploiement IoT ou depuis un appareil. L'attaquant lance ce type d'attaques via MQTT contre des sources de données côté cloud.

Métriques associées :

Métrique	Justification
<u>IP Source</u>	En cas de vol d'un appareil, son adresse IP source se situera en dehors de la plage d'adresses IP normalement attendue pour les appareils circulant dans une chaîne d'approvisionnement standard.
<u>Nombre de messages reçus</u>	Étant donné qu'un attaquant peut utiliser un appareil dans le cadre d'une exfiltration de données basée sur le protocole MQTT, les indicateurs liés au nombre de messages ou à la taille des messages envoyés à l'appareil depuis le AWS IoT cloud peuvent augmenter, indiquant un éventuel problème de sécurité.
<u>Taille du message</u>	

Usurpation d'identité :

Dans le cadre d'une attaque par usurpation d'identité, les attaquants se font passer pour des entités connues ou de confiance dans le but d'accéder à des services, à des applications et à des données AWS IoT côté cloud ou de prendre le commandement et le contrôle d'appareils IoT.

Métriques associées :

Métrique	Justification
<u>Échecs d'autorisation</u>	
<u>Tentatives de connexion</u>	Lorsque des attaquants se font passer pour des entités de confiance en utilisant des identités volées, les indicateurs liés à la connectivité augmentent souvent, car les informations d'identification ne sont peut-être plus valides ou peuvent déjà être utilisées par un appareil fiable.
<u>Déconnexions</u>	

Métrique	Justification
	Les comportements anormaux liés aux échecs d'autorisation, aux tentatives de connexion ou aux déconnexions indiquent un scénario d'usurpation d'identité potentiel.

Abus de l'infrastructure cloud :

L'utilisation abusive des services AWS IoT cloud se produit lors de la publication ou de l'abonnement à des sujets comportant un volume de messages élevé ou des messages de grande taille. Des politiques trop permissives ou une exploitation de la vulnérabilité des appareils à des fins de commande et de contrôle peuvent également entraîner une utilisation abusive de l'infrastructure cloud. L'un des principaux objectifs de cette attaque est d'augmenter votre AWS facture. Vous devez revoir vos [politiques relatives à l'IoT](#) et activer l'option [Auditer les contrôles trop permis](#)s afin d'identifier les politiques trop permissives.

Métriques associées :

Métrique	Justification
Nombre de messages reçus	
Nombre de messages envoyés	
Taille du message	L'objectif de cette attaque est d'augmenter votre AWS facture. Les indicateurs qui surveillent des activités telles que le nombre de messages, les messages reçus et la taille des messages augmenteront.
IP Source	Des listes d'adresses IP sources suspectes peuvent apparaître, à partir desquelles les attaquants génèrent leur volume de messagerie.

Cas d'utilisation côté appareil

Device Defender peut surveiller les cas d'utilisation suivants du côté de votre appareil.

Denial-of-service Attaque D :

Une attaque denial-of-service (DoS) vise à arrêter un appareil ou un réseau, rendant ainsi l'appareil 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 ralentissent le système ou provoquent sa défaillance. Vos appareils IoT peuvent être utilisés dans le cadre d'attaques DoS.

Métriques associées :

Métrique	Justification
Paquets sortis	
Octets sortants	Les attaques DoS impliquent généralement des taux plus élevés de communications sortantes à partir d'un appareil donné et, selon le type d'attaque DoS, il peut y avoir une augmentation du nombre de paquets sortants et d'octets sortants, voire des deux.
IP de destination	Si vous définissez les adresses IP/plages d'adresses CIDR avec lesquelles vos appareils doivent communiquer, une anomalie dans l'adresse IP de destination peut indiquer une

Métrique	Justification
	communication IP non autorisée depuis vos appareils.
<u>Ports TCP d'écoute</u>	
<u>Nombre de ports TCP d'écoute</u>	
<u>Ports UDP d'écoute</u>	
<u>Nombre de ports UDP d'écoute</u>	Une attaque DoS nécessite généralement une infrastructure de commande et de contrôle plus importante dans laquelle les malwares installés sur vos appareils reçoivent des commandes et des informations indiquant qui attaquer et quand attaquer. Par conséquent, pour recevoir ces informations, le logiciel malveillant écoute généralement des ports qui ne sont pas normalement utilisés par vos appareils.

Intensification de la menace latérale :

L'escalade de la menace latérale commence généralement lorsqu'un attaquant accède à un point d'un réseau, par exemple un appareil connecté. L'attaquant tente ensuite d'augmenter son niveau de priviléges ou son accès à d'autres appareils en utilisant des méthodes telles que le vol d'informations d'identification ou des exploits de vulnérabilité.

Métriques associées :

Métrique	Justification
<u>Paquets sortis</u>	Dans des situations typiques, l'attaquant devrait exécuter une analyse sur le réseau local afin d'effectuer une reconnaissance et d'identifier les appareils disponibles afin d'affiner sa sélection de cibles d'attaque. Ce type d'analyse peut entraîner une augmentation du nombre d'octets et du nombre de paquets sortants.
<u>Octets sortants</u>	
<u>IP de destination</u>	Si un appareil est censé communiquer avec un ensemble connu d'adresses IP ou de CIDR, vous pouvez déterminer s'il tente de communiquer avec une adresse IP anormale, qui est souvent une adresse IP privée sur le réseau local en cas d'escalade de menace latérale.
<u>Échecs d'autorisation</u>	Lorsque l'attaquant tente d'augmenter son niveau de priviléges 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 du nombre d'échecs d'autorisation.

Exfiltration ou surveillance de données :

L'exfiltration de données se produit lorsqu'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 a normalement deux objectifs pour l'attaquant : obtenir des données ou de la propriété intellectuelle, ou effectuer la reconnaissance d'un réseau. La surveillance signifie qu'un 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 indicateurs ci-dessous peuvent fournir un point de départ pour enquêter sur l'un ou l'autre type d'attaque.

Métriques associées :

Métrique	Justification
<u>Paquets sortis</u>	Lorsque des attaques d'exfiltration de données ou de surveillance se produisent, l'attaquant reproduit souvent les données envoyées depuis l'appareil plutôt que de simplement les rediriger, qui sont identifiées par le défenseur lorsqu'il ne voit pas les données prévues arriver. Ces données en miroir augmenteraient considérablement la quantité totale de données envoyées depuis l'appareil, ce qui entraînerait une augmentation du nombre de paquets et d'octets sortants.
<u>Octets sortants</u>	
<u>IP de destination</u>	Lorsqu'un attaquant utilise un appareil dans le cadre d'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'adresse IP de destination peut aider à identifier une telle attaque.

Minage de crypto-monnaies

Les attaquants exploitent la puissance de traitement des appareils pour miner des cryptomonnaies. Le minage de cryptomonnaies est un processus intensif qui nécessite généralement une communication réseau avec d'autres pairs et pools de minage.

Métriques associées :

Métrique	Justification
<u>IP de destination</u>	La communication réseau est généralement requise lors du minage de cryptomonnaies. Le fait de disposer d'une liste étroitement contrôlée d'adresses IP avec lesquelles l'appareil doit communiquer peut aider à identifier les communications involontaires sur un appareil, comme le minage de cryptomonnaies.
<u>Métrique personnalisée</u> d'utilisation du processeur	Le minage de cryptomonnaies nécessite des calculs intensifs, ce qui entraîne une utilisation élevée du processeur de l'appareil. Si vous choisissez de collecter et de surveiller cette métrique, l'utilisation higher-than-normal du processeur peut être un indicateur des activités de minage de cryptomonnaies.

Commande et contrôle, logiciels malveillants et rançongiciels

Les malwares ou les rançongiciels limitent votre contrôle sur vos appareils et limitent leurs fonctionnalités. Dans le cas d'une attaque par ransomware, l'accès aux données serait perdu en raison du cryptage utilisé par le ransomware.

Métriques associées :

Métrique	Justification
IP de destination	Les attaques sur le réseau ou à distance représentent une grande partie des attaques contre les appareils IoT. Une liste étroitement contrôlée d'adresses IP avec lesquelles l'appareil doit communiquer peut aider à identifier les adresses IP de destination anormales résultant d'une attaque de logiciel malveillant ou de ransomware.
Ports TCP d'écoute	Plusieurs attaques de malwares impliquent le démarrage d'un command-and-control serveur qui envoie des commandes à exécuter sur un appareil. Ce type de serveur est essentiel à une opération de malware ou de 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 Defender Detect 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 par vous. Ces métriques peuvent provenir de deux sources : les métriques côté cloud et les métriques côté appareil. ML Detect prend en charge 6 métriques côté cloud et 7 métriques côté appareil. Pour obtenir une liste des métriques prises en charge pour ML Detect, consultez [Métriques prises en charge \(p. 1103\)](#).

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 Defender Detect peut également collecter, regrouper et surveiller les données de métriques générées par AWS IoT appareils (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. Consultez [Envoi de métriques à partir d'appareils \(p. 1125\)](#).

Profil de sécurité

Un profil de sécurité définit les comportements anormaux pour un groupe d'appareils ([un groupe d'objets \(p. 294\)](#)) ou tous les appareils de votre compte, et spécifie les mesures à prendre lorsqu'une anomalie est détectée. Vous pouvez utiliser le plugin AWS IoT Command Line Interface ou d'API pour 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 les anomalies dans le comportement des appareils.

comportement

Un comportement indique AWS IoT Device Defender Detect 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 un seuil de

valeur absolue ou 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 consiste en une mesure et une configuration ML Detect, qui définissent un modèle ML pour connaître le comportement normal des périphériques.

modèle ML

Un modèle ML est un modèle d'apprentissage automatique créé pour surveiller chaque comportement configuré par un client. Le modèle s'entraîne sur des modèles de données métriques provenant de groupes d'appareils ciblés et génère trois seuils de confiance en anomalies (élevé, moyen et faible) pour le comportement basé sur les mesures. Il déduit les anomalies basées sur 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 des mesures. Pour plus d'informations, consultez [Déetectez ML \(p. 1100\)](#).

niveau de fiabilité

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 et Low la confiance signifie une sensibilité élevée et un nombre souvent 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, consultez [CreateDimension](#).

alarme

Lorsqu'une anomalie est détectée, une notification d'alarme peut être envoyée via un CloudWatch métrique (voir [Utilisation de métriques AWS IoT \(p. 474\)](#)) ou une notification SNS. Une notification d'alarme est également affichée dans le AWS IoT ainsi que des informations sur l'alarme et un historique des alarmes de 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.

État de vérification d'alarme

Une fois qu'une alarme a été créée, vous pouvez vérifier que l'alarme est Vrai positive, Bénigne positive, Faux positive ou Inconnue. Vous pouvez également ajouter une description à l'état de vérification de votre alarme. Vous pouvez afficher, organiser et filtrer AWS IoT Device Defender alarms en utilisant l'un des quatre états de vérification. Vous pouvez utiliser les états de vérification des alarmes et les descriptions associées pour informer les membres de votre équipe. Cela aide votre équipe à prendre des mesures de suivi, par exemple en effectuant des actions d'atténuation sur des alarmes positives vraies, en ignorant les alarmes positives bénignes ou en poursuivant l'enquête sur les alarmes inconnues. L'état de vérification par défaut de toutes les alarmes est Inconnu.

suppression d'alarme

Gérer les notifications de détection d'alarme SNS en définissant la notification de comportement sur ou non suppressed. La suppression des alarmes n'empêche pas Détecter d'effectuer des évaluations du comportement des appareils ; Détecter continue de signaler les comportements anormaux comme des alarmes de violation. Toutefois, les alarmes supprimées ne seraient pas transmises pour notification SNS. Ils sont uniquement accessibles via le AWS IoT console 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 entrent dans deux catégories principales : Règles Détecte les comportements et ML Detect comportements. Avec les comportements Rules Detect, vous définissez le comportement de vos appareils, tandis 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 deux types de seuils : ML ou basée sur des règles. ML Security Profiles détecte automatiquement les anomalies opérationnelles et de sécurité au niveau de l'appareil dans votre flotte en apprenant des données antérieures. Les profils de sécurité basés sur des règles nécessitent 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 :

Commun à Rules Detect et ML Detect

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 deux types de seuils : Basé sur ML ou Rules. ML Security Profiles détecte automatiquement les anomalies opérationnelles et de sécurité au niveau de l'appareil dans votre flotte en apprenant des données antérieures. Les profils de sécurité basés sur des règles nécessitent 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 `notified`. La suppression des alarmes n'empêche pas Détecter d'effectuer des évaluations du comportement des appareils ; Détecter continue de signaler les comportements anormaux comme des alarmes de violation. Toutefois, les alarmes supprimées ne sont pas transmises pour notification SNS. Ils sont accessibles uniquement via la AWS IoT console ou API.

Règles Détecter

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, reportez-vous à la section [CreateDimension](#). 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), `cids` (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. Les percentiles sont ensuite dérivés 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 `statisticalThreshold`, 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.

Détecter ML

ML Detect confidence

ML Detect prend en charge trois niveaux de confiance :High,Medium, etLow.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 la confiance signifie une sensibilité élevée et un nombre souvent plus élevé d'alarmes.

Détectez ML

Avec Machine Learning Detect (ML Detect), vous créez des profils de sécurité qui utilisent l'apprentissage automatique pour connaître les comportements attendus des appareils en créant automatiquement des modèles basés sur les données historiques des appareils, et attribuez ces profils à un groupe d'appareils ou à tous les appareils de votre parc. AWS IoT Device Defender identifie ensuite les anomalies et déclenche des alarmes à l'aide des modèles ML.

Note

ML Detect prend désormais en charge la surveillance des indicateurs de santé opérationnelle propres à votre flotte. [Vous pouvez utiliser des mesures personnalisées \(p. 1105\) côté appareil et une surveillance plus précise de votre flotte grâce à la fonction de dimensions. \(p. 1133\)](#) En plus de régler manuellement des alarmes statiques avec Rules Detect, vous pouvez désormais utiliser l'apprentissage automatique pour connaître automatiquement les comportements attendus de votre flotte sur la base de métriques personnalisées. Vous pouvez également filtrer les métriques côté cloud en fonction des dimensions.

Pour plus d'informations sur la façon de commencer à utiliser ML Detect, consultez[Guide de ML \(p. 989\).](#)

Ce chapitre contient les sections suivantes :

- [Cas d'utilisation de ML Detect \(p. 1101\)](#)
- [Comment fonctionne ML Detect \(p. 1101\)](#)
- [Configuration requise \(p. 1102\)](#)
- [Limites \(p. 1102\)](#)
- [Marquage des faux positifs et autres états de vérification dans les alarmes \(p. 1103\)](#)
- [Métriques prises en charge \(p. 1103\)](#)
- [Service Quotas \(p. 1103\)](#)
- [Commandes CLI de ML Detect \(p. 1103\)](#)
- [API de ML Detect ML \(p. 1104\)](#)
- [Suspendre ou supprimer un profil de sécurité ML Detect \(p. 1104\)](#)

Cas d'utilisation de ML Detect

Vous pouvez utiliser ML Detect pour surveiller les appareils de votre parc lorsqu'il est difficile de définir les comportements attendus des appareils. Par exemple, pour surveiller la métrique du nombre de déconnexions, il se peut que le seuil considéré comme acceptable ne soit pas clair. Dans ce cas, vous pouvez activer ML Detect pour identifier les points de données métriques de déconnexion anormaux sur la base des données historiques signalées par les appareils.

Un autre cas d'utilisation de ML Detect consiste à surveiller les comportements des appareils qui évoluent de manière dynamique au fil du temps. ML Detect apprend régulièrement les comportements dynamiques attendus des appareils en fonction de l'évolution des modèles de données des appareils. Par exemple, le volume des messages envoyés par l'appareil peut varier entre les jours de la semaine et le week-end, et ML Detect apprendra ce comportement dynamique.

Comment fonctionne ML Detect

À l'aide de ML Detect, vous pouvez créer des comportements pour identifier les anomalies opérationnelles et de sécurité sur la base de [6 indicateurs côté cloud \(p. 1103\)](#) et de [7 indicateurs côté appareil. \(p. 1103\)](#) Après la période d'apprentissage initiale du modèle, ML Detect actualise les modèles quotidiennement en fonction des 14 derniers jours de données. Il surveille les points de données pour ces métriques à l'aide des modèles ML et déclenche une alarme si une anomalie est détectée.

ML Detect fonctionne mieux si vous associez un profil de sécurité à un ensemble d'appareils présentant des comportements attendus similaires. Par exemple, si certains de vos appareils sont utilisés au domicile des clients et d'autres dans les bureaux, les modèles de comportement des appareils peuvent être très différents entre les deux groupes. Vous pouvez organiser les appareils en un groupe d'objets pour les appareils domestiques et un groupe d'objets pour les appareils de bureau. Pour une détection des anomalies optimale, associez chaque groupe d'objets à un profil de sécurité ML Detect distinct.

ML Detect construit le modèle initial, mais il lui faut 14 jours et un minimum de 25 000 points de données par métrique au cours des 14 derniers jours pour générer un modèle. Ensuite, il met à jour le modèle chaque jour en fonction d'un nombre minimum de points de données métriques. Si l'exigence minimale n'est pas atteinte, ML Detect tente de créer le modèle le jour suivant et réessaiera quotidiennement pendant les 30 prochains jours avant d'arrêter le modèle pour des évaluations.

Configuration requise

Pour la formation et la création du modèle ML initial, ML Detect répond aux exigences minimales suivantes.

Période minimale de formation

Il faut 14 jours pour construire les premiers modèles. Ensuite, le modèle est actualisé tous les jours avec les données métriques d'une période de 14 jours consécutifs.

Points de données minimaux

Le minimum de points de données requis pour créer un modèle ML est de 25 000 points de données par métrique au cours des 14 derniers jours. Pour la formation continue et l'actualisation du modèle, ML Detect exige que le minimum de points de données soit atteint pour les appareils surveillés. C'est à peu près l'équivalent des configurations suivantes :

- 60 appareils connectés et actifs toutes AWS IoT les 45 minutes.
- 40 appareils à intervalles de 30 minutes.
- 15 appareils à intervalles de 10 minutes.
- 7 appareils à 5 minutes d'intervalle.

Cibles des groupes d'appareils

Pour collecter des données, vous devez disposer d'éléments dans les groupes d'objets cibles du 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 14 jours.

Limites

Vous pouvez utiliser ML Detect avec des dimensions relatives aux métriques cloud suivantes :

- [Échecs d'autorisation \(aws:num-authorization-failures\) \(p. 1129\)](#)
- [Messages reçus \(aws:num-messages-reçues\) \(p. 1128\)](#)
- [Messages envoyés \(aws:num-messages-sent\) \(p. 1127\)](#)
- [Taille du message \(aws:taille d'octet de message\) \(p. 1126\)](#)

Les mesures suivantes ne sont pas prises en charge par ML Detect.

Les métriques côté cloud ne sont pas prises en charge par ML Detect :

- [IP source \(aws:adresse IP source\) \(p. 1130\)](#)

Les métriques côté appareil ne sont pas prises en charge par ML Detect :

- [IP de destination \(aws:destination-ip-addresses\) \(p. 1117\)](#)
- [Ports TCP d'écoute \(aws:listening-tcp-ports\) \(p. 1118\)](#)
- [Ports UDP d'écoute \(aws:listening-udp-ports\) \(p. 1118\)](#)

Les métriques personnalisées ne prennent en charge que le type de nombre.

Marquage des faux positifs et autres états de vérification dans les alarmes

Si vous vérifiez qu'une alarme ML Detect est un faux positif au cours de votre enquête, vous pouvez définir l'état de vérification de l'alarme sur Faux positif. Cela peut vous aider, vous et votre équipe, à identifier les alarmes auxquelles vous n'avez pas à répondre. Vous pouvez également marquer les alarmes comme étant Vrai positif, BÉNIN positif ou Inconnu.

Vous pouvez marquer les alarmes via la [AWS IoT Device Defenderconsole](#) ou à l'aide de l'action de [l'PutVerificationStateOnViolationAPI](#).

Métriques prises en charge

Vous pouvez utiliser les métriques suivantes de l'interface de ligne de commande avec ML Detect :

- [Échecs d'autorisation \(aws:num-authorization-failures\) \(p. 1129\)](#)
- [Tentatives de connexion \(aws:num-connection-tentatives\) \(p. 1131\)](#)
- [Se déconnecte \(aws:num-déconnecte\) \(p. 1132\)](#)
- [Taille du message \(aws:taille d'octet de message\) \(p. 1126\)](#)
- [Messages envoyés \(aws:num-messages-sent\) \(p. 1127\)](#)
- [Messages reçus \(aws:num-messages-reçues\) \(p. 1128\)](#)

Vous pouvez utiliser les métriques suivantes de l'appareil avec ML Detect :

- [Octets sortants \(aws:all-bytes-out\) \(p. 1111\)](#)
- [Octets en \(aws:all-bytes-in\) \(p. 1112\)](#)
- [Nombre de ports TCP d'écoute \(aws:num-listening-tcp-ports\) \(p. 1113\)](#)
- [Nombre de ports UDP d'écoute \(aws:num-listening-udp-ports\) \(p. 1114\)](#)
- [Paquets sortants \(aws:all-packets-out\) \(p. 1115\)](#)
- [Paquets dans \(aws:all-packets-in\) \(p. 1116\)](#)
- [Nombre de connexions TCP établies \(aws:num-established-tcp-connections\) \(p. 1119\)](#)

Service Quotas

Pour plus d'informations sur les quotas et les limites du service ML Detect, consultez la section [AWS IoT Device DefenderPoints de terminaison et quotas](#).

Commandes CLI de ML Detect

Vous pouvez utiliser les commandes suivantes de l'interface de ligne de commande pour créer et gérer ML Detect.

- [create-security-profile](#)
- [attach-security-profile](#)
- [list-security-profiles](#)
- [describe-security-profile](#)
- [update-security-profile](#)

- [delete-security-profile](#)
- [get-behavior-model-training-résumés](#)
- [list-active-violations](#)
- [list-violation-events](#)

API de ML Detect ML

Les API suivantes peuvent être utilisées pour créer et gérer les profils de sécurité ML Detect.

- [CreateSecurityProfile](#)
- [AttachSecurityProfile](#)
- [ListSecurityProfiles](#)
- [DescribeSecurityProfile](#)
- [UpdateSecurityProfile](#)
- [DeleteSecurityProfile](#)
- [GetBehaviorModelTrainingSummaries](#)
- [ListActiveViolations](#)
- [ListViolationEvents](#)
- [PutVerificationStateOnViolation](#)

Suspendre ou supprimer un profil de sécurité ML Detect

Vous pouvez suspendre votre profil de sécurité ML Detect pour arrêter temporairement de surveiller les comportements des appareils, ou supprimer votre profil de sécurité ML Detect pour arrêter de surveiller les comportements des appareils pendant une période prolongée.

Mettre en pause le profil de sécurité ML Detect à l'aide de la console

Pour suspendre un profil de sécurité ML Detect à l'aide de la console, vous devez d'abord disposer d'un groupe d'objets vide. Pour créer un groupe d'objets vide, reportez-vous à la section [Groupes d'objets statiques \(p. 294\)](#). Si vous avez créé un groupe d'objets vide, définissez-le comme cible du profil de sécurité ML Detect.

Note

Vous devez redéfinir la cible de votre profil de sécurité sur un groupe d'appareils dans un délai de 30 jours, faute de quoi vous ne pourrez pas réactiver le profil de sécurité.

Supprimer le profil de sécurité ML Detect à l'aide de la console

Pour supprimer un profil de sécurité, suivez ces étapes :

1. Dans la AWS IoT console, accédez à la barre latérale et choisissez la section Défendre.
2. Sous Défendre, choisissez Détecter, puis Profils de sécurité.
3. Choisissez le profil de sécurité de ML Detect que vous voulez supprimer.
4. Choisissez Actions, puis parmi les options, choisissez Supprimer.

Note

Une fois qu'un profil de sécurité ML Detect est supprimé, vous ne pouvez pas le réactiver.

Mettre en pause un profil de sécurité ML Detect à l'aide de l'interface de ligne de commande

Pour suspendre un profil de sécurité ML Detect à l'aide de l'interface de ligne de commande, utilisez la `detach-security-profile` commande suivante :

```
$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 n'est disponible que dans l'AWSinterface de ligne de commande. Comme pour le workflow de la console, vous devez redéfinir la cible de votre profil de sécurité sur un groupe d'appareils dans un délai de 30 jours, faute de quoi vous ne pourrez pas réactiver le profil de sécurité. Pour associer un profil de sécurité à un groupe d'appareils, utilisez la [attach-security-profile](#) 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 la `delete-security-profile` commande ci-dessous :

```
delete-security-profile --security-profile-name SecurityProfileName
```

Note

Une fois qu'un profil de sécurité ML Detect est supprimé, vous ne pouvez pas le réactiver.

Métriques personnalisées

avec AWS IoT Device Defender des indicateurs personnalisés, vous pouvez définir et surveiller des indicateurs spécifiques à votre parc ou à votre cas d'utilisation, tels 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 mesure personnalisés sont définis dans les profils de sécurité, qui spécifient les comportements attendus pour un groupe d'appareils (un groupe d'objets) ou pour 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 une métrique personnalisée dans la console \(p. 1105\)](#)
- [Comment utiliser les métriques personnalisées depuis l'interface de ligne de commande \(p. 1107\)](#)
- [Métriques personnalisées, commandes CLI \(p. 1110\)](#)
- [API de métriques personnalisées \(p. 1110\)](#)

Comment utiliser une métrique personnalisée dans la console

Didacticiels

- [AWS IoT Device Defender SDK de l'agent \(Python\) \(p. 1106\)](#)
- [Créez une métrique personnalisée et ajoutez-la à un profil de sécurité \(p. 1106\)](#)
- [Afficher les détails des mesures personnalisées \(p. 1106\)](#)
- [Mettre à jour une métrique personnalisée \(p. 1107\)](#)

- [Supprimer une métrique personnalisée \(p. 1107\)](#)

AWS IoT Device DefenderSDK de l'agent (Python)

Pour commencer, téléchargez AWS IoT Device Defender Exemple d'agent du SDK de l'agent (Python). L'agent collecte les indicateurs et publie des rapports. Une fois que les indicateurs côté appareil sont publiés, vous pouvez consulter les indicateurs collectés et déterminer les seuils de configuration des alarmes. Les instructions relatives à la configuration de l'agent de l'appareil sont disponibles sur le [AWS IoT Fichier Lisez-moi du SDK Device Defender Agent \(Python\)](#). Pour plus d'informations, veuillez consulter la rubrique [AWS IoT Device DefenderSDK de l'agent \(Python\)](#).

Créez une métrique personnalisée et ajoutez-la à un profil de sécurité

La procédure suivante vous montre comment créer une métrique personnalisée dans la console.

1. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Détect, Métriques.
2. Sur la page Métriques personnalisées, choisissez Création.
3. Sur la page Créer une métrique personnalisée, procédez comme suit.
 1. Sous Nom, entrez le nom de votre métrique personnalisée. Vous ne pouvez pas modifier ce nom après avoir créé la métrique personnalisée.
 2. Sous Nom d'affichage (facultatif), vous pouvez saisir un nom convivial pour votre métrique personnalisée. Ils n'ont pas besoin d'être uniques et peuvent être modifiés après création.
 3. Sous Type, choisissez le type de métrique que vous souhaitez surveiller. Les types métriques incluent liste de chaînes, ip-address-list, liste de numéros, et nombre. Le type ne peut pas être modifié après sa création.

Note

ML Detect autorise uniquement nombre type.

4. Sous Étiquettes, vous pouvez sélectionner les balises à associer à la ressource.

Lorsque vous avez terminé, choisissez Confirmer.

4. Une fois que vous avez créé votre métrique personnalisée, Métriques personnalisées une page apparaît, où vous pouvez voir votre nouvelle métrique personnalisée.
5. Vous devez ensuite ajouter votre métrique personnalisée à un profil de sécurité. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Détect, Profils de sécurité.
6. Choisissez le profil de sécurité auquel vous souhaitez ajouter votre métrique personnalisée.
7. Choisissez Actions, Edit (Modifier).
8. Choisissez Indicateurs supplémentaires à conserver, puis choisissez votre métrique personnalisée. Choisissez Suivant sur les écrans suivants jusqu'à ce que vous atteignez Confirmer page. Choisissez Save et Continuer. Une fois que votre métrique personnalisée a été ajoutée avec succès, la page de 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.

Afficher les détails des mesures personnalisées

La procédure suivante vous montre comment afficher les détails d'une métrique personnalisée dans la console.

1. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Detec, Métriques.
2. Choisissez le Nom de métrique de la métrique personnalisée dont vous souhaitez consulter les détails.

Mettre à jour une métrique personnalisée

La procédure suivante vous montre comment mettre à jour une métrique personnalisée dans la console.

1. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Detec, Métriques.
2. Choisissez le bouton d'option en regard de la métrique personnalisée à mettre à jour. Ensuite, pour Actions, choisissez Modifier.
3. Sur la page Mettre à jour une métrique, vous pouvez modifier le nom d'affichage et supprimer ou ajouter des balises.
4. Lorsque vous avez terminé, sélectionnez Mise à jour. Dans la barre de navigation, cliquez sur Défendre, puis Detec, Métriques.

Supprimer une métrique personnalisée

La procédure suivante vous montre comment supprimer une métrique personnalisée dans la console.

1. Tout d'abord, supprimez votre métrique personnalisée de tout profil de sécurité dans lequel elle est référencée. Vous pouvez voir quels profils de sécurité contiennent votre métrique personnalisée sur la page de détails de votre métrique personnalisée. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Detec, Métriques.
2. Choisissez la métrique personnalisée que vous souhaitez supprimer. Supprimez la métrique personnalisée de tout profil de sécurité répertorié sous Profils de sécurité sur la page de détails des métriques personnalisées.
3. Dans le [AWS IoT console](#), dans le panneau de navigation, développez Défendre, puis Detec, Métriques.
4. Choisissez le bouton d'option en regard de la métrique personnalisée à supprimer. Ensuite, pour Actions, choisissez Supprimer.
5. Sur la page Etes-vous sûr de vouloir supprimer une métrique personnalisée ? Message, choisissez Supprimer une métrique personnalisée.

Warning

Une fois que vous avez supprimé une métrique personnalisée, vous perdez toutes les données associées à cette métrique. Cette action ne peut pas être annulée.

Comment utiliser les métriques personnalisées depuis l'interface de ligne de commande

Didacticiels

- [AWS IoT Device Defender SDK de l'agent \(Python\) \(p. 1107\)](#)
- [Créer une métrique personnalisée et ajoutez-la à un profil de sécurité \(p. 1108\)](#)
- [Afficher les détails des mesures personnalisées \(p. 1109\)](#)
- [Mettre à jour une métrique personnalisée \(p. 1109\)](#)
- [Supprimer une métrique personnalisée \(p. 1109\)](#)

AWS IoT Device Defender SDK de l'agent (Python)

Pour commencer, téléchargez [AWS IoT Device Defender Exemple d'agent du SDK de l'agent \(Python\)](#). L'agent collecte les indicateurs et publie des rapports. Une fois que les mesures côté appareil ont été

publiées, vous pouvez consulter les mesures collectées et déterminer les seuils de configuration des alarmes. Les instructions relatives à la configuration de l'agent de l'appareil sont disponibles sur le [AWS IoT Fichier Lisez-moi du SDK Device Defender Agent \(Python\)](#). Pour plus d'informations, veuillez consulter la rubrique [AWS IoT Device Defender SDK de l'agent \(Python\)](#).

Créez une métrique 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. Utiliser la [create-custom-metric](#) commande pour créer votre métrique personnalisée. L'exemple suivant crée une métrique 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" \
```

Sortie :

```
{
  "metricName": "batteryPercentage",
  "metricArn": "arn:aws:iot:us-east-1:1234564789012:custommetric/batteryPercentage"
}
```

2. Après avoir créé votre métrique personnalisée, vous pouvez l'ajouter à un profil existant en utilisant [update-security-profile](#) ou créez un nouveau profil de sécurité pour ajouter la métrique personnalisée à utiliser [create-security-profile](#). Ici, nous créons un nouveau profil de sécurité appelé **Utilisation de la batterie** pour ajouter notre nouveau **Pourcentage de batterie** métrique personnalisée de. Nous ajoutons également une métrique de détection des règles appelée **Bande passante cellulaire**.

```
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
```

Sortie :

```
{
  "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.

Afficher les détails des mesures personnalisées

La procédure suivante vous montre comment afficher les détails d'une métrique personnalisée à partir de l'interface de ligne de commande.

- Utiliser le [list-custom-metrics](#) commande pour afficher toutes vos métriques personnalisées.

```
aws iot list-custom-metrics \
--region us-east-1
```

La sortie de cette commande ressemble à ce qui suit.

```
{
  "metricNames": [
    "batteryPercentage"
  ]
}
```

Mettre à jour une métrique personnalisée

La procédure suivante vous explique comment mettre à jour une métrique personnalisée à partir de l'interface de ligne de commande.

- Utiliser le [update-custom-metric](#) commande pour mettre à jour une métrique personnalisée. L'exemple suivant met à jour `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 métrique personnalisée de l'interface de ligne de commande.

- Pour supprimer une métrique personnalisée, commencez par la supprimer de tous les profils de sécurité auxquels elle est associée. Utiliser le [list-security-profiles](#) commande pour afficher les profils de sécurité avec une certaine métrique personnalisée.
- Pour supprimer une métrique personnalisée d'un profil de sécurité, utilisez [update-security-profiles](#) commande. Entrez toutes les informations que vous souhaitez conserver, mais excluez la métrique 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 métrique personnalisée détachée, utilisez [delete-custom-metric](#) pour supprimer la métrique 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
```

Métriques personnalisées, commandes CLI

Vous pouvez utiliser les commandes suivantes de l'interface de ligne de commande pour créer et gérer une métrique personnalisée.

- [create-custom-metric](#)
- [describe-custom-metric](#)
- [list-custom-metrics](#)
- [update-custom-metric](#)
- [delete-custom-metric](#)
- [list-security-profiles](#)

API de métriques personnalisées

Les API suivantes peuvent être utilisées pour créer et gérer des métriques personnalisées.

- [CreateCustomMetric](#)
- [DescribeCustomMetric](#)
- [ListCustomMetrics](#)
- [UpdateCustomMetric](#)
- [DeleteCustomMetric](#)

- [ListSecurityProfiles](#)

Métriques côté appareil

Lorsque vous créez un profil de sécurité, vous pouvez spécifier le comportement attendu de votre appareil IoT en configurant des comportements et des seuils pour les mesures générées par les appareils IoT. Les mesures suivantes concernent les appareils, qui proviennent des agents que vous installez sur vos appareils.

Octets sortants (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 maximale ou minimale de trafic sortant qu'un appareil doit envoyer, mesurée en octets, sur une période donnée.

Compatible avec : Rules Detect | ML Detect

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",  
    "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-equals",  
    "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 en (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 maximale ou minimale de trafic entrant qu'un appareil doit recevoir, mesurée en octets, au cours d'une période donnée.

Compatible avec : Rules Detect | ML Detect

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",  
        "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-equals",  
        "statisticalThreshold": {  
            "statistic": "p90"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1  
    },  
    "suppressAlerts": true  
}
```

```
        "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)

Nombre de ports TCP sur lesquels l'appareil écoute.

Utilisez cette métrique pour spécifier le nombre maximum de ports TCP que chaque périphérique doit surveiller.

Compatible avec : Rules Detect | ML Detect

Unités : Échecs

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",
    "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-equals",  
        "statisticalThreshold": {  
            "statistic": "p50"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1  
    },  
    "suppressAlerts": true  
}
```

Example Exemple utilisant ML Detect

```
{  
    "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)

Nombre de ports UDP sur lesquels le périphérique écoute.

Utilisez cette métrique pour spécifier le nombre maximum de ports UDP que chaque périphérique doit surveiller.

Compatible avec : Rules Detect | ML Detect

Unités : Échecs

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",  
        "value": {  
            "count": 5  
        },  
        "durationSeconds": 300,  
    },  
    "suppressAlerts": true  
}
```

```
        "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-equals",
        "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 sortants (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 : Rules Detect | ML Detect

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",  
    "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-equals",  
        "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 dans (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 : Rule Detect | ML Detect

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

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-equals",  
    "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-equals",  
    "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 routages interdomaines (CIDR) autorisés (anciennement appelés « liste blanche ») ou refusés (anciennement appelés « liste noire ») à partir desquels chaque appareil doit ou ne doit pas se connecterAWS IoT.

Compatible avec : Rules Detect

Opérateurs : in-cidr-set | not-in-cidr-set

Valeurs : une liste de CIDR

Unités : N/A

Example

```
{  
    "name": "Denied source IPs",  
    "metric": "aws:destination-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)

Le port TCP sur lequel écoute le périphérique.

Utilisez cette métrique pour spécifier un ensemble de ports TCP autorisés (anciennement appelés « liste blanche ») ou refusés (anciennement appelés « liste noire ») sur lesquels chaque appareil doit ou ne doit pas écouter.

Compatible avec : Rules Detect

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 sur lesquels écoute le périphérique.

Utilisez cette métrique pour spécifier un ensemble de ports UDP autorisés (anciennement appelés « liste blanche ») ou refusés (anciennement appelés « liste noire ») sur lesquels chaque appareil doit ou ne doit pas écouter.

Compatible avec : Rules Detect

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 que chaque appareil doit avoir (tous les états TCP).

Compatible avec : Rules Detect | ML Detect

Opérateurs : inférieur à less-than-equals || supérieur à greater-than-equals

Valeur : Nombre entier non négatif.

Unités : Connexions

Example

```
{  
    "name": "TCP Connection Count",  
    "metric": "aws:num-established-tcp-connections",  
    "criteria": {  
        "comparisonOperator": "less-than-equals",  
        "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-equals",  
        "statisticalThreshold": {  
            "count": 3  
        }  
    }  
}
```

```

        "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	Y	Objet		Bloc complet requis pour rapport correct
metrics	met	Y	Objet		Un rapport peut contenir les deux ou au moins unmetrics ou plusieurscustom_metrics blocs.
métriques personnalisées	cmet	Y	Objet		Un rapport peut contenir les deux ou au moins unmetrics ou plusieurscustom_metrics blocs.

Bloc d'en-tête

Nom long	Nom court	Obligatoire	Type	Contraintes	Remarques
report_id	rid	Y	Entier		Valeur augmentant de façon monotone.

Nom long	Nom court	Obligatoire	Type	Contraintes	Remarques
					Horodatage epoch conseillé.
version ;	v	Y	Chaîne	Major.Minor	Incréments mineurs avec ajout de champ. Incréments majeurs si métriques supprimées.

Bloc de mesures :

Connexions TCP

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
tcp_connection\$ c	metrics		N	Objet		
established_connections		tcp_connection\$ N		Objet		État TCP établie
connexions	cs	established_connections		List<Object>		
remote_addr	rad	connexions	Y	Nombre	ip:port	IP peut être IPv6 ou IPv4
local_port	lp	connexions	N	Nombre	>= 0	
local_interface	li	connexions	N	Chaîne		Nom d'interface
total	t	established_connections		Nombre	>= 0	Nombre de connexions établies

Ports TCP d'écoute

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
listening_tcp_ports	pts	metrics	N	Objet		
ports	pts	listening_tcp_ports		List<Object>	> 0	
port	pt	ports	N	Nombre	> 0	les ports doivent être des nombres supérieurs à 0
interface	if	ports	N	Chaîne		Nom d'interface

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
total	t	listening_tcp_ports		Nombre	>= 0	

Ports UDP d'écoute

Nom long	Nom court	Élément parent	Obligatoire	Type	Contraintes	Remarques
listening_udp_ports	pts	metrics	N	Objet		
ports	pts	listening_udp_ports		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		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": [
        {
          "if": "eth0",
          "port": 80
        }
      ],
      "total": 1
    }
  }
}
```

```
        "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": {
      "t": 1
    }
}
```

```

    "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 la version 2 du AWS IoT SDK sur vos appareils AWS IoT connectés ou vos passerelles d'appareils afin de collecter des mesures côté appareil. Consultez la liste complète des SDK [ici](#).

Vous pouvez utiliser AWS IoT Device Client pour publier des métriques, car il fournit un agent unique qui couvre les fonctionnalités présentes à la fois dans Device Management AWS IoT Device Defender

et dans AWS IoT Device Management. Ces fonctionnalités incluent les tâches, le tunneling sécurisé, la publication de AWS IoT Device Defender statistiques, etc.

Vous publiez les métriques côté appareil dans la [rubrique réservée](#) AWS IoT Device Defender à AWS IoT des fins de collecte et d'évaluation.

Utilisation du AWS IoT Device Client pour publier des métriques

Pour installer AWS IoT Device Client, vous pouvez le télécharger depuis [Github](#). Après avoir installé le AWS IoT Device Client sur l'appareil pour lequel vous souhaitez collecter des données côté appareil, vous devez le configurer pour envoyer des métriques côté appareil AWS IoT Device Defender. Vérifiez que les paramètres suivants sont définis dans la `device-defender` section du [fichier de configuration AWS IoT Device Client](#) :

```
"device-defender": {  
    "enabled": true,  
    "interval-in-seconds": 300  
}
```

Warning

Vous devez régler l'intervalle de temps sur un minimum de 300 secondes. Si vous définissez l'intervalle de temps sur une valeur inférieure à 300 secondes, vos données métriques risquent d'être limitées.

Après avoir mis à jour votre configuration, vous pouvez créer des profils et des comportements de sécurité dans la AWS IoT Device Defender console afin de surveiller les mesures que vos appareils publient dans le cloud. Vous pouvez trouver les métriques publiées dans la AWS IoT Core console en choisissant Defend, Detect, puis Metrics.

Mesures côté cloud

Lorsque vous créez 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 cloud, qui sont des métriques de l'AWS IoT.

Taille du message (aws:taille d'octet de 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-equals | greater-than | greater-than-equals

Valeur : Nombre entier non négatif.

Unité : Octets

Example

```
{  
    "name": "Max Message Size",  
    "metric": "aws:message-byte-size",  
    "criteria": {  
        "comparisonOperator": "less-than-equals",  
        "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-equals",
        "statisticalThreshold": {
            "statistic": "p90"
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 1
    },
    "suppressAlerts": true
}
```

Example Exemple d'utilisation de 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-sent)

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 entre AWS IoT et chaque appareil au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than-equals | greater-than | greater-than-equals

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-equals",  
        "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-equals",  
        "statisticalThreshold": {  
            "statistic": "p99"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1  
    },  
    "suppressAlerts": true  
}
```

Example Exemple d'utilisation de 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-reçues)

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 entre AWS IoT et chaque appareil au cours d'une période donnée.

Compatible avec : Détection des règles | ML Detect

Opérateurs : less-than | less-than-equals | greater-than | greater-than-equals

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-equals",
    "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-equals",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de 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-failures)

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-equals | greater-than | greater-than-equals

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-equals",  
    "statisticalThreshold": {  
      "statistic": "p50"  
    },  
    "durationSeconds": 300,  
    "consecutiveDatapointsToAlarm": 1,  
    "consecutiveDatapointsToClear": 1  
  },  
  "suppressAlerts": true  
}
```

Example Exemple d'utilisation de ML Detect

```
{  
  "name": "Authorization failures ML behavior",  
  "metric": "aws:num-authorization-failures",  
  "criteria": {  
    "consecutiveDatapointsToAlarm": 1,  
    "consecutiveDatapointsToClear": 1,  
    "mlDetectionConfig": {  
      "confidenceLevel": "HIGH"  
    }  
  },  
  "suppressAlerts": true  
}
```

IP source (aws:adresse IP source)

L'adresse IP à partir de laquelle un appareil s'est connecté à AWS IoT.

Utilisez cette métrique pour spécifier un ensemble de routages interdomaines sans classe (auparavant sur liste blanche) ou non autorisés (auparavant sur liste noire) à partir duquel chaque appareil doit ou non se connecter à AWS IoT.

Compatible avec : Détection des règles

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-equals | greater-than | greater-than-equals

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-equals",  
        "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-equals",
    "statisticalThreshold": {
      "statistic": "p10"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de ML Detect

```
{
  "name": "Connection attempts ML behavior",
  "metric": "aws:num-connection-attempts",
  "criteria": {
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1,
    "mlDetectionConfig": {
      "confidenceLevel": "HIGH"
    }
  },
  "suppressAlerts": false
}
```

Se 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-equals | greater-than | greater-than-equals

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": "Disconnects",
  "metric": "aws:num-disconnects",
  "criteria": {
    "comparisonOperator": "less-than-equals",
    "value": {
      "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 1
  },
  "suppressAlerts": true
}
```

Example Exemple d'utilisation de **statisticalThreshold**

```
{  
    "name": "Disconnects",  
    "metric": "aws:num-disconnects",  
    "criteria": {  
        "comparisonOperator": "less-than-equals",  
        "statisticalThreshold": {  
            "statistic": "p10"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 1  
    },  
    "suppressAlerts": true  
}
```

Example Exemple d'utilisation de 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 section [CreateDimension](#).

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 de l'identifiant 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 (uniquement disponible pour Rules Detect)

- 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. Dans le [AWS IoT console](#), dans le panneau de navigation, DévelopperDéfendre, DévelopperDétecenter, puisProfils 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.
4. Pour Dimension, choisissez Ajouter une dimension.
5. Dans la page Create a new dimension (Créer une nouvelle dimension) entrez les détails de votre nouvelle dimension. Valeurs de dimensions prend en charge les caractères génériques # et + du MQTT et la variable de substitution de l'identifiant client MQTT \$ {iot :ClientId}.

The screenshot shows the 'Create a new dimension' dialog box. At the top, it says 'Create a new dimension'. Below that, a descriptive text states: 'Dimensions control the scope of behaviors that you define in your security profiles. For example, define a dimension that monitors specific MQTT topics.' The form fields are as follows:

- Dimension name: Room_Temperature
- Dimension type: Topic filter
- Dimension values: /temperature/room/+
- Add value button
- Tags section (with a plus icon)
- Cancel and Save buttons at the bottom

6. Choisissez Save (Enregistrer).
7. Vous pouvez également ajouter des dimensions aux mesures en option sous Indicateurs supplémentaires à conserver.
8. Pour terminer la création du comportement, saisissez les informations dans les autres champs obligatoires, puis choisissez Suivant.
9. Effectuez les étapes restantes pour terminer la création d'un profil de sécurité.

Pour afficher vos violations

1. Dans le [AWS IoT console](#), dans le panneau de navigation, DévelopperDéfendre, DévelopperDétecenter, puisInfractions.

The screenshot shows the AWS IoT Device Defender Violations page. At the top, it says "Device Defender > Detect > Violations". Below that, a section titled "Violations" has tabs for "Now" and "History", with "Now" selected. It displays "2 Thing(s) in alarm as of Mar 27, 2020 9:24:25 AM -0700". A table lists two entries:

Event time	Thing name	Security profile	Behavior	Last emitted
Mar 26, 2020 10:55:00 PM -0700	iotconsole-1585288160280-0	test_SP	TamperDetected	4 message(s)
Mar 26, 2020 10:55:00 PM -0700	iotconsole-1585288160280-1			Messages sent less than 1 in 5 minutes with dimension Tamper, with datapoints to Alarm: 1, and datapoints to Clear: 2

2. Dans le Behavior colonne, faites une pause sur le comportement pour lequel vous souhaitez consulter les informations relatives à la violation.

Pour afficher et mettre à jour vos dimensions

1. Dans le [AWS IoT console](#), dans le panneau de navigation, DévelopperDéfendre, DévelopperDétecter, puis Dimensions.
2. Sélectionnez la dimension que vous souhaitez modifier.
3. Sélectionnez Actions, puis Edit (Modifier).

The screenshot shows the AWS IoT Dimensions page. At the top, it says "Device Defender > Dimensions". Below that, a section titled "Dimensions (1)" has an "Actions" button with "Edit" and "Delete" options, and a "Create" button. A table lists one dimension:

Created date	Dimension name	Type	Value	Security profile(s)
Mar 27, 2020 3:22:51 PM -0700	Sensor_Temperature	Topic filter	/sensor/temperature/+	0 Security profile(s)

Pour supprimer une dimension

1. Dans le [AWS IoT console](#), dans le panneau de navigation, DévelopperDéfendre, DévelopperDétecter, 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.

EDIT SECURITY PROFILE

Expected behaviors

STEP 1/4

Name	Description (optional)
Temperature_Profile	An optional short description

Behaviors

Specify how your device **should behave**. You can use cloud-side metrics without a device agent deployed [learn more ↗](#)

Note: once created, behavior names cannot be edited. [?](#)

Name	Metric	Dimension (optional)	Dimension operator	...
Sensor_failures	Authorization failures	Sensor_Temperature	In	

Check type	Operator	Value	Duration
Absolute value	Greater than	5	5 minutes

Datapoints to alarm	Datapoints to clear
1	1

Name	Metric	Dimension (optional)	Dimension operator	...
Behavior name	Authorization failures	Select	Select	

Check type	Operator	Value	Duration
Absolute value	Greater than	5	5 minutes

Datapoints to alarm	Datapoints to clear
1	1

Add behavior

4. Choisissez Actions, puis Delete (Supprimer).

Comment utiliser les dimensions sur AWS 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. Utilisez la commande [CreateDimension](#) pour 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. Vous pouvez soit ajouter la dimension à un profil de sécurité existant en utilisant[UpdateSecurityProfile](#), ou ajoutez la dimension à un nouveau profil de sécurité en utilisant[CreateSecurityProfile](#). 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-2:123456789012:securityprofile/  
ProfileForConnectedDevice",  
    "securityProfileName": "ProfileForConnectedDevice"  
}
```

Pour gagner du temps, vous pouvez également charger un paramètre à partir d'un fichier au lieu de le saisir en tant que valeur de paramètre de ligne de commande. Pour plus d'informations, veuillez consulter la rubrique[Charger AWS CLI Paramètres 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"  
  }  
]
```

Ou utiliser[CreateSecurityProfile](#)utilisation de dimension avec ML, comme dans l'exemple qui suit :

```
aws iot create-security-profile --security-profile-name ProfileForConnectedDeviceML \
    --security-profile-description "Check to see if messages to
    TopicFilterForAuthMessages are abnormal" \
    --behaviors "[{\\"name\\":\\"test1\\",\\"metric\\":\\"aws:message-byte-size\\",
    \\"metricDimension\\":{\\"dimensionName\\": \"TopicFilterForAuthMessages\",\\"operator
    \\\": \"IN\"},\\"criteria\\":{\\"mlDetectionConfig\\":{\\"confidenceLevel\\\":\"HIGH\"},
    \\"consecutiveDatapointsToAlarm\\\":1,\"consecutiveDatapointsToClear\\\":1}}]" \
    --region us-west-2
```

Pour afficher les profils de sécurité avec une dimension

- Utilisez la commande [ListSecurityProfiles](#) 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

- Utilisez la commande [UpdateDimension](#) 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. Utilisez la commande [ListSecurityProfiles](#) pour afficher les profils de sécurité avec une certaine dimension.
2. Pour supprimer une dimension d'un profil de sécurité, utilisez la commande [UpdateSecurityProfile](#). 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 la commande [DeleteDimension](#) pour la supprimer :

```
aws iot delete-dimension \
```

```
--name TopicFilterForAuthMessages
```

Autorisations

Cette section contient des informations sur la manière de configurer les rôles et les stratégies IAM requises pour gérer AWS IoT Device DefenderDetect. Pour plus d'informations, consultez le [Guide de l'utilisateur IAM](#).

Accorder AWS IoT Device DefenderDetect l'autorisation de publier des alarmes dans une rubrique SNS.

Si vous utilisez le plugin `alarmTargets` paramètre dans [CreateSecurityProfile](#), 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"  
            ]  
        }  
    ]  
}
```

Politique 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 les profils de sécurité ML Detect ou Rules Detect, afin d'identifier et de surveiller les comportements inhabituels susceptibles d'indiquer qu'un périphérique est endommagé.

Commandes d'action d'DetectMitigation

Lancer et gérer l'exécution d'Detect

[Effectuez la tâche Détecter les actions d'atténuation](#)

[Décrire la tâche Détecter les actions d'atténuation](#)

[Liste des tâches Détecter les actions d'atténuation](#)

[Tâche Démarrer Détecter les actions d'atténuation](#)

[Lister les exécutions des actions d'atténuation de détection](#)

Commandes d'action de dimension

Lancer et gérer l'exécution de Dimension

[CreateDimension](#)

[DescribeDimension](#)

[ListDimensions](#)

[DeleteDimension](#)

[UpdateDimension](#)

Commandes d'action CustomMetric

Lancer et gérer CustomMetric exécution

[Créer une mesure personnalisée](#)

[Mettre à jour la mesure personnalisée](#)

[Décrire une mesure personnalisée](#)

[Lister les mesures personnalisées](#)

Lancer et gérer CustomMetric exécution

[Supprimer une mesure personnalisée](#)

Commandes d'action de profil de sécurité

Lancer et gérer l'exécution d'un profil de sécurité

[CreateSecurityProfile](#)

[AttachSecurityProfile](#)

[DetachSecurityProfile](#)

[DeleteSecurityProfile](#)

[DescribeSecurityProfile](#)

[ListTargetsForSecurityProfile](#)

[UpdateSecurityProfile](#)

[ValidateSecurityProfileBehaviors](#)

[ListSecurityProfilesForTarget](#)

Commandes d'action d'alarme

Gestion des alarmes et des cibles

[ListActiveViolations](#)

[ListViolationEvents](#)

[Mettre l'état de vérification en cas de violation](#)

Commandes d'action ML Detect

Liste des données d'entraînement de modèle ML

[Obtenir des résumés de formation sur le modèle comportemental](#)

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 [Envoyer de métriques à partir d'appareils \(p. 1125\)](#).
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. Utilisez [CreateSecurityProfile](#), mais ne spécifiez que ceux `additionalMetricsToRetain` qui vous intéresse. Ne spécifiez pas de valeur `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 \(p. 1126\)](#) et [Métriques côté appareil \(p. 1111\)](#). Après avoir créé un ensemble de comportements, vous pouvez les valider avec [ValidateSecurityProfileBehaviors](#).
 4. Utilisation de l'[CreateSecurityProfile](#)Actions pour créer un profil de sécurité incluant vos comportements. Vous pouvez utiliser le plugin `alarmTargets` pour envoyer des alertes à une cible (une rubrique SNS) lorsqu'un appareil ne respecte pas un comportement. (Si vous envoyez des alertes à l'aide de SNS, sachez que celles-ci sont comptabilisées pour votre Compte AWS Quota de sujets SNS. Il est possible qu'un grand nombre de violations dépasse votre quota de rubriques SNS. Vous pouvez également utiliser CloudWatch Métriques pour vérifier les violations. Pour plus d'informations, consultez [Utilisation de métriques AWS IoT \(p. 474\)](#).
 5. Utilisation de l'[AttachSecurityProfile](#)Utilisez pour 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 alertes. 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 (sans tenir compte des 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 le [ListActiveViolations](#)Utilisez pour voir les violations détectées pour un profil de sécurité ou un appareil cible donné.

Utilisation de l'[ListViolationEvents](#)Utilisez pour voir les violations détectées au cours d'une période donnée. Vous pouvez filtrer ces résultats par profil de sécurité, appareil ou état de vérification d'alarme.

7. Vous pouvez vérifier, organiser et gérer vos alertes en marquant leur état de vérification et en fournissant une description de cet état de vérification, à l'aide du [Mettre l'état de vérification en cas de violation](#)action.

8. Si vos appareils violent trop souvent ou trop rarement les comportements définis, vous pouvez peaufiner la définition de ces comportements.
9. Pour examiner les profils de sécurité configurés et les appareils surveillés, utilisez le [ListSecurityProfiles](#), [ListSecurityProfilesForTarget](#), et [ListTargetsForSecurityProfile](#) Actions.
Utilisation de l'[DescribeSecurityProfile](#) Utilisez pour obtenir plus de détails sur un profil de sécurité.
10. Pour mettre à jour un profil de sécurité, utilisez le [UpdateSecurityProfile](#) action. Utilisation de l'[DetachSecurityProfile](#) Action pour détacher un profil de sécurité d'un compte ou d'un groupe d'objets cible. Utilisation de l'[DeleteSecurityProfile](#) Actions pour supprimer entièrement un profil de sécurité.

Actions d'atténuation

Vous pouvez l'utiliser AWS IoT Device Defender pour prendre des mesures pour atténuer les problèmes détectés lors d'un résultat d'audit ou d'une alarme de détection.

Note

Aucune action d'atténuation ne sera exécutée sur les résultats d'audit supprimés. Pour plus plus sur la suppression des résultats d'audit, consultez [Vérification de la recherche de suppressions \(p. 1081\)](#).

Les mesures d'désynchronisation

AWS IoT Device Defender fournit des actions prédéfinies pour les différents contrôles d'audit. Vous configurez ces actions pour vous, Compte AWS puis vous les appliquez 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
INTERMEDIATE_CA_REVOKED_FOR_ACTIVE_DEVICE_CERTIFICATE_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_DEVICE_CERTIFICATE, ADD_THINGS_TO_THING_GROUP
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

Contrôle d'audit	Actions d'atténuation prises en charge
IOT_POLICY_OVERLY_PERMISSIVE_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK	PUBLISH_FINDING_TO_SNS, REPLACE_DEFAULT_POLICY_VERSION
CA_CERT_APPROACHING_EXPIRATION_CHECK	PUBLISH_FINDING_TO_SNS, UPDATE_CA_CERTIFICATE
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 sur Amazon SNS afin que vous puissiez prendre des mesures personnalisées en réponse 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.

IOT_POLICY_POTENTIAL_MISCONFIGURATION_CHECK

- Identifiez les erreurs de configuration potentielles dans AWS IoT les politiques.

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 Defender prend en charge les types de mesures d'atténuation suivants concernant les 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 politique avec une politique 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 incapables 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.

Déetecter les désynchronisation

AWS IoT Device Defender prend en charge les types de mesures d'atténuation suivants sur les alarmes de détection :

Type d'action	Remarques
ADD_THINGS_TO_THING_GROUP	You 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 AWS IoT console ou le AWS CLI pour 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 [page Actions d'atténuation dans la AWS IoT console](#).
2. Sur la page Actions d'atténuation, choisissez Créer.
3. Sur la page Créer une nouvelle action d'atténuation, dans Nom de l'action, entrez un nom unique pour votre action d'atténuation.
4. Dans Action Type (Type d'action), spécifiez le type d'action que vous souhaitez définir.
5. Dans Autorisations, choisissez le rôle IAM sous les autorisations duquel l'action est appliquée.
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 thing group (Ajouter des objets au groupe d'objets), choisissez le groupe de destination et sélectionnez ou désélectionnez Override dynamic groups (Remplacer groupes dynamiques).

7. Choisissez Créer pour enregistrer votre mesure d'atténuation sur votre AWS compte.

Utiliser l'interface de ligne de commande AWS CLI pour créer des actions d'atténuation

- Utilisez la commande [CreateMitigationAction](#) pour 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 [page Actions d'atténuation dans la AWS IoT console](#).

La page Actions d'atténuation affiche la liste de toutes les actions d'atténuation définies pour votre Compte AWS.

2. Choisissez le lien du nom de l'action d'atténuation que vous voulez modifier.
3. Choisissez Modifier et apportez vos modifications à l'action d'atténuation. Vous ne pouvez pas modifier le nom car le nom de l'action d'atténuation est utilisé pour l'identifier.
4. Choisissez Mettre à jour pour 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

- Utilisez la commande [ListMitigationAction](#) pour 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

- Utilisez la commande [UpdateMitigationAction](#) pour modifier votre action d'atténuation.

Pour utiliser la console AWS IoT pour supprimer une action d'atténuation

1. Ouvrez la [page Actions d'atténuation dans la AWS IoT console](#).

La page Actions d'atténuation affiche toutes les actions d'atténuation définies pour votre Compte AWS.

2. Choisissez l'action de désynchronisation, puis choisissez Remove (Supprimer).
3. Dans la fenêtre Êtes-vous sûr de vouloir supprimer, choisissez Supprimer.

Utiliser l'interface de ligne de commande AWS CLI pour supprimer des actions d'atténuation

- Utilisez la commande [UpdateMitigationAction](#) pour modifier votre action d'atténuation.

Utiliser la console AWS IoT pour afficher les détails d'une action d'atténuation

1. Ouvrez la [page Actions d'atténuation dans la AWS IoT console](#).

La page Actions d'atténuation affiche toutes les actions d'atténuation définies pour votre Compte AWS.

2. Choisissez le nom de l'action de désynchronisation que vous souhaitez afficher.

Utiliser l'interface de ligne de commande AWS CLI pour afficher les détails de l'action d'atténuation

- Utilisez la commande [DescribeMitigationAction](#) pour afficher les détails de votre action d'atténuation.

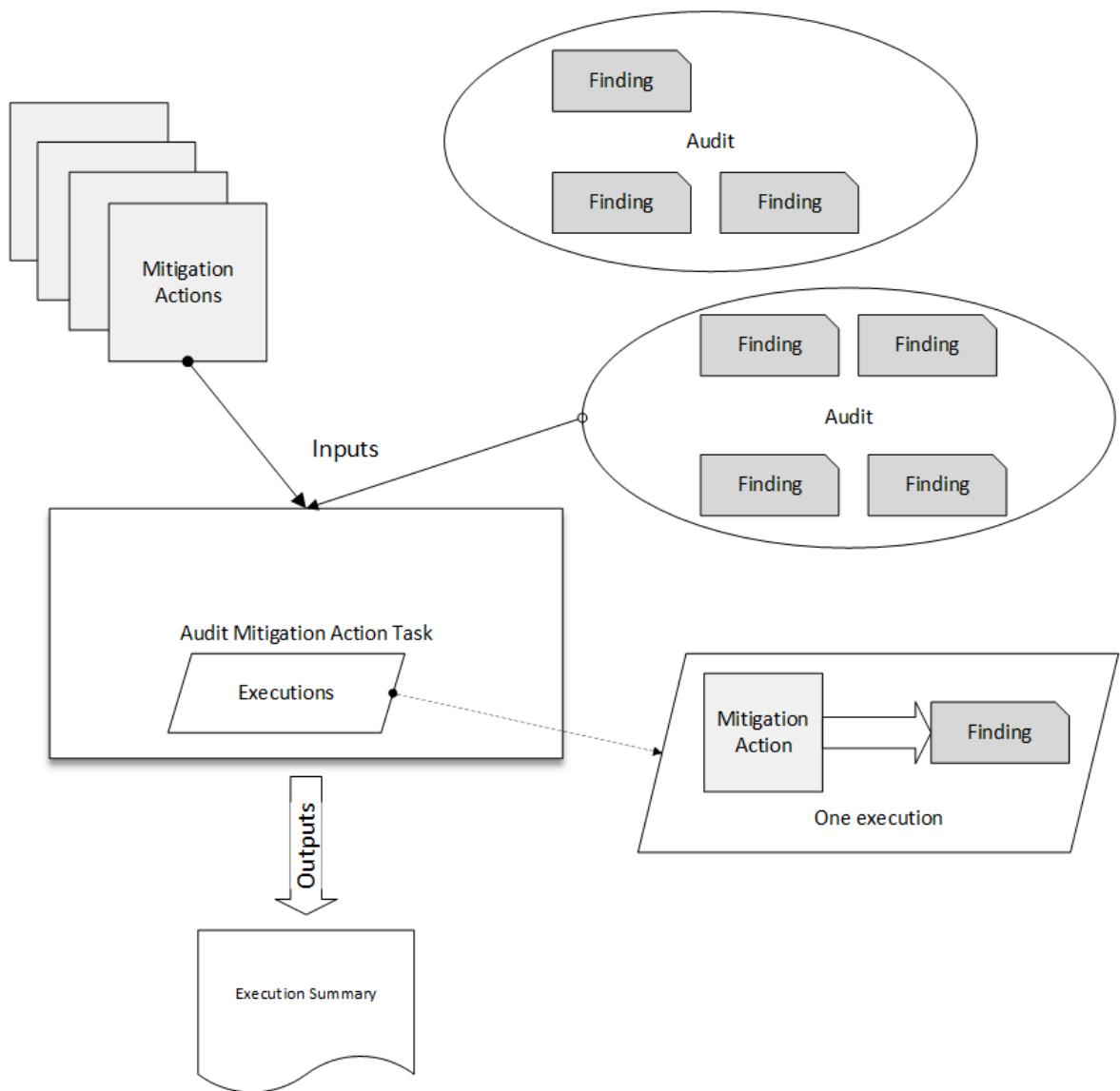
Appliquer des actions d'atténuation

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.

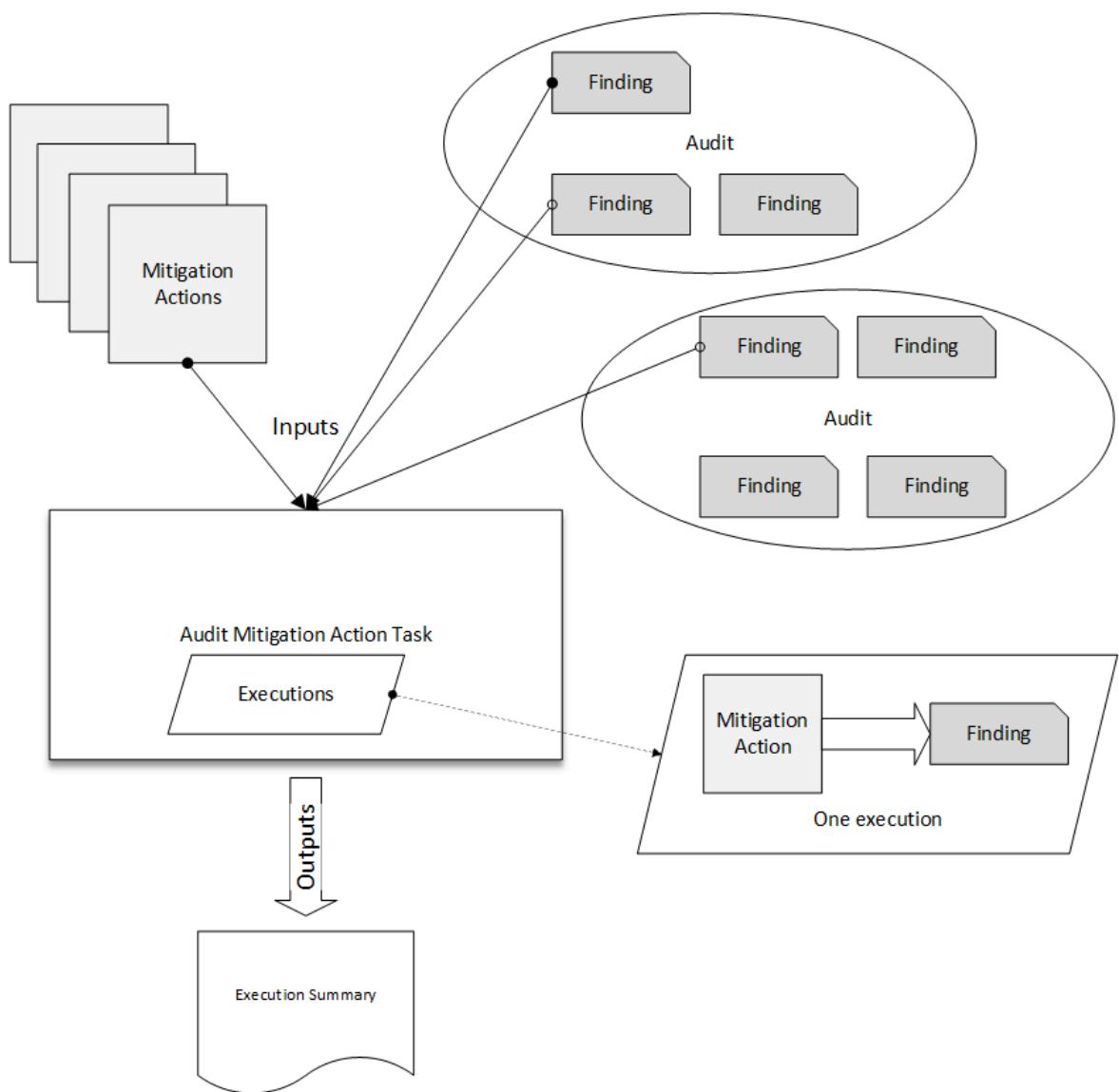
Vous pouvez afficher la liste des exécutions d'actions et annuler une exécution qui n'est pas encore terminée. Les actions déjà effectuée dans le cadre de l'exécution de l'action annulée ne sont pas restaurées. Si vous appliquez plusieurs actions à un ensemble de résultats et l'une de ces actions a échoué, 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 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 [page des résultats de l'audit dans laAWS IoT console](#).
2. Choisissez le nom pour l'audit auquel vous souhaitez appliquer des actions.
3. Choisissez Lancer les actions d'atténuation. Ce bouton n'est pas disponible si toutes vos vérifications sont conformes.
4. Dans Démarrer une nouvelle action d'atténuation, le nom de la tâche est par défaut l'ID d'audit, mais vous pouvez le remplacer par un nom plus significatif.
5. 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 est vide. Vous pouvez cliquer sur le lien [Créer une action d'atténuation](#) pour créer une ou plusieurs actions d'atténuation.

6. Lorsque vous avez spécifié toutes les actions que vous souhaitez appliquer, choisissez [Démarrer la tâche](#).

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 tous les résultats pour l'audit, utilisez la commande [ListAuditTasks](#) pour trouver l'ID de tâche.
2. Si vous souhaitez appliquer uniquement des actions à des résultats sélectionnés, utilisez la commande [ListAuditFindings](#) pour obtenir les ID des résultats.
3. Utilisez la commande [ListMitigationActions](#) et notez les noms des actions d'atténuation que vous souhaitez appliquer.
4. Utilisez la commande [StartAuditMitigationActionsTask](#) pour 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 [page Tâches d'action dans la AWS IoT console](#).

Une liste des tâches d'action indique le moment où chacune a été lancée et l'état actuel.

2. 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.

The screenshot shows the AWS IoT Device Defender console interface. At the top, a navigation bar includes 'Device Defender', 'Audit', and 'Action executions'. Below this, a specific task is selected: 'ff82164a6439e6024e83b4fc104817d7'. A large dark box contains the title 'MITIGATION ACTION EXECUTION TASK' and the task ID. Below this, the 'Details' section provides the following information:

Status	COMPLETED
Started at	Jun 6, 2019 6:09:07 PM -0700
Completed at	Jun 6, 2019 6:09:09 PM -0700

Under the 'Check summary' heading, there is a table showing the execution results for a single check named 'IoT policies overly permissive':

Check name	Failed	Successful	Skipped	Canceled	Total	Executions
IoT policies overly permissive	0	2	0	0	2	Show

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.

3. Pour afficher les détails de la tâche, dans Executions (Exécutions), choisissez Afficher.

The screenshot shows the AWS Device Defender Audit Action executions page. At the top, it displays a mitigation action execution task with the ID ff82164a6439e6024e83b4fc104817d7 and the finding "IoT policies overly permissive". Below this, there is a table titled "Action executions (4)" showing four completed actions. The table has columns for Started at, Status, Action, and Finding. The data is as follows:

Started at	Status	Action	Finding
Jun 6, 2019 6:09:08 PM -0700	Completed	sns_publish	053cff17-1da4-4479-996b-8b...
Jun 6, 2019 6:09:08 PM -0700	Completed	replace_default_policy_version	053cff17-1da4-4479-996b-8b...
Jun 6, 2019 6:09:08 PM -0700	Completed	replace_default_policy_version	2b966f76-b499-4986-836c-f8...

Utiliser l'interface de ligne de commande AWS CLI pour répertorier vos tâches lancées

1. Utilisez [ListAuditMitigationActionsTasks](#) pour 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. Utilisez [ListAuditMitigationActionsExecutions](#) pour afficher les détails d'exécution d'une tâche d'actions d'atténuation d'audit particulière.
3. Utilisez [DescribeAuditMitigationActionsTask](#) pour afficher des détails sur la tâche, tels que les paramètres spécifiés lorsqu'elle a été lancée.

Utiliser l'interface de ligne de commande AWS CLI pour annuler une tâche d'actions d'atténuation d'audit

1. Utilisez la commande [ListAuditMitigationActionsTasks](#) pour trouver l'ID de tâche pour la tâche dont vous voulez annuler l'exécution. Vous pouvez fournir des filtres pour affiner les résultats.
2. Utilisez la commande [ListDetectMitigationActionsExecutions](#), en utilisant 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.

Autorisations

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	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iot:UpdateCertificate"], "Resource": ["*"] }] }</pre>
UPDATE_CA_CERTIFICATE	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iot:UpdateCACertificate"], "Resource": ["*"] }] }</pre>
ADD_THINGS_TO_THING_GROUP	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iot>ListPrincipalThings", "iot>AddThingToThingGroup"], "Resource": ["*"] }] }</pre>

Type d'action	Modèle de stratégie d'autorisations
REPLACE_DEFAULT_POLICY_VERSION	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iot:CreatePolicyVersion"], "Resource": ["*"] }] }</pre>
ENABLE_IOT_LOGGING	<pre>{ "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>"] }] }</pre>

Type d'action	Modèle de stratégie d'autorisations
PUBLISH_FINDING_TO_SNS	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["sns:Publish"], "Resource": ["<The SNS topic to which the finding is published>"] }] }</pre>

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",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:iot:111122223333:*"
                },
                "StringEquals": {
                    "aws:SourceAccount": "111122223333:"
                }
            }
        }
    ]
}
```

Commandes d'action d'atténuation

Vous pouvez utiliser ces commandes d'action d'atténuation pour définir un ensemble d'actions pour votreCompte AWSque 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 Détecenter
CreateMitigationAction	CancelAuditMitigationActionsTask	Annuler la tâche Détecenter les actions d'atténuation
DeleteMitigationAction	DescribeAuditMitigationActionsTask	Décrire la tâche Détecenter les actions d'atténuation
DescribeMitigationAction	ListAuditMitigationActionsTasks	Liste des tâches Détecenter les actions d'atténuation
ListMitigationActions	StartAuditMitigationActionsTask	Task Démarrer Détecenter les actions d'atténuation
UpdateMitigationAction	ListAuditMitigationActionsExecutions	Lister les exécutions des actions d'atténuation de détection

Utilisation d'AWS IoT Device Defender avec d'autres services AWS

Utilisation AWS IoT Device Defender avec des appareils en cours d'exécution AWS IoT Greengrass

AWS IoT Greengrass fournit une intégration prédéfinie AWS IoT Device Defender permettant de surveiller en permanence le comportement des appareils.

- [Intégrez Device Defender à AWS IoT Greengrass V1](#)
- [Intégrez Device Defender à AWS IoT Greengrass V2](#)

Utilisation AWS IoT Device Defender avec FreeRTOS et appareils intégrés

Pour l'utiliser AWS IoT Device Defender sur un appareil FreeRTOS, le [SDK FreeRTOS Embedded C](#) ou la [bibliothèque AWS IoT Device Defender doivent être installés sur votre appareil](#). Le SDK FreeRTOS Embedded C inclut la bibliothèque AWS IoT Device Defender. Pour plus sur l'intégration AWS IoT Device Defender à vos appareils FreeRTOS, consultez les démos suivantes :

- [AWS IoT Device Defender pour des démonstrations de métriques standard et de métriques personnalisées FreeRTOS](#)
- [Utilisation de l'agent MQTT pour envoyer des métriques à AWS IoT Device Defender](#)
- [Utilisation de la bibliothèque principale MQTT pour envoyer des métriques à AWS IoT Device Defender](#)

Pour l'utiliser AWS IoT Device Defender sur un appareil intégré sans FreeRTOS, votre appareil doit disposer du [SDK AWS IoT Embedded C](#) ou de la [bibliothèque AWS IoT Device Defender](#). Le SDK AWS IoT Embedded C inclut la bibliothèque AWS IoT Device Defender. Pour plus d'informations sur l'intégration AWS

IoT Device Defender à vos appareils intégrés, consultez les démos suivantes, [AWS IoT Device Defender pour des démonstrations de mesures standard et personnalisées du SDK AWS IoT Embedded](#).

Utilisation de AWS IoT Device Defender avec AWS IoT Device Management

Vous pouvez utiliser l'indexation de la AWS IoT Device Management flotte pour indexer, rechercher et agréger AWS IoT Device Defender les violations détectées. Une fois les données relatives aux violations de Device Defender indexées dans l'indexation de la flotte, vous pouvez accéder aux données de violations de Device Defender et les interroger à partir des applications Fleet Hub, créer des alarmes de flotte basées sur les données de violations pour surveiller les anomalies sur votre parc d'appareils et consulter les alarmes de flotte dans les tableaux de bord de Fleet Hub.

Note

La fonction d'indexation des données sur les AWS IoT Device Defender violations, est en version préliminaire AWS IoT Management et susceptible d'être modifiée.

- [Gestion de l'indexation de la flotte](#)
- [de requête](#)
- [Gestion de l'indexation de la flotte pour les applications Fleet Hub](#)
- [Prise en main](#)

Intégration à AWS Security Hub

[AWS Security Hub](#) fournit une vue complète de votre état de sécurité dans AWS et vous permet de vérifier votre environnement par rapport aux normes et aux bonnes pratiques du secteur de la sécurité. Security Hub collecte des données de sécurité entre les Comptes AWS, les services et les produits tiers pris en charge. Security Hub pour analyser les tendances en matière de sécurité et identifier les problèmes de sécurité hautement prioritaires.

Grâce à AWS IoT Device Defender Security Hub, vous pouvez envoyer des résultats AWS IoT Device Defender à Security Hub. Security Hub inclut ces résultats dans son analyse de votre posture de sécurité.

Table des matières

- [Activation et configuration de l'intégration \(p. 1158\)](#)
- [Comment AWS IoT Device Defender envoie des résultats à Security Hub \(p. 1159\)](#)
 - [Types de résultats que AWS IoT Device Defender envoie \(p. 1159\)](#)
 - [Latence pour l'envoi des résultats \(p. 1159\)](#)
 - [Réessayer lorsque Security Hub n'est pas disponible \(p. 1160\)](#)
 - [Mise à jour des résultats existants dans Security Hub \(p. 1160\)](#)
- [Résultats types de AWS IoT Device Defender \(p. 1160\)](#)
- [Envoyer AWS IoT Device Defender des résultats à Security Hub \(p. 1164\)](#)

Activation et configuration de l'intégration

Avant AWS IoT Device Defender d'intégrer Security Hub, vous devez activer Security Hub. Pour obtenir Security Hub, veuillez consulter [Configuration Security Hub dans Security Hub dans Security Hub](#) dans Security Hub dans AWS Security Hub.

Une fois que vous avez activé Security Hub AWS IoT Device Defender et que vous avez activé Security Hub, ouvrez [la page Intégrations dans la console Security Hub](#), puis choisissez Accepter les résultats pour

l'audit, la détection ou les deux. AWS IoT Device Defender commence à envoyer les résultats à Security Hub.

Comment AWS IoT Device Defender envoie des résultats à Security Hub

Security Hub, les problèmes de sécurité sont suivis en tant que résultats. Certains résultats proviennent de problèmes qui sont détectés par d'autres AWS services ou par des produits tiers.

Security Hub fournit des outils permettant de gérer les résultats provenant de toutes ces sources. Vous pouvez afficher et filtrer les listes de résultats et afficher les informations sur un résultat. Pour de plus amples informations, consultez la section [Viewing findings](#) (Affichage des résultats) dans le Guide de l'utilisateur AWS Security Hub. Vous pouvez également suivre le statut d'une analyse dans un résultat. Pour de plus amples informations, veuillez consulter [Prendre des mesures en fonction des résultats](#) dans le Guide de l'utilisateur AWS Security Hub.

Tous les résultats dans Security Hub utilisent un format JSON standard appelé AWS Security Hub dans Security Hub dans Security Hub. Le format ASFF comprend des informations sur la source du problème, les ressources affectées et le statut actuel du résultat. Pour plus d'informations sur Security Finding Format (ASFF) dans le Guide de [AWS Security Finding Format \(ASFF\)](#) dans le Guide de AWS Security Hub l'utilisateur.

AWS IoT Device Defender est un des AWS services qui envoie les résultats à Security Hub.

Types de résultats que AWS IoT Device Defender envoie

Une fois que vous avez activé l'intégration de Security Hub, AWS IoT Device Defender Audit envoie les résultats qu'il génère (appelés résumés de vérification) à Security Hub. Les résumés des contrôles sont des informations générales relatives à un type de contrôle d'audit spécifique et à une tâche d'audit spécifique. Pour plus d'informations, veuillez consulter [Contrôle d'audit](#).

AWS IoT Device Defender Audit envoie des mises à jour de recherche à Security Hub pour les résumés des contrôles d'audit et les résultats d'audit pour chaque tâche d'audit. Si toutes les ressources trouvées dans les contrôles d'audit sont conformes ou si une tâche d'audit est annulée, Audit met à jour les résumés des contrôles dans Security Hub et les fait passer à l'état d'enregistrement ARCHIVÉ. Si une ressource a été signalée comme non conforme lors d'un contrôle d'audit, mais qu'elle l'a été lors de la dernière tâche d'audit, Audit la rend conforme et met également à jour le résultat dans Security Hub en tant qu'enregistrement ARCHIVÉ.

AWS IoT Device Defender Security Hub vers Security Hub. Ces résultats de violation incluent l'apprentissage automatique (ML), les comportements statistiques et statiques.

Pour envoyer les résultats à Security Hub, AWS IoT Device Defender utilisez Security Hub dans [AWS Security Hub, utilisez Security Hub](#). Dans le format ASFF, le champ Types fournit le type de résultat. Les résultats de AWS IoT Device Defender peuvent avoir les valeurs suivantes pour Types.

ComCommonComComComCom

Type de recherche pour les identifiants de client MQTT et les vérifications partagées de certificats d'appareils en conflit, et type de recherche pour Detect.

Vérification du logiciel et de la configuration/vulnérabilités

Type de résultat pour tous les autres contrôles d'audit.

Latence pour l'envoi des résultats

Lorsqu'AWS IoT Device Defender Audit crée un résultat, Security Hub est immédiatement envoyé à Security Hub. La latence dépend du volume des résultats générés lors de la tâche d'audit. Security Hub reçoit généralement les résultats dans l'heure qui suit.

AWS IoT Device Defender Detect vers presque en temps réel. Lorsqu'une violation entre en mode alarme ou sort du système d'alarme (c'est-à-dire que l'alarme est créée ou supprimée), le résultat correspondant au Security Hub est immédiatement créé ou archivé.

Réessayer lorsque Security Hub n'est pas disponible

Si Security Hub n'est pas disponible, AWS IoT Device Defender Audit et AWS IoT Device Defender Detect essaie de renvoyer les résultats jusqu'à ce qu'ils soient reçus.

Mise à jour des résultats existants dans Security Hub

Une fois le résultat AWS IoT Device Defender d'un audit envoyé à Security Hub, vous pouvez l'identifier à l'aide de l'identifiant de ressource vérifié et du type de contrôle d'audit. Si un résultat est généré avec Security Hub. AWS IoT Device Defender Si aucun résultat d'audit supplémentaire n'est généré lors d'une tâche d'audit ultérieure pour la même ressource et le même contrôle d'audit, la ressource devient conforme au contrôle d'audit. AWS IoT Device Defender Audit vers Security Hub.

AWS IoT Device Defender L'audit met également à jour les résumés des vérifications dans Security Hub. Si des ressources non conformes sont détectées lors d'un contrôle d'audit ou si le contrôle échoue, le statut de la découverte du Security Hub devient actif. Sinon, AWS IoT Device Defender Audit archive le résultat dans Security Hub.

AWS IoT Device Defender Detect permet à Security Hub de détecter une violation (par exemple, en cas d'alarme). Ce résultat n'est mis à jour que si l'un des critères suivants est rempli :

- La découverte expirera bientôt dans Security Hub. Envoyez donc AWS IoT Device Defender une mise à jour pour la tenir à jour. Les conclusions sont supprimées 90 jours après la dernière mise à jour ou 90 jours après la date de création si aucune mise à jour n'a lieu. Pour plus d'informations, consultez [la section Quotas de Security Hub](#) dans le Guide de AWS Security Hub l'utilisateur.
- La violation correspondante est désactivée. Son statut de recherche est donc AWS IoT Device Defender mis à jour sur ARCHIVÉ.

Résultats types de AWS IoT Device Defender

AWS IoT Device Defender utilise [AWS Security Hub dans](#) Security Hub.

L'exemple suivant montre un résultat typique de Security Hub pour un résultat d'audit. La `ReportType` victoireProductFields est `AuditFinding`.

```
{  
  "SchemaVersion": "2018-10-08",  
  "Id": "336757784525/IOT_POLICY/policyexample/1/IOT_POLICY_OVERLY_PERMISSIVE_CHECK/  
  ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",  
  "ProductArn": "arn:aws:securityhub:us-west-2::product/aws/iot-device-defender-audit",  
  "ProductName": "IoT Device Defender - Audit",  
  "CompanyName": "AWS",  
  "Region": "us-west-2",  
  "GeneratorId": "1928b87ab338ee2f541f6fab8c41c4f5",  
  "AwsAccountId": "123456789012",  
  "Types": [  
    "Software and Configuration Check/Vulnerabilities"  
  ],  
  "CreatedAt": "2022-11-06T22:11:40.941Z",  
  "UpdatedAt": "2022-11-06T22:11:40.941Z",  
  "Severity": {  
    "Label": "CRITICAL",  
    "Normalized": 90  
  },  
}
```

```

    "Title": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK:  
ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",  
    "Description": "IOT_POLICY policyexample:1 is reported as non-compliant for  
IOT_POLICY_OVERLY_PERMISSIVE_CHECK by Audit task 9f71b6e90cfb57d4ac671be3a4898e6a.  
The non-compliant reason is Policy allows broad access to IoT data plane actions:  
[iot:Connect].",  
    "SourceUrl": "https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/policy/  
policyexample",  
    "ProductFields": {  
        "CheckName": "IOT_POLICY_OVERLY_PERMISSIVE_CHECK",  
        "TaskId": "9f71b6e90cfb57d4ac671be3a4898e6a",  
        "TaskType": "ON_DEMAND_AUDIT_TASK",  
        "PolicyName": "policyexample",  
        "IsSuppressed": "false",  
        "ReasonForNonComplianceCode": "ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",  
        "ResourceType": "IOT_POLICY",  
        "FindingId": "1928b87ab338ee2f541f6fab8c41c4f5",  
        "PolicyVersionId": "1",  
        "ReportType": "AuditFinding",  
        "TaskStartTime": "1667772700554",  
        "aws/securityhub/FindingId": "arn:aws:securityhub:us-west-2::product/aws/iot-device-  
defender-audit/336757784525/IOT_POLICY/policyexample/1/IOT_POLICY_OVERLY_PERMISSIVE_CHECK/  
ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS",  
        "aws/securityhub/ProductName": "IoT Device Defender - Audit",  
        "aws/securityhub/CompanyName": "AWS"  
    },  
    "Resources": [  
        {  
            "Type": "AwsIotPolicy",  
            "Id": "policyexample",  
            "Partition": "aws",  
            "Region": "us-west-2",  
            "Details": {  
                "Other": {  
                    "PolicyVersionId": "1"  
                }  
            }  
        }  
    ],  
    "WorkflowState": "NEW",  
    "Workflow": {  
        "Status": "NEW"  
    },  
    "RecordState": "ACTIVE",  
    "FindingProviderFields": {  
        "Severity": {  
            "Label": "CRITICAL"  
        },  
        "Types": [  
            "Software and Configuration Check/Vulnerabilities"  
        ]  
    }  
}

```

L'exemple suivant montre un résultat provenant de Security Hub concernant un résumé des vérifications d'audit. La `ReportType` `victoire` `ProductFields` est `CheckSummary`.

```
{
    "SchemaVersion": "2018-10-08",
    "Id": "615243839755/SCHEDULED_AUDIT_TASK/daily_audit_schedule_checks/  
DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-audit",
}
```

```

    "ProductName": "IoT Device Defender - Audit",
    "CompanyName": "AWS",
    "Region": "us-east-1",
    "GeneratorId": "f3021945485adf92487c273558fcaa51",
    "AwsAccountId": "123456789012",
    "Types": [
        "Software and Configuration Check/Vulnerabilities/CVE"
    ],
    "CreatedAt": "2022-10-18T14:20:13.933Z",
    "UpdatedAt": "2022-10-18T14:20:13.933Z",
    "Severity": {
        "Label": "CRITICAL",
        "Normalized": 90
    },
    "Title": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK Summary: Completed with 2 non-compliant resources",
    "Description": "Task f3021945485adf92487c273558fcaa51 of weekly scheduled Audit daily_audit_schedule_checks completes. 2 non-compliant resources are found for DEVICE_CERTIFICATE_KEY_QUALITY_CHECK out of 1000 resources in the account. The percentage of non-compliant resources is 0.2.%.",
    "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/audit/results/f3021945485adf92487c273558fcaa51/DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
    "ProductFields": {
        "TaskId": "f3021945485adf92487c273558fcaa51",
        "TaskType": "SCHEDULED_AUDIT_TASK",
        "ScheduledAuditName": "daily_audit_schedule_checks",
        "CheckName": "DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
        "ReportType": "CheckSummary",
        "CheckRunStatus": "COMPLETED_NON_COMPLIANT",
        "NonCompliantResourcesCount": "2",
        "SuppressedNonCompliantResourcesCount": "1",
        "TotalResourcesCount": "1000",
        "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-audit/615243839755/SCHEDULED/daily_audit_schedule_checks/DEVICE_CERTIFICATE_KEY_QUALITY_CHECK",
        "aws/securityhub/ProductName": "IoT Device Defender - Audit",
        "aws/securityhub/CompanyName": "AWS"
    },
    "Resources": [
        {
            "Type": "AwsIotAuditTask",
            "Id": "f3021945485adf92487c273558fcaa51",
            "Region": "us-east-1"
        }
    ],
    "WorkflowState": "NEW",
    "Workflow": {
        "Status": "NEW"
    },
    "RecordState": "ACTIVE",
    "FindingProviderFields": {
        "Severity": {
            "Label": "CRITICAL"
        },
        "Types": [
            "Software and Configuration Check/Vulnerabilities/CVE"
        ]
    }
}

```

L'exemple suivant Security Hub représente Security Hub dans AWS IoT Device Defender Security Hub.

```
{
}
```

```

"SchemaVersion": "2018-10-08",
"Id": "e92a782593c6f5b1fc7cb6a443dc1a12",
"ProductArn": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-detect",
"ProductName": "IoT Device Defender - Detect",
"CompanyName": "AWS",
"Region": "us-east-1",
"GeneratorId": "arn:aws:iot:us-east-1:123456789012:securityprofile/MySecurityProfile",
"AwsAccountId": "123456789012",
"Types": [
    "Unusual Behaviors"
],
"CreatedAt": "2022-11-09T22:45:00Z",
"UpdatedAt": "2022-11-09T22:45:00Z",
"Severity": {
    "Label": "MEDIUM",
    "Normalized": 40
},
"Title": "Registered thing MyThing is in alarm for STATIC behavior MyBehavior.",
"Description": "Registered thing MyThing violates STATIC behavior MyBehavior of security profile MySecurityProfile. Violation was triggered because the device did not conform to aws:num-disconnects less-than 1.",
"SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/dd/securityProfile/MySecurityProfile?tab=violations",
"ProductFields": {
    "ComparisonOperator": "less-than",
    "BehaviorName": "MyBehavior",
    "ViolationId": "e92a782593c6f5b1fc7cb6a443dc1a12",
    "ViolationStartTime": "1668033900000",
    "SuppressAlerts": "false",
    "ConsecutiveDatapointsToAlarm": "1",
    "ConsecutiveDatapointsToClear": "1",
    "DurationSeconds": "300",
    "Count": "1",
    "MetricName": "aws:num-disconnects",
    "BehaviorCriteriaType": "STATIC",
    "ThingName": "MyThing",
    "SecurityProfileName": "MySecurityProfile",
    "aws/securityhub/FindingId": "arn:aws:securityhub:us-east-1::product/aws/iot-device-defender-detect/e92a782593c6f5b1fc7cb6a443dc1a12",
    "aws/securityhub/ProductName": "IoT Device Defender - Detect",
    "aws/securityhub/CompanyName": "AWS"
},
"Resources": [
{
    "Type": "AwsIotRegisteredThing",
    "Id": "MyThing",
    "Region": "us-east-1",
    "Details": {
        "Other": {
            "SourceUrl": "https://us-east-1.console.aws.amazon.com/iot/home?region=us-east-1#/thing/MyThing?tab=violations",
            "IsRegisteredThing": "true",
            "ThingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyThing"
        }
    }
}
],
"WorkflowState": "NEW",
"Workflow": {
    "Status": "NEW"
},
"RecordState": "ACTIVE",
"FindingProviderFields": {
    "Severity": {
        "Label": "MEDIUM"
    }
},

```

```
    "Types": [
        "Unusual Behaviors"
    ]
}
```

Envoi AWS IoT Device Defender des résultats à Security Hub

Pour arrêter l'envoi des résultats à Security Hub, vous pouvez utiliser la console Security Hub ou l'API.

Pour plus d'informations, veuillez consulter [Désactivation et activation du flux de résultats à partir d'une intégration \(console\)](#) ou [Désactivation du flux de résultats d'une intégration \(API Security Hub,AWS CLI\)](#) dans le Guide deAWS Security Hub l'utilisateur.

Prévention du député confus entre services

Le problème de député confus est un problème de sécurité dans lequel une entité qui n'est pas autorisée à effectuer une action peut contraindre une entité plus privilégiée à le faire. Dans AWS, l'emprunt d'identité entre services peut entraîner le problème de député confus. L'usurpation d'identité entre services peut se produire lorsqu'un service (le service appelant) appelle un autre service (le service appelé). Le service appelant peut être manipulé pour utiliser ses autorisations afin d'agir sur les ressources d'un autre client de sorte qu'il n'y aurait pas accès autrement. Pour éviter cela, AWS fournit des outils qui vous aident à protéger vos données pour tous les services avec des principaux de service qui ont eu accès aux ressources de votre compte.

Vous pouvez accéder à trois ressources AWS IoT Device Defender qui peuvent être affectés par ce problème de sécurité confus lié à un adjoint, à savoir l'exécution d'audits, l'envoi de notifications SNS en cas de violation du profil de sécurité et l'exécution de mesures d'atténuation. Pour chacune de ces actions, les valeurs de `aws:SourceArn` doivent être les suivantes :

- Pour les ressources transmises dans [UpdateAccountAuditConfiguration](#)l'API (`RoleArn` et les `RoleArn` attributs `NotificationTarget`), vous devez définir la politique des ressources en utilisant `aws:SourceArn asarn:arnPartition:iot:region:accountId:`.
- Pour les ressources transmises dans [CreateMitigationAction](#)l'API (`'RoleArn` attribut), vous devez définir la politique des ressources en utilisant `aws:SourceArn asarn:arnPartition:iot:region:accountId:mitigationaction/mitigationActionName.`
- Pour les ressources transmises dans l'[CreateSecurityProfile](#)API (attribut `alertTargets`), vous devez définir la politique des ressources en utilisant `aws:SourceArn asarn:arnPartition:iot:region:accountId:securityprofile/securityprofileName.`

Le moyen le plus efficace de se protéger contre le problème de député confus consiste à utiliser la clé de contexte de condition globale `aws:SourceArn` avec l'ARN complet de la ressource. Si vous ne connaissez pas l'ARN complet de la ressource ou si vous spécifiez plusieurs ressources, utilisez la clé de contexte de condition globale `aws:SourceArn` avec des caractères génériques (*) pour les parties inconnues de l'ARN. Par exemple, `arn:aws:servicename:*:123456789012:*`.

L'exemple suivant montre comment utiliser les clés de contexte de condition globale `aws:SourceArn` et `aws:SourceAccount` dans AWS IoT Device Defender afin d'éviter le problème de l'adjoint confus.

```
{
"Version": "2012-10-17",
"Statement": {
    "Sid": "ConfusedDeputyPreventionExamplePolicy",
```

```
"Effect": "Allow",
"Principal": {
    "Service": "iot.amazonaws.com"
},
>Action": "sts:AssumeRole",
"Condition": {
    "ArnLike": {
        "aws:SourceArn": "arn:aws:iot:*:123456789012::*"
    },
    "StringEquals": {
        "aws:SourceAccount": "123456789012"
    }
}
}
```

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 configuration d'un processus pour qu'il s'exécute avec des autorisations réduites, consultez les instructions de configuration incluses dans 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 et les noms de unique-per-device processus sont préférés. 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 \(p. 315\)](#)
- [Comprendre le modèleAWS IoT 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 Advisor](#) est une fonctionnalité de test entièrement gérée sur le cloud qui permet de valider les appareils IoT lors du développement logiciel des appareils. Device Advisor fournit des tests prédéfinis que vous pouvez utiliser pour valider les appareils IoT afin de garantir une connectivité fiable et sécurisée AWS IoT Core, avant de les déployer en production. [Les tests prédéfinis de Device Advisor vous aident à valider le logiciel de votre appareil par rapport aux meilleures pratiques en matière d'utilisation des protocoles TLS, MQTT, Device Shadow et IoT Jobs.](#) Vous pouvez également télécharger des rapports de qualification signés à soumettre au réseau de AWS partenaires afin que votre appareil soit qualifié pour le [catalogue d'appareils AWS partenaires](#) sans avoir à envoyer votre appareil et à attendre qu'il soit testé.

Note

Device Advisor est pris en charge dans les régions us-west-2, ap-northeast-1 et eu-west-1. Device Advisor prend en charge les appareils et les clients qui utilisent les protocoles MQTT et MQTT over WebSocket Secure (WSS) pour publier des messages et s'y abonner. Tous les protocoles prennent en charge IPv4 et IPv6.

Device Advisor prend en charge les certificats de serveur RSA.

Tout appareil conçu pour se connecter AWS IoT Core peut tirer parti de Device Advisor. Vous pouvez accéder à Device Advisor depuis la [AWS IoTconsole](#) ou à l'aide du SDK AWS CLI ou. Lorsque vous êtes prêt à tester votre appareil, enregistrez-le auprès du terminal Device Advisor AWS IoT Core et configurez le logiciel de l'appareil à l'aide du terminal Device Advisor. Choisissez ensuite les tests prédéfinis, configurez-les, exécutez les tests sur votre appareil et obtenez les résultats des tests ainsi que des journaux détaillés ou un rapport de qualification.

Device Advisor est un point de terminaison de test dans le AWS cloud. Vous pouvez tester vos appareils en les configurant pour qu'ils se connectent au point de terminaison de test fourni par Device Advisor. Une fois qu'un appareil est configuré pour se connecter au terminal de test, vous pouvez accéder à la console du Device Advisor ou utiliser le AWS SDK pour choisir les tests que vous souhaitez 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 finaux sous la forme d'un rapport de test.

Protocoles TLS

Le protocole Transport Layer Security (TLS) est utilisé pour chiffrer les données confidentielles sur des réseaux non sécurisés, comme Internet. Le protocole TLS est le successeur du protocole SSL (Secure Sockets Layer).

Device Advisor prend en charge les protocoles TLS suivants :

- TLS 1.3 (recommandé)
- TLS 1.2

Protocoles, mappages de ports et authentification

Le protocole de communication de l'appareil est utilisé par un appareil ou un client pour se connecter au courtier de messages à l'aide d'un point de terminaison de l'appareil. Le tableau suivant répertorie les protocoles pris en charge par les terminaux Device Advisor, ainsi que les méthodes d'authentification et les ports utilisés.

Protocoles, authentification et mappages de port

Protocole	Opérations prises en charge	Authentification	Port	Nom du protocole ALPN
MQTT terminé WebSocket	Publier, s'abonner	Signature Version 4	443	N/A
MQTT	Publier, s'abonner	Certificat de client X.509	8883	x-amzn-mqtt-ca
MQTT	Publier, s'abonner	Certificat de client X.509	443	N/A

Ce chapitre contient les sections suivantes :

- [Configuration \(p. 1168\)](#)
- [Commencer à utiliser Device Advisor dans la console \(p. 1173\)](#)
- [Flux de Device Advisor \(p. 1180\)](#)
- [Flux de travail détaillé de Device Advisor sur console \(p. 1184\)](#)
- [Flux de travail de console de tests de longue durée \(p. 1194\)](#)
- [Points de terminaison VPC Device Advisor \(\) AWS PrivateLink \(p. 1200\)](#)
- [Scénarios de test de Device Advisor \(p. 1203\)](#)

Configuration

Avant d'utiliser Device Advisor pour la première fois, exécutez les tâches suivantes :

Créez un objet IoT

Tout d'abord, créez un objet IoT et associez-y un certificat. Pour un didacticiel sur la création d'objets, voir [Création d'un objet IoT](#).

Créez un rôle IAM à utiliser comme rôle de votre appareil

Note

Vous pouvez créer rapidement le rôle de l'appareil à l'aide de la console Device Advisor. Pour savoir comment configurer votre rôle sur l'appareil avec la console Device Advisor, consultez la section [Prise en main du Device Advisor dans la console](#).

1. Accédez à la [AWS Identity and Access Managementconsole](#) et connectez-vous à celle Compte AWS que vous utilisez pour les tests de Device Advisor.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Choisissez Create Policy (Créer une politique).
4. Sous Create Politiques, procédez comme suit :
 - a. Pour Service, choisissez IoT.
 - b. Sous Actions, effectuez l'une des opérations suivantes :

- (Recommandé) Sélectionnez les actions en fonction de la politique associée à l'objet ou au certificat IoT que vous avez créé dans la section précédente.
- Recherchez les actions suivantes dans la zone Action de filtre et sélectionnez-les :

- Connect
- Publish
- Subscribe
- Receive
- RetainPublish

c. Sous Ressources, limitez le client, le sujet et les ressources du sujet. Restreindre ces ressources est une bonne pratique de sécurité. Pour limiter les ressources, procédez comme suit :

- i. Choisissez Spécifier l'ARN de la ressource client pour l'action Connect.
- ii. Choisissez Ajouter un ARN, puis effectuez l'une des opérations suivantes :

Note

Le clientId est l'ID client MQTT que votre appareil utilise pour interagir avec Device Advisor.

- Spécifiez la région, l'accountID et l'clientId dans l'éditeur visuel ARN.
 - Entrez manuellement les noms de ressources Amazon (ARN) des sujets liés à l'IoT avec lesquels vous souhaitez exécuter vos scénarios de test.
- iii. Choisissez Add (Ajouter).
 - iv. Choisissez Spécifier l'ARN de la ressource thématique pour la réception et une autre action.
 - v. Choisissez Ajouter un ARN, puis effectuez l'une des opérations suivantes :

Note

Le nom du sujet est le sujet MQTT sur lequel votre appareil publie des messages.

- Spécifiez la région, l'accountID et le nom du sujet dans l'éditeur visuel ARN.
 - Entrez manuellement les ARN des sujets IoT avec lesquels vous souhaitez exécuter vos scénarios de test.
- vi. Choisissez Add (Ajouter).
 - vii. Choisissez Spécifier l'ARN de la ressource TopicFilter pour l'action S'abonner.
 - viii. Choisissez Ajouter un ARN, puis effectuez l'une des opérations suivantes :

Note

Le nom de la rubrique est la rubrique MQTT à laquelle votre appareil est abonné.

- Spécifiez la région, l'accountID et le nom du sujet dans l'éditeur visuel ARN.
- Entrez manuellement les ARN des sujets IoT avec lesquels vous souhaitez exécuter vos scénarios de test.

ix. Choisissez Add (Ajouter).

5. Choisissez Next: Tags (Suivant : Balises).
6. Choisissez Next: Review (Suivant : Vérification).
7. Dans la section Révision de la politique, entrez le nom de votre politique.
8. Choisissez Create Policy (Créer une politique).
9. Dans le panneau de navigation de gauche, choisissez Rôles.
10. Choisissez Create Role (Créer un rôle).
11. Sous Sélectionner une entité de confiance, choisissez Politique de confiance personnalisée.
12. Entrez la politique de confiance suivante dans la zone Politique de confiance personnalisée :

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

13. Vous voyez [Prévention confus entre services pour Device Advisor \(p. 464\)](#), pour éviter le problème du député confus.
14. Choisissez Suivant.
15. Choisissez la politique que vous avez créée à l'étape 4.
16. (Facultatif) Sous Définir la limite des autorisations, choisissez Utiliser une limite d'autorisations pour contrôler les autorisations de rôle maximales, puis sélectionnez la politique que vous avez créée.
17. Choisissez Suivant.
18. Entrez un nom de rôle et une description du rôle.
19. Sélectionnez Create role (Créer un rôle).

Création d'une politique gérée personnalisée permettant à un utilisateur IAM d'utiliser Device Advisor

1. Connectez-vous à la console IAM à l'adresse <https://console.aws.amazon.com/iam/>. Si vous y êtes invité, saisissez vos AWS informations d'identification pour vous connecter.
2. Dans le panneau de navigation de gauche, choisissez Politiques.
3. Choisissez Create Policy, puis choisissez l'onglet JSON.
4. Ajoutez les autorisations nécessaires pour utiliser Device Advisor. Le document de politique se trouve dans la rubrique [Bonnes 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).

Création d'un utilisateur IAM pour utiliser Device Advisor

Note

Nous vous recommandons de créer un utilisateur IAM à utiliser lorsque vous exécutez les tests Device Advisor. L'exécution des tests de Device Advisor par un utilisateur administrateur peut présenter des risques de sécurité et n'est pas recommandée.

1. Accédez à la console IAM à l'[adresse https://console.aws.amazon.com/iam/](https://console.aws.amazon.com/iam/) Si vous y êtes invité, entrez vos AWS informations d'identification pour vous connecter.
2. Dans le panneau de navigation de gauche, choisissez Utilisateurs.
3. Sélectionnez Add User (Ajouter un utilisateur).
4. Entrez un nom d'utilisateur.
5. Les utilisateurs ont besoin d'un accès programmatique s'ils souhaitent interagir avec AWS en dehors de la AWS Management Console. La manière d'octroyer un accès par programmation dépend du type d'utilisateur qui accède à AWS.

Pour accorder aux utilisateurs un accès programmatique, choisissez l'une des options suivantes.

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Bit
Identité de la main-d'œuvre (Utilisateurs gérés dans IAM Identity Center)	Utilisez des informations d'identification temporaires pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	<p>Suivez les instructions de l'interface que vous souhaitez utiliser.</p> <ul style="list-style-type: none"> • Pour l'AWS CLI, veuillez consulter la rubrique Configuration de l'AWS CLI pour l'utilisation d'AWS IAM Identity Center (successor to AWS Single Sign-On) dans le Guide de l'utilisateur AWS Command Line Interface. • Pour les kits SDK et les outils AWS ainsi que les API AWS, veuillez consulter la rubrique Authentification IAM Identity Center dans le Guide de référence des kits SDK et des outils AWS.
IAM	Utilisez des informations d'identification temporaires pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	<p>Suivez les instructions de la section Utilisation d'informations d'identification temporaires avec des ressources AWS dans le Guide de l'utilisateur IAM.</p>
IAM	(Non recommandé) Utilisez des informations d'identification à long terme pour signer des demandes par programmation destinées à l'AWS CLI, aux kits SDK AWS ou aux API AWS.	<p>Suivez les instructions de l'interface que vous souhaitez utiliser.</p> <ul style="list-style-type: none"> • Pour l'AWS CLI, veuillez consulter la rubrique Authentification à l'aide des informations d'identification d'utilisateur IAM dans le Guide de l'utilisateur AWS Command Line Interface. • Pour les kits SDK et les outils AWS, veuillez consulter la rubrique Authentification à l'aide d'informations

Quel utilisateur a besoin d'un accès programmatique ?	Pour	Bit
		<p><u>d'identification à long terme</u> dans le Guide de référence des kits SDK et des outils AWS.</p> <ul style="list-style-type: none"> Pour les API AWS, veuillez consulter la rubrique Gestion des clés d'accès pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

6. Sélectionnez Next: Permissions (Étape suivante : autorisations).
7. Pour activer l'accès, ajoutez des autorisations à vos utilisateurs, groupes ou rôles :
 - Utilisateurs et groupes dans AWS IAM Identity Center (successor to AWS Single Sign-On) :

Créez un jeu d'autorisations. Suivez les instructions de la rubrique [Création d'un jeu d'autorisations](#) du Guide de l'utilisateur AWS IAM Identity Center (successor to AWS Single Sign-On).
 - Utilisateurs gérés dans IAM par un fournisseur d'identité :

Créez un rôle pour la fédération d'identité. Pour plus d'informations, voir la rubrique [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) du Guide de l'utilisateur IAM.
 - Utilisateurs IAM :
 - Créez un rôle que votre utilisateur peut assumer. Suivez les instructions de la rubrique [Création d'un rôle pour un utilisateur IAM](#) du Guide de l'utilisateur IAM.
 - (Non recommandé) Attachez une politique directement à un utilisateur ou ajoutez un utilisateur à un groupe d'utilisateurs. Suivez les instructions de la rubrique [Ajout d'autorisations à un utilisateur \(console\)](#) du Guide de l'utilisateur IAM.
8. Entrez le nom de la politique personnalisée que vous avez créée dans le champ de recherche. Ensuite, cochez la case pour le nom de la stratégie.
9. Choisissez Next: Tags (Suivant : Balises).
10. Choisissez Next: Review (Suivant : Vérification).
11. Choisissez Create user.
12. Choisissez Close (Fermer).

Device Advisor doit accéder à vos AWS ressources (objets, certificats et points 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 politique d'autorisations nécessaire à votre utilisateur IAM.

Configurer votre appareil

Device Advisor utilise l'extension TLS d'indication du nom de serveur (SNI) pour appliquer les configurations TLS. Les appareils doivent utiliser cette extension lorsqu'ils se connectent et transmettent un nom de serveur identique à celui du point de terminaison de test de Device Advisor.

Device Advisor autorise la connexion TLS lorsqu'un test est en coursRunning. Il refuse la connexion TLS avant et après chaque test. C'est pourquoi nous vous recommandons d'utiliser le mécanisme de nouvelle tentative de connexion à l'appareil pour bénéficier d'une expérience de test entièrement automatisée avec Device Advisor. Vous pouvez exécuter des suites de tests qui incluent plusieurs scénarios de test, tels que TLS connect, MQTT connect et MQTT publish. Si vous exécutez plusieurs scénarios de test, nous recommandons que votre appareil essaie de se connecter à notre point de terminaison de test toutes les cinq secondes. Vous pouvez ensuite automatiser l'exécution de plusieurs scénarios de test en séquence.

Note

Pour préparer le logiciel de votre appareil à des fins de test, nous vous recommandons d'utiliser un SDK pouvant se connecter à AWS IoT Core. Vous devez ensuite mettre à jour le SDK avec le point de terminaison de test Device Advisor fourni pour votre Compte AWS.

Device Advisor prend en charge deux types de terminaux : les terminaux au niveau du compte et au niveau de l'appareil. Choisissez le point de terminaison le mieux adapté à votre cas d'utilisation. Pour exécuter simultanément plusieurs suites de tests pour différents appareils, utilisez un point de terminaison au niveau de l'appareil.

Exécutez la commande suivante pour obtenir le point de terminaison au niveau du périphérique :

Pour les clients MQTT utilisant des certificats client X.509 :

```
aws iotdeviceadvisor get-endpoint --thing-arn your-thing-arn
```

or

```
aws iotdeviceadvisor get-endpoint --certificate-arn your-certificate-arn
```

Pour les clients MQTT plutôt que pour WebSocket les clients utilisant la version 4 de Signature :

```
aws iotdeviceadvisor get-endpoint --device-role-arn your-device-role-arn --authentication-method SignatureVersion4
```

Pour exécuter une suite de tests à la fois, choisissez un point de terminaison au niveau du compte. Exécutez la commande suivante pour obtenir le point de terminaison au niveau du compte :

```
aws iotdeviceadvisor get-endpoint
```

Commencer à utiliser Device Advisor dans la console

Ce didacticiel vous aide à commencer avec AWS IoT Core Device Advisor la console. Device Advisor propose des fonctionnalités telles que les tests obligatoires et les rapports de qualification signés. Vous pouvez utiliser ces tests et rapports pour qualifier et répertorier les appareils dans le [catalogue des appareils AWS partenaires](#), comme indiqué dans le [programme de AWS IoT Core qualification](#).

Pour plus d'informations sur l'utilisation de Device Advisor, consultez [Flux de Device Advisor \(p. 1180\)](#) et [Flux de travail détaillé de Device Advisor sur console \(p. 1184\)](#).

Pour effectuer ce didacticiel, suivez les étapes décrites dans [Configuration \(p. 1168\)](#).

Note

Device Advisor est pris en charge dans les versions suivantes Régions AWS :

- USA Est (Virginie du Nord)
- USA Ouest (Oregon)
- Asie Pacifique (Tokyo)
- Europe (Irlande)

Démarrer

1. Dans le volet [de navigation de la AWS IoT console](#), sous Test, choisissez Device Advisor. Cliquez ensuite sur le bouton Démarrer la procédure pas à pas sur la console.

2. La page Premiers pas avec Device Advisor fournit une vue d'ensemble 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 ici. Vous devez configurer le microprogramme ou le logiciel de l'appareil utilisé pour les tests afin de vous connecter à ce point de terminaison de test.

Pour effectuer ce didacticiel, vous devez d'abord [créer un objet et un certificat](#). Après avoir examiné les informations de la section Fonctionnement, choisissez Suivant.

3. À l'étape 1 : Sélectionnez un protocole, sélectionnez un protocole parmi les options répertoriées. Ensuite, choisissez Next (Suivant).

- À l'étape 2, vous créez et configurez 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 scénario de test. Nous avons ajouté le scénario de test MQTT Connect pour vous aider à démarrer.

Choisissez Propriétés de la suite de tests.

The screenshot shows the 'Create test suite' wizard in the AWS IoT Device Advisor. The left sidebar shows 'Monitor', 'Connect', 'Test' (selected), and 'Manage' sections. The main area shows 'Step 2: Create test suite' with a sub-step 'Select a protocol'. Below it is 'Step 3: Configure device settings' and 'Step 4: Review'. The 'Test cases' section lists 14 MQTT test cases. On the right, there's a 'Start' section with instructions about the execution order, a 'Configure' section with a link to 'Test suite properties', and a 'Test group 1' section with an 'Edit' button.

Vous devez fournir les propriétés de la suite de tests lorsque vous créez votre suite de tests. Vous pouvez configurer les propriétés suivantes au niveau de la suite :

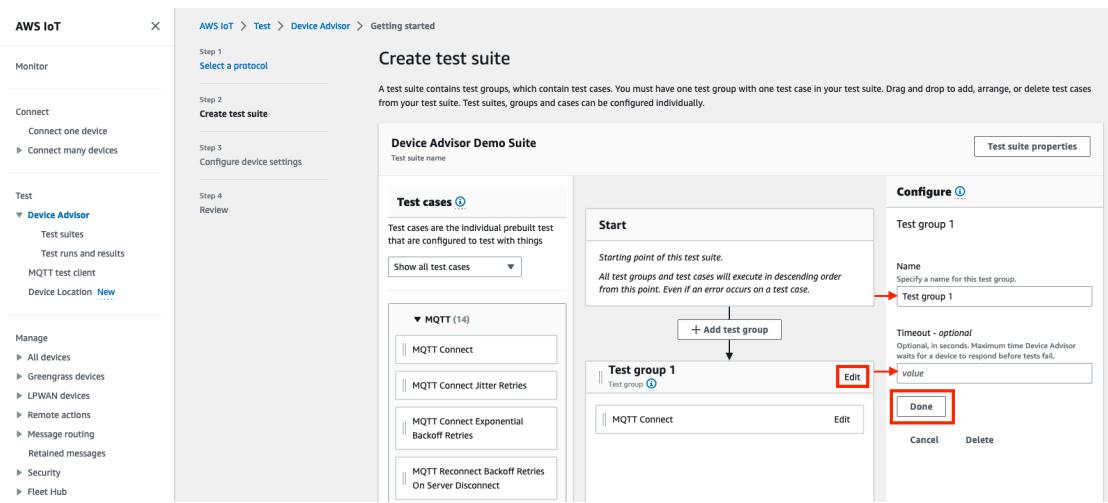
- Nom de la suite de tests : Entrez le nom de votre suite de tests.
- Délai d'expiration (facultatif) : délai d'expiration (en secondes) pour chaque scénario de test de la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.
- Balises (facultatives) : ajoutez des balises à la suite de tests.

Lorsque vous avez terminé, choisissez Mettre à jour les propriétés.

The dialog box is titled 'Test suite properties'. It contains a 'Test suite name' field with 'Device Advisor Demo Suite', a 'Timeout - optional' field with '300', and a 'Tags' section with an 'Add new tag' button. At the bottom are 'Cancel' and 'Update properties' buttons, with 'Update properties' highlighted by a red box.

- (Facultatif) Pour mettre à jour la configuration du groupe de suites de tests, cliquez sur le bouton Modifier à côté du nom du groupe de tests.
- Nom : Entrez un nom personnalisé pour le groupe de suites de tests.
 - Délai d'expiration (facultatif) : délai d'expiration (en secondes) pour chaque scénario de test de la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.

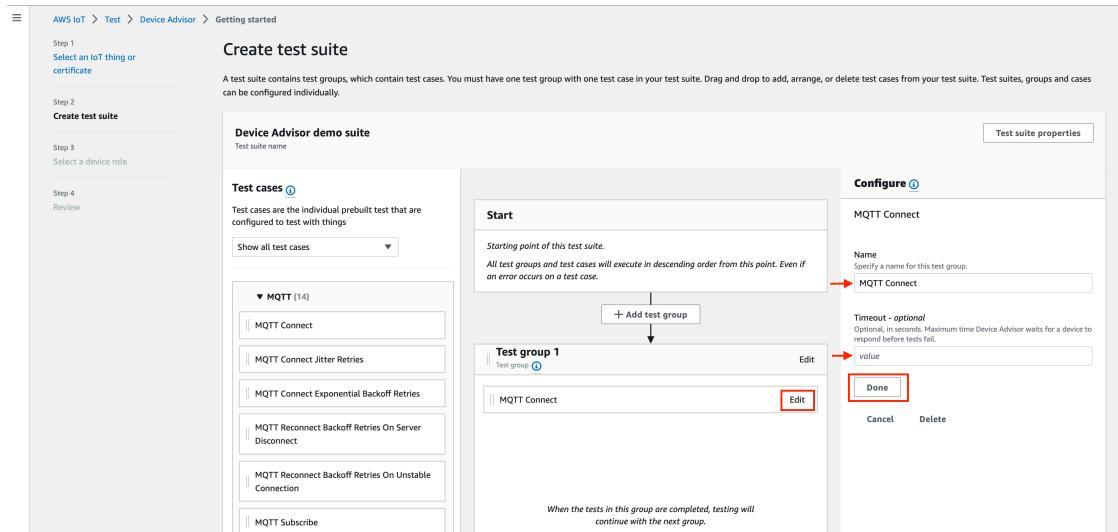
Lorsque vous avez terminé, choisissez OK pour continuer.



- (Facultatif) Pour mettre à jour la configuration d'un scénario de test, cliquez sur le bouton Modifier à côté du nom du scénario de test.

- Nom : Entrez un nom personnalisé pour le groupe de suites de tests.
- Délai d'expiration (facultatif) : délai d'expiration (en secondes) pour le scénario de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.

Lorsque vous avez terminé, choisissez OK pour continuer.



- (Facultatif) Pour ajouter d'autres groupes de test à la suite de tests, choisissez Ajouter un groupe de test, puis suivez les instructions de l'étape 5.
- (Facultatif) Pour ajouter d'autres scénarios de test, faites glisser les cas de test de la section Cas de test vers l'un de vos groupes de tests.

AWS IoT Core Guide du développeur

Commencer à utiliser Device Advisor dans la console

9. Vous pouvez modifier l'ordre de vos groupes de test et de vos scénarios de test. Pour apporter des modifications, faites glisser les scénarios de test vers le haut ou vers le bas de la liste. Device Advisor exécute les tests dans l'ordre dans lequel vous les avez répertoriés.

Après avoir configuré votre suite de tests, choisissez Suivant.

10. À l'étape 3, sélectionnez un AWS IoT objet ou un certificat à tester à l'aide de Device Advisor. Si vous ne possédez aucun objet ou certificat, consultez [Configuration](#).

11. Vous pouvez configurer un rôle d'appareil que Device Advisor utilise pour effectuer des actions AWS IoT MQTT pour le compte de votre appareil de test. Pour le scénario de test MQTT Connect uniquement, l'action Connect est sélectionnée automatiquement. Cela est dû au fait que le rôle de l'appareil nécessite cette autorisation pour exécuter la suite de tests. Pour les autres cas de test, les actions correspondantes sont sélectionnées.

Fournissez les valeurs des ressources pour chacune des actions sélectionnées. Par exemple, pour l'action Connect, indiquez l'ID client que votre appareil utilise pour se connecter au point de terminaison Device Advisor. Vous pouvez fournir plusieurs valeurs séparées par des virgules et les préfixer par un caractère générique (*). Par exemple, pour autoriser la publication sur n'importe quel sujet commençant par MyTopic, entrez **MyTopic*** comme valeur de ressource.

Pour utiliser un rôle d'appareil créé précédemment dans [Configuration](#), choisissez Sélectionner un rôle existant. Choisissez ensuite le rôle de votre appareil sous Sélectionner un rôle.

Configurez le rôle de votre appareil à l'aide de l'une des deux options proposées, puis choisissez Suivant.

12. Dans la section Point de terminaison de test, choisissez le point de terminaison le mieux adapté à votre cas d'utilisation. Pour exécuter simultanément plusieurs suites de testsCompte AWS, sélectionnez le point de terminaison au niveau de l'appareil. Pour exécuter une suite de tests à la fois, sélectionnez Point de terminaison au niveau du compte.

13. L'étape 4 présente un aperçu du dispositif de test sélectionné, du point de terminaison du test, de la suite de tests et du rôle de l'appareil de test configuré. Pour apporter des modifications à une section, cliquez sur le bouton Modifier correspondant à la section que vous souhaitez modifier. Une fois que vous avez confirmé votre configuration de test, choisissez Exécuter pour créer la suite de tests et exécuter vos tests.

Note

Pour de meilleurs résultats, vous pouvez connecter l'appareil de test que vous avez sélectionné au point de terminaison de test Device Advisor avant de démarrer l'exécution de la suite de tests. Nous vous recommandons de créer un mécanisme permettant à votre appareil d'essayer de se connecter à notre point de terminaison de test toutes les cinq secondes pendant une à deux minutes au maximum.

AWS IoT Core Guide du développeur

Commencer à utiliser Device Advisor dans la console

AWS IoT > Test > Device Advisor > Getting started

Review

Step 1: Select a protocol

Step 2: Create test suite

Step 3: Configure device settings

Step 4: Review

Test suite type

Test suite type: Custom test suite | Protocol: MQTT 3.1.1

Test suite details

Test suite name: Device Advisor Demo Suite | Suite version: v1 | Test type: Custom test suite

Start

Starting point of this test suite.

Test group 1

MQTT Connect

When the tests in this group are completed, testing will continue with the next group.

End

End point of this test suite.

Step 3: Configure device settings

Device role details

Device: MyThing | Thing ID: redacted | Thing name: MyThing | Thing ARN: redacted

Device role type: Create new role | Device role name: DeviceAdvisorServiceRole

Test endpoint

redacted.amazonaws.com

Cancel Previous Run

14. Dans le volet de navigation, sous Test, choisissez Device Advisor, puis choisissez Test runs and results. Sélectionnez l'exécution d'une suite de tests pour afficher les détails et les journaux de son exécution.

AWS IoT > Device Advisor > Test suites > Device Advisor Demo Suite > March 22, 2023, 11:20:48 (UTC-0700)

Connect your device now

Connect your device to the Device Advisor test endpoint: redacted.amazonaws.com now to validate your device for MQTT Connect. For more information, refer to [Configure your test device](#).

Last updated: 11:21:43 (UTC-0700). Auto-refreshes every 10 seconds

Test suite log Actions

March 22, 2023, 11:20:48 (UTC-0700)

Summary

Device	Protocol	Suite version	Created	Status
MyThing	MQTT 3.1.1	v1	March 22, 2023, 11:20:48 (UTC-0700)	In Progress

Test group 1 (1)

Test	Result	System message	Logs
MQTT Connect	In Progress		

Tags (0)

Tags are metadata that you can assign to AWS resources. Each tag consists of a key and an optional value. You can use tags to search and filter test suites.

Key	Value
	No tags
	No tags associated with the resource.

Manage tags

15. Pour accéder aux CloudWatch journaux Amazon de la suite, exécutez :

- Choisissez Journal de la suite de tests pour afficher les CloudWatch journaux de l'exécution de la suite de tests.
 - Choisissez Journal des cas de test pour n'importe quel scénario de test pour afficher les CloudWatch journaux spécifiques au cas de test.
16. En fonction des résultats de vos tests, [dépannez votre appareil jusqu'à](#) ce que tous les tests soient réussis.

Flux de Device Advisor

Ce didacticiel explique comment créer une suite de tests personnalisée et exécuter des tests sur l'appareil que vous souhaitez tester dans la console. Une fois les tests terminés, vous pouvez consulter les résultats des tests et les journaux détaillés.

Prérequis

Avant de commencer ce didacticiel, suivez les étapes décrites dans[Configuration \(p. 1168\)](#).

Création d'une définition de suite de test

Tout d'abord, [installez un AWS SDK](#).

Syntaxe de rootGroup

Un groupe racine est une chaîne JSON qui spécifie les cas de test à inclure dans votre suite de tests. Il spécifie également toutes les configurations nécessaires pour ces scénarios de test. Utilisez le groupe racine pour structurer et organiser votre suite de tests en fonction de vos besoins. La hiérarchie d'une suite de tests est la suivante :

```
test suite # test group(s) # test case(s)
```

Une suite de tests doit comporter au moins un groupe de tests, et chaque groupe de tests doit comporter au moins un scénario de test. Device Advisor exécute les tests dans l'ordre dans lequel vous définissez les groupes de tests et les cas de test.

Chaque groupe de racines suit 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": ""
          }
        }
      ]
    }
  ]
}
```

```

        }]
    }
}
```

Dans le groupe racine, vous définissez la suite de tests avec un `name` configuration, et celle `tests` que contient le groupe. Le `tests` groupe contient les définitions des tests individuels. Vous définissez chaque test à l'aide d'un `name` configuration, et d'un `test` bloc qui définit les cas de test pour ce test. Enfin, chaque scénario de test est défini par un `id` et `version`.

Pour plus d'informations sur l'utilisation des "version" champs "`id`" et pour chaque scénario de test (`test` bloc), reportez-vous à la section [Scénarios de test de Device Advisor \(p. 1203\)](#). Cette section contient également des informations sur les configurations paramètres disponibles.

Le bloc suivant est un exemple de configuration de groupe racine. Cette configuration décrit les scénarios 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 votre suite de tests. Enregistrez `suiteDefinitionId` ce qui est renvoyé dans l'objet de réponse. Vous pouvez utiliser cet ID pour récupérer les informations de définition de votre suite de tests et 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")  
        .deviceRoleArn("your-device-role-arn") //if using SigV4 for MQTT  
    over WebSocket  
        .build()  
    )  
    .rootGroup("your-root-group-configuration")  
    .devicePermissionRoleArn("your-device-permission-role-arn")  
    .protocol("MqttV3_1_1 || MqttV5 || MqttV3_1_1_OverWebSocket ||  
MqttV5_OverWebSocket")  
    .build()  
)  
.build()
```

Obtenir une définition de la suite de tests

Une fois que vous avez créé la définition de votre suite de tests, vous recevez l'suiteDefinitionId objet de réponse de l'opération d'CreateSuiteDefinitionAPI.

Lorsque l'opération renvoie lesuiteDefinitionId, vous pouvez voir de nouveaux id champs dans chaque groupe et une définition de scénario de test au sein du groupe racine. Vous pouvez utiliser ces ID pour exécuter un sous-ensemble de la définition de votre suite de tests.

Exemple de SDK Java :

```
response = iotDeviceAdvisorClient.GetSuiteDefinition(  
    GetSuiteDefinitionRequest.builder()  
        .suiteDefinitionId("your-suite-definition-id")  
    .build()  
)
```

Obtenir un point de terminaison de test

Utilisez l'opération GetEndpoint API pour obtenir le point de terminaison de test utilisé par votre appareil. Sélectionnez le point de terminaison qui correspond le mieux à votre test. Pour exécuter simultanément plusieurs suites de tests, utilisez le point de terminaison au niveau du périphérique en fournissant un thing ARNcertificate ARN, ou. device role ARN Pour exécuter une seule suite de tests, ne fournissez aucun argument à l'GetEndpointopération visant à choisir le point de terminaison au niveau du compte.

Exemple de kit SDK :

```
response = iotDeviceAdvisorClient.getEndpoint(GetEndpointRequest.builder()  
    .certificateArn("your-test-device-certificate-arn")  
    .thingArn("your-test-device-thing-arn")  
    .deviceRoleArn("your-device-role-arn") //if using SigV4 for MQTT over WebSocket  
    .build())
```

Lancer l'exécution d'une suite de tests

Après avoir créé une définition de suite de tests et configuré votre appareil de test pour qu'il se connecte à votre point de terminaison de test Device Advisor, exécutez votre suite de tests avec l'StartSuiteRunAPI.

Pour les clients MQTT, utilisez l'une `certificateArn` ou l'autre `thingArn` pour exécuter la suite de tests. Si les deux sont configurés, le certificat est utilisé s'il appartient à l'objet.

Pour MQTT plutôt que pour WebSocket le client, `deviceRoleArn` utilisez-le pour exécuter la suite de tests. Si le rôle spécifié est différent du rôle spécifié dans la définition de la suite de tests, le rôle spécifié remplace le rôle défini.

À `.parallelRun()` utiliser `true` si vous utilisez un point de terminaison au niveau de l'appareil pour exécuter plusieurs suites de tests en parallèle à l'aide d'une seule. Compte AWS

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")
            .deviceRoleArn("your-device-role-arn") //if using SigV4 for MQTT over WebSocket
        )
        .build())
    .parallelRun(true | false)
    .build())
.build()
```

Enregistrez le `suiteRunId` à partir de la réponse. Vous allez l'utiliser pour récupérer les résultats de cette suite de tests exécutée.

Lancer une suite de tests

Après avoir lancé l'exécution d'une suite de tests, vous pouvez vérifier sa progression et ses résultats à l'aide de l'`GetSuiteRunAPI`.

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

Pour arrêter l'exécution d'une suite de tests qui est toujours en cours, vous pouvez appeler l'opération `StopSuiteRun API`. Une fois que vous avez appelé l'`StopSuiteRun` opération, le service lance le processus de nettoyage. Pendant que le service exécute le processus de nettoyage, la suite de tests exécute les mises à jour de statut `deStopping`. Le processus de nettoyage peut prendre plusieurs minutes. Une fois le processus terminé, la suite de tests exécute les mises à jour de statut `versStopped`. Une fois qu'une exécution de test est complètement arrêtée, vous pouvez démarrer une autre exécution de la suite de tests. Vous pouvez vérifier régulièrement l'état d'exécution de la suite à l'aide de l'opération `GetSuiteRun API`, comme 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())
```

Obtenez un rapport de qualification pour une exécution réussie de la suite de tests de qualification

Si vous exécutez une suite de tests de qualification qui aboutit, vous pouvez récupérer un rapport de qualification à l'aide de l'opération GetSuiteRunReport API. Vous utilisez ce rapport de qualification pour qualifier votre appareil dans le cadre du programme de AWS IoT Core qualification. Pour déterminer si votre suite de tests est une suite de tests de qualification, vérifiez si le intendedForQualification paramètre est défini sur true. Après avoir appelé l'opération GetSuiteRunReport d'API, vous pouvez télécharger le rapport à partir de l'URL renvoyée pendant 90 secondes maximum. Si plus de 90 secondes se sont écoulées depuis la dernière fois que vous avez appelé l'GetSuiteRunReport opération,appelez l'opération pour récupérer une nouvelle 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()  
)
```

Flux de travail détaillé de Device Advisor sur console

Dans ce didacticiel, vous allez créer une suite de tests personnalisée et exécuter des tests sur l'appareil que vous souhaitez tester dans la console. Une fois les tests terminés, vous pouvez consulter les résultats des tests et les journaux détaillés.

Didacticiels

- [Prérequis \(p. 1184\)](#)
- [Création d'une définition de suite de test \(p. 1185\)](#)
- [Lancer l'exécution d'une suite de tests \(p. 1190\)](#)
- [Arrêter l'exécution d'une suite de tests \(facultatif\) \(p. 1191\)](#)
- [Afficher les détails et les journaux d'exécution de la suite de tests \(p. 1192\)](#)
- [Télécharger un rapport AWS IoT de qualification \(p. 1193\)](#)

Prérequis

Pour effectuer ce didacticiel, vous devez [créer un objet et un certificat](#).

Création d'une définition de suite de test

- Dans la [AWS IoT console](#), dans le volet de navigation, développez Test, Device Advisor, puis choisissez Suites de tests.

Choisissez Create Test Suite.

- Sélectionnez Use the AWS Qualification test suite ou Create a new test suite.

Pour le protocole, choisissez MQTT 3.1.1 ou MQTT 5.

Sélectionnez cette option Use the AWS Qualification test suite pour qualifier votre appareil et le répertorier dans le catalogue des appareils AWS partenaires. En choisissant cette option, les scénarios de test requis pour la qualification de votre appareil dans le cadre du programme de AWS IoT Core qualification sont présélectionnés. Les groupes de test et les cas de test ne peuvent être ni ajoutés ni supprimés. Vous devez toujours configurer les propriétés de la suite de tests.

Sélectionnez Create a new test suite cette option pour 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 scénario de test. Dans le cadre de ce didacticiel, nous allons sélectionner cette option et choisir Suivant.

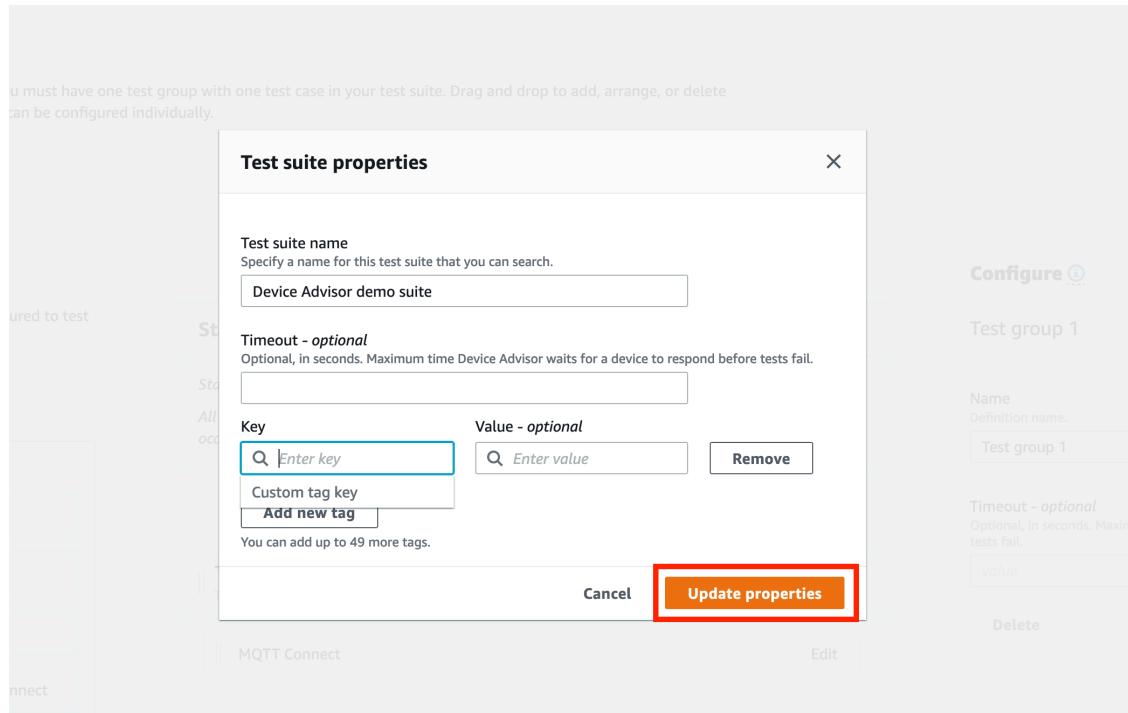
The screenshot shows the 'Configure test suite' step of the 'Create test suite' wizard. On the left, the navigation bar includes 'Monitor', 'Connect', 'Test' (selected), 'Device Advisor' (selected), and 'Device Software'. The 'Test' section under 'Device Advisor' lists 'Test suites', 'Test runs and results', 'MQTT test client', and 'Device Location'. The main area shows 'Step 1 Create test suite' and 'Step 2 Configure test suite'. The 'Configure test suite' panel displays a 'Test suite December 22, 2022, 11:24:37 (UTC-0800)' with a 'Test suite name' field. The 'Test cases' section lists 14 MQTT test cases: 'MQTT Connect', 'MQTT Connect Jitter Retries', 'MQTT Connect Exponential Backoff Retries', 'MQTT Reconnect Backoff Retries On Server Disconnect', and 'MQTT Reconnect Backoff Retries On Unstable Connection'. The 'Start' section specifies the 'Starting point of this test suite' and notes that 'All test groups and test cases will execute in descending order from this point. Even if an error occurs on a test case.' The 'Configure' section on the right is titled 'Configure' and says 'Select a test group or test case to configure it.' A 'Test suite properties' button is also visible.

- Choisissez Propriétés de la suite de tests. Vous devez créer les propriétés de la suite de tests lorsque vous créez votre suite de tests.

This screenshot is identical to the previous one, showing the 'Configure test suite' step of the 'Create test suite' wizard. The 'Test cases' section lists the same 14 MQTT test cases. The 'Start' section and 'Configure' section are also identical. However, the 'Test suite properties' button in the 'Configure' section is now highlighted with a red box.

Sous Propriétés de la suite de tests, renseignez les informations suivantes :

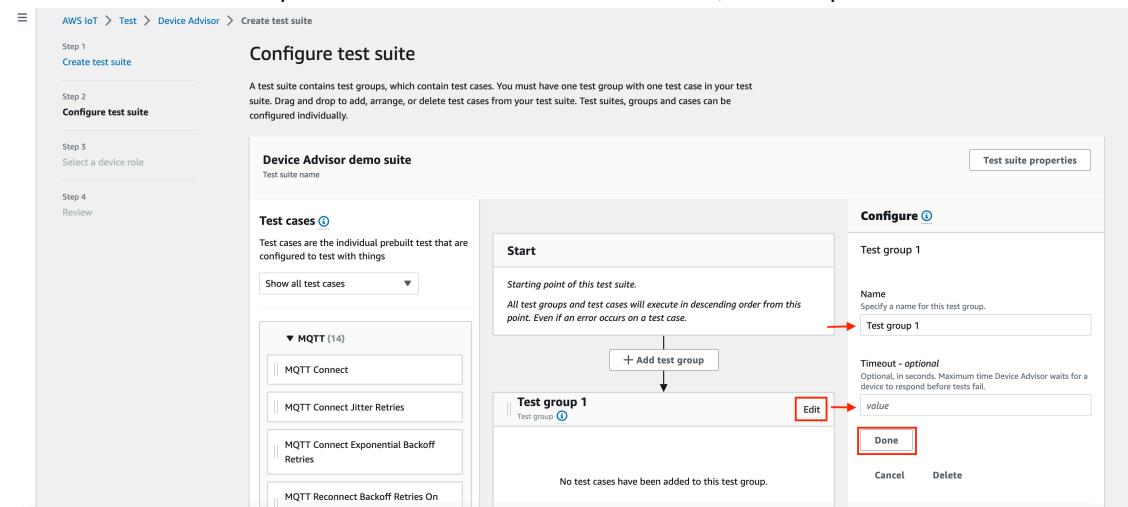
- Nom de la suite de tests : vous pouvez créer la suite avec un nom personnalisé.
- Délai d'expiration (facultatif) : délai d'expiration en secondes pour chaque scénario de test de la suite de tests en cours. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.
- Balises (facultatives) : ajoutez des balises à la suite de tests.



Lorsque vous avez terminé, choisissez Mettre à jour les propriétés.

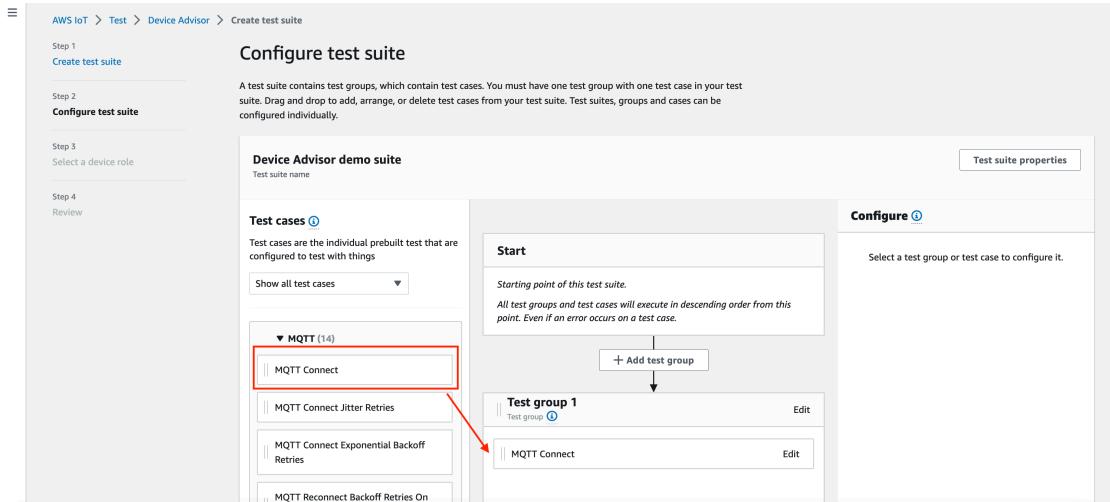
- Pour modifier la configuration au niveau du groupeTest group 1, cliquez sur Modifier. Entrez ensuite un nom pour attribuer un nom personnalisé au groupe.

Vous pouvez également saisir une valeur de délai d'expiration en secondes pour le groupe de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.



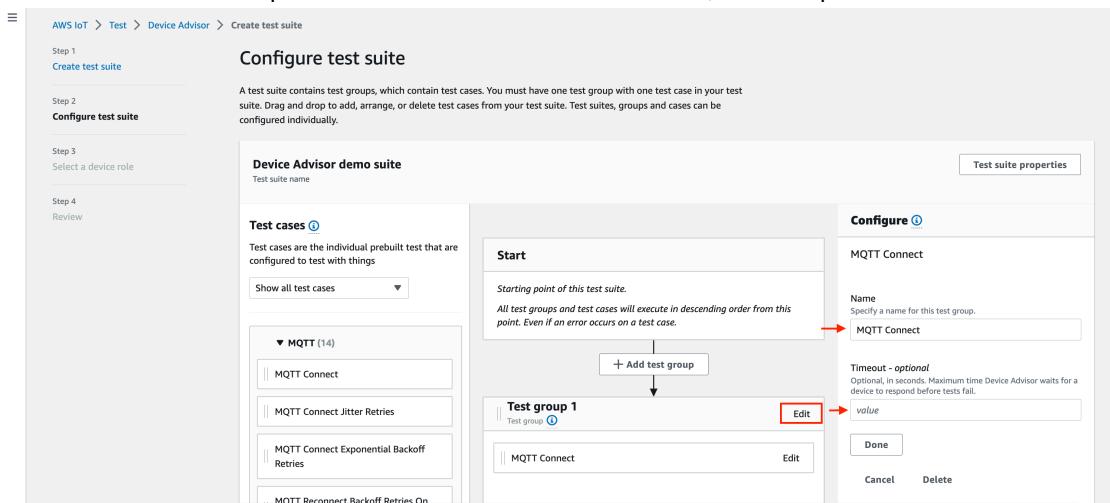
Sélectionnez Done (Exécuté).

- Faites glisser l'un des scénarios de test disponibles depuis Cas de test vers le groupe de tests.



- Pour modifier la configuration au niveau du scénario de test pour le scénario de test que vous avez ajouté à votre groupe de test, choisissez Modifier. Entrez ensuite un nom pour attribuer un nom personnalisé au groupe.

Vous pouvez également saisir une valeur de délai d'expiration en secondes pour le groupe de test sélectionné. Si vous ne spécifiez aucune valeur de délai d'attente, la valeur par défaut est utilisée.



Sélectionnez Done (Exécuté).

Note

Pour ajouter d'autres groupes de test à la suite de tests, choisissez Ajouter un groupe de 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 tests. Les groupes de test et les cas de test peuvent être réorganisés en choisissant un cas de test et en le faisant glisser vers la position souhaitée. Device Advisor exécute les tests dans l'ordre dans lequel vous définissez les groupes de tests et les cas de test.

- Choisissez Suivant.
- À l'étape 3, configurez un rôle d'appareil que Device Advisor utilisera pour effectuer des actions AWS IoT MQTT pour le compte de votre appareil de test.

Si vous avez sélectionné le scénario de test MQTT Connect uniquement à l'étape 2, l'action Connect sera vérifiée automatiquement car cette autorisation est requise sur le rôle de l'appareil pour exécuter

cette suite de tests. Si vous avez sélectionné d'autres scénarios de test, les actions requises correspondantes seront vérifiées. Assurez-vous que les valeurs des ressources pour chacune des actions sont fournies. Par exemple, pour l'action Connect, indiquez l'identifiant client avec lequel votre appareil se connectera au point de terminaison Device Advisor. Vous pouvez fournir plusieurs valeurs en les séparant par des virgules. Vous pouvez également fournir des préfixes à l'aide d'un caractère générique (*). Par exemple, pour autoriser la publication sur n'importe quel sujet commençant par `MyTopic`, vous pouvez fournir « `MyTopic*` » comme valeur de ressource.

AWS IoT > Test > Device Advisor > Create test suite

Step 1 Create test suite

Step 2 Configure test suite

Step 3 Select a device role

Step 4 Review

Select a device role

Device role info
AWS IoT Core Device Advisor requires permission to perform AWS IoT MQTT actions on behalf of your test device.

Create new role
Create and use a new device role

Select an existing role
Use an existing device role

Role name
MyDevicevisorDeviceRole

Permissions info
Choose which actions and the associated resources for AWS IoT Core Device Advisor to access using this role. You can enter a specific resource or resource prefix. To enter multiple values for a resource, use commas to separate the values. [Learn more](#)

Action	Resource type	Resource
<input checked="" type="checkbox"/> Connect	ClientId	MyClient
<input type="checkbox"/> Publish	Topic	Specify topics to publish to, e.g. <code>MyTopic, MyTopic*</code>
<input type="checkbox"/> Subscribe	TopicFilter	Specify topic filters to subscribe to, e.g. <code>MyTopic, MyTopic*</code>
<input type="checkbox"/> Receive	Topic	Specify topics to receive from e.g. <code>MyTopic, MyTopic*</code>
<input type="checkbox"/> RetainPublish	Topic	Specify topics to publish a retained message to, e.g. <code>MyTopic, MyTopic*</code>

Cancel Previous Next

Si vous avez déjà créé un rôle sur un appareil et que vous souhaitez l'utiliser, sélectionnez Sélectionner un rôle existant et choisissez votre rôle sur l'appareil sous Sélectionner un rôle.

AWS IoT > Test > Device Advisor > Create test suite

Step 1 Create test suite

Step 2 Configure test suite

Step 3 Select a device role

Step 4 Review

Select a device role

Device role info
AWS IoT Core Device Advisor requires permission to perform AWS IoT MQTT actions on behalf of your test device.

Create new role
Create and use a new device role

Select an existing role
Use an existing device role

Select role
Select a device role

Cancel Previous Next

Configurez le rôle de votre appareil à l'aide de l'une des deux options proposées et choisissez Suivant.

- À l'étape 4, assurez-vous que la configuration fournie à chacune des étapes est précise. Pour modifier la configuration fournie pour une étape particulière, choisissez Modifier pour l'étape correspondante.

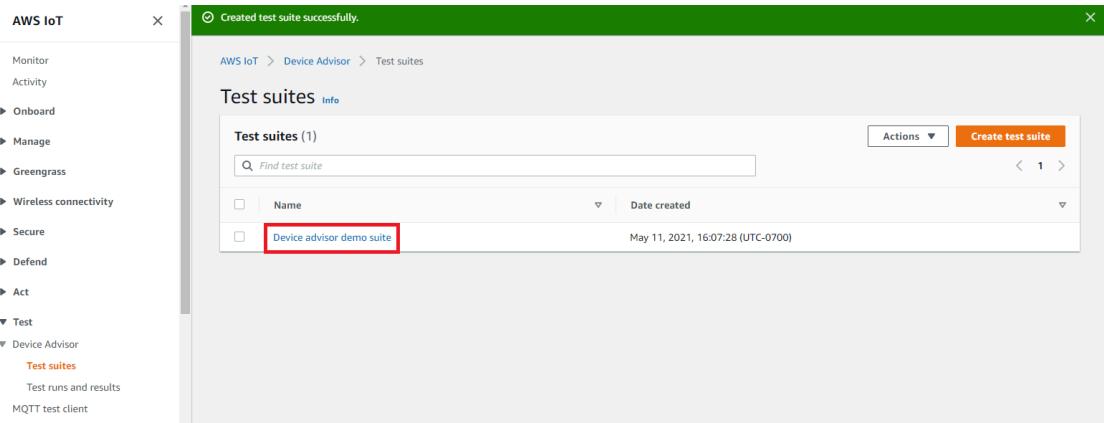
Après avoir vérifié la configuration, choisissez Create test suite.

La suite de tests doit être créée avec succès et vous serez redirigé vers la page Suites de tests où vous pouvez voir 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, les scénarios de test et le rôle de l'appareil ont été configurés conformément aux instructions précédentes.

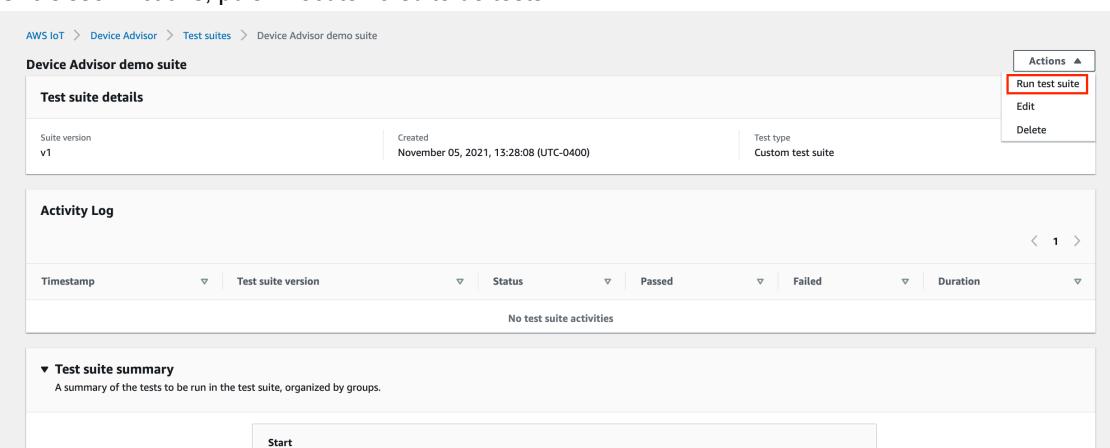
Lancer l'exécution d'une suite de tests

1. Dans la [AWS IoT console](#), dans le volet de navigation, développez Test, Device Advisor, puis choisissez Suites de tests.
2. Choisissez la suite de tests pour laquelle vous souhaitez consulter 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.

3. Choisissez Actions, puis Exécuter la suite de tests.



4. Sous Exécuter la configuration, vous devez sélectionner un AWS IoT élément ou un certificat à tester à l'aide de Device Advisor. Si vous n'avez aucun objet ou certificat existant, [créez d'abord des AWS IoT Core ressources \(p. 1168\)](#).

Dans la section Point de terminaison du test, sélectionnez le point de terminaison qui correspond le mieux à votre cas. Si vous prévoyez d'exécuter plusieurs suites de tests simultanément à l'aide du même AWS compte à l'future, sélectionnez Device-level Endpoint. Sinon, si vous prévoyez de n'exécuter qu'une seule suite de tests à la fois, sélectionnez Point de terminaison au niveau du compte.

Configurez votre appareil de test avec le point de terminaison de test du Device Advisor sélectionné.

Après avoir sélectionné un objet ou un certificat et choisi un point de terminaison Device Advisor, choisissez Exécuter le test.

Run configuration

Select test devices

Select the IoT thing/certificate to test using the test suite. If not listed below, you must first create a thing/certificate registered with IoT Core before you can run the test suite.

Things Choose a thing for this test suite. To create a new thing, go to IoT Things [\[?\]](#)

Certificates Choose a certificate for this test suite. To create a new certificate, go to IoT Certificates [\[?\]](#)

Things (1)

Filter things

Name Type

MyThing

Test endpoint

Choose the endpoint that best fits your situation. If you want to simultaneously run multiple test suites then use 'Device-level endpoint'; if you want to run only one test suite at a time then choose the 'Account-level endpoint'.

Account-level endpoint Using this endpoint, you can only run one test suite at a time.

Device-level endpoint Copy and paste this endpoint to your test device. 98d6c41394d1959c92a.gammas.us-west-2.advisor.iot.awss [\[?\]](#)

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

Add new tag You can add up to 50 more tags.

Cancel Run test

5. Choisissez Accéder aux résultats dans la bannière supérieure pour afficher les détails du test.

'Device Advisor demo suite' is in progress with 'MyThing'. [Go to results](#) [X](#)

AWS IoT > Device Advisor > Test suites > Device Advisor demo suite

Device Advisor demo suite

Test suite details

Suite version v1	Created November 05, 2021, 13:40:33 (UTC-0400)	Test type Custom test suite
---------------------	---	--------------------------------

Activity Log

Timestamp	Test suite version	Status	Passed	Failed	Duration
November 05, 2021, 13:53:23 (UTC-0400)	v1	Pending	-	-	-

Arrêter l'exécution d'une suite de tests (facultatif)

1. Dans la [AWS IoTconsole](#), dans le volet de navigation, développez Test, Device Advisor, puis choisissez Test runs and results.
2. Choisissez la suite de tests en cours que vous voulez arrêter.

AWS IoT

Monitor Activity

Onboard

Manage

- Things
- Types
- Thing groups
- Billing groups
- Jobs
- Tunnels

Greengrass

Secure

Defend

Act

Test

Device Advisor

- Test suites
- Test runs and results [\[?\]](#)

Software

Settings

Learn

Documentation [\[?\]](#)

AWS IoT > Device Advisor > Test runs and results

Test runs and results

Summary

Number of IoT things available 1	Go to IoT things	Number of IoT certificates available 6	Go to IoT certificates	Number of test suites running 1	Go to test suites
-------------------------------------	------------------	---	------------------------	------------------------------------	-------------------

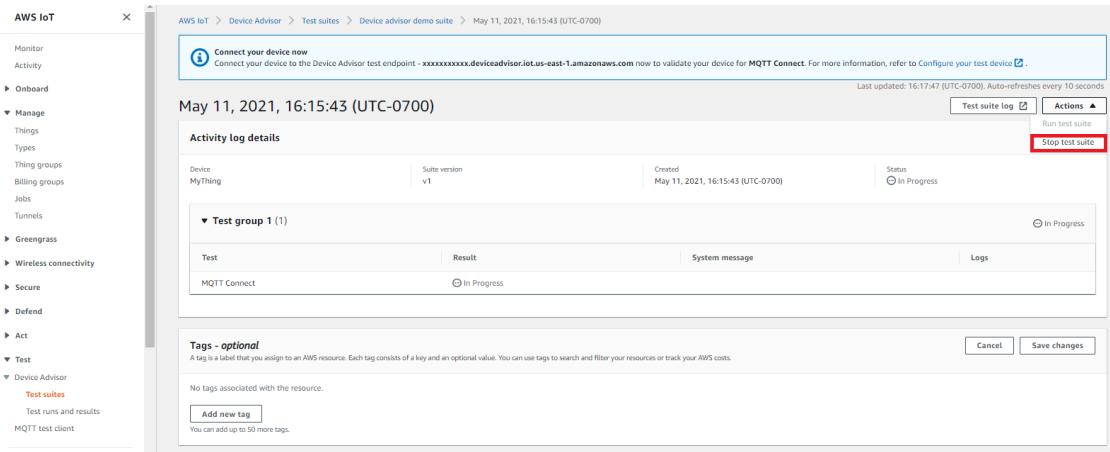
Results of test runs (in progress and completed)

Name	Timestamp	Test suite version	Status	Passed	Failed	Duration
Device Advisor demo suite	December 07, 2020, 11:16:46 (UTC-0800)	v1	In Progress	-	-	-

AWS IoT Core Guide du développeur

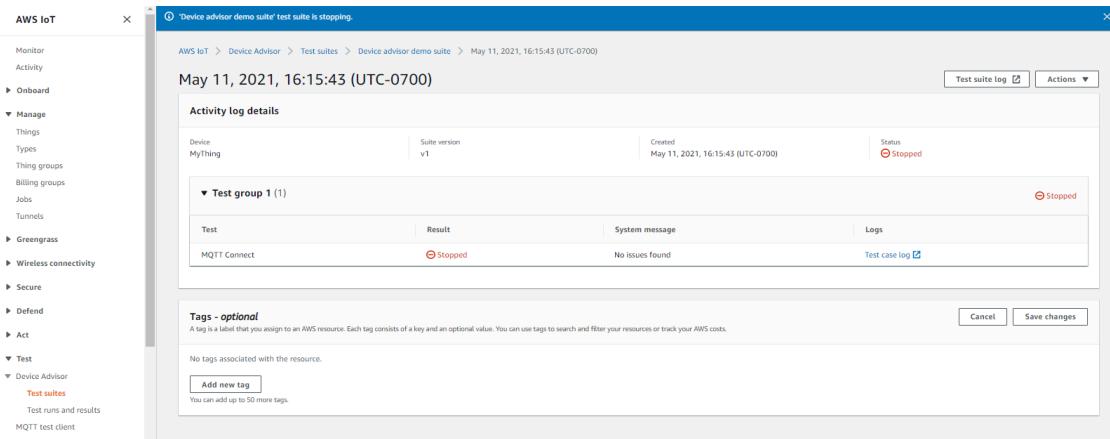
Afficher les détails et les journaux d'exécution de la suite de tests

3. Choisissez Actions, puis Arrêter la suite de tests.



The screenshot shows the AWS IoT Core Device Advisor Test Suite details page. The sidebar on the left lists various AWS services like Monitor, Activity, Onboard, Manage, Test, and Device Advisor. Under Device Advisor, 'Test suites' is selected. The main content area displays a test suite named 'Device advisor demo suite' created on May 11, 2021, at 16:15:43 (UTC-0700). The status is 'In Progress'. A 'Test group 1 (1)' is listed with a single test case 'MQTT Connect' also in progress. There is a 'Tags - optional' section where no tags are associated with the resource. Buttons for 'Test suite log' and 'Actions' (with options 'Run test suite' and 'Stop test suite') are visible.

4. L'exécution du processus de nettoyage peut prendre plusieurs minutes. Pendant l'exécution du processus de nettoyage, l'état de l'exécution du test sera STOPPING. Attendez que le processus de nettoyage soit terminé et que l'état de la suite de tests passe à l'état actuel avant de démarrer l'STOPPED exécution d'une nouvelle suite.



The screenshot shows the same AWS IoT Core Device Advisor Test Suite details page as the previous one, but the status has changed to 'Stopped'. The 'Test group 1 (1)' section now shows the 'MQTT Connect' test case as 'Stopped'. The 'System message' column indicates 'No issues found'. The 'Logs' and 'Test case log' buttons are present. The 'Actions' button still includes the option to 'Stop test suite'.

Afficher les détails et les journaux d'exécution de la suite de tests

1. Dans la [AWS IoT console](#), dans le volet de navigation, développez Test, Device Advisor, puis choisissez Test runs and results.

Cette page affiche :

- Nombre d'IoT
- Nombre de certificats IoT
- Nombre de suites de tests en cours d'exécution
- Toutes les suites de tests qui ont été créées

2. Choisissez la suite de tests pour laquelle vous souhaitez consulter les détails d'exécution et les journaux.

La page de résumé de l'exécution affiche l'état de l'exécution de la suite de tests en cours. Cette page est automatiquement actualisée toutes les 10 secondes. Nous vous recommandons de créer un mécanisme permettant à votre appareil d'essayer de se connecter à notre point de terminaison de test toutes les cinq secondes pendant une à deux minutes. Vous pouvez ensuite exécuter plusieurs scénarios de test en séquence de manière automatisée.

3. Pour accéder aux CloudWatch journaux de l'exécution de la suite de tests, choisissez Journal de la suite de tests.
Pour accéder aux CloudWatch journaux de n'importe quel scénario de test, choisissez Journal du cas de test.
4. En fonction des résultats de vos tests, dépannez votre appareil jusqu'à ce que tous les tests soient réussis.

Télécharger un rapport AWS IoT de qualification

Si vous avez choisi l'option Utiliser la suite de tests de AWS IoT qualification lors de la création d'une suite de tests et que vous avez pu exécuter une suite de tests de qualification, vous pouvez télécharger un rapport de qualification en choisissant Télécharger le rapport de qualification sur la page de résumé du test.

The screenshot shows the AWS IoT Core Qualification demo suite page. At the top, it displays the date and time: December 07, 2020, 23:33:16 (UTC-0800). Below this, the 'Activity log details' section shows a 'MyThing' device with a 'Suite revision v1' and a 'Status Passed'. The 'Qualification Program' section lists several tests: MQTT Connect, MQTT Subscribe, MQTT Publish, TLS Connect, TLS Unsigned Server Cert, and TLS Incorrect Subject Name Server Cert, all of which have passed. A 'Logs' column provides links to test case logs for each. At the bottom, there are 'Tags - optional' fields and a 'Save changes' button.

Flux de travail de console de tests de longue durée

Ce didacticiel vous aide à démarrer les tests de longue durée sur Device Advisor à l'aide de la console. Pour terminer le didacticiel, suivez les étapes décrites dans [Configuration \(p. 1168\)](#).

1. Dans le volet de navigation de la [AWS IoT console](#), développez Test, Device Advisor, puis Suites de tests. Sur la page, sélectionnez Créez une suite de tests de longue durée.

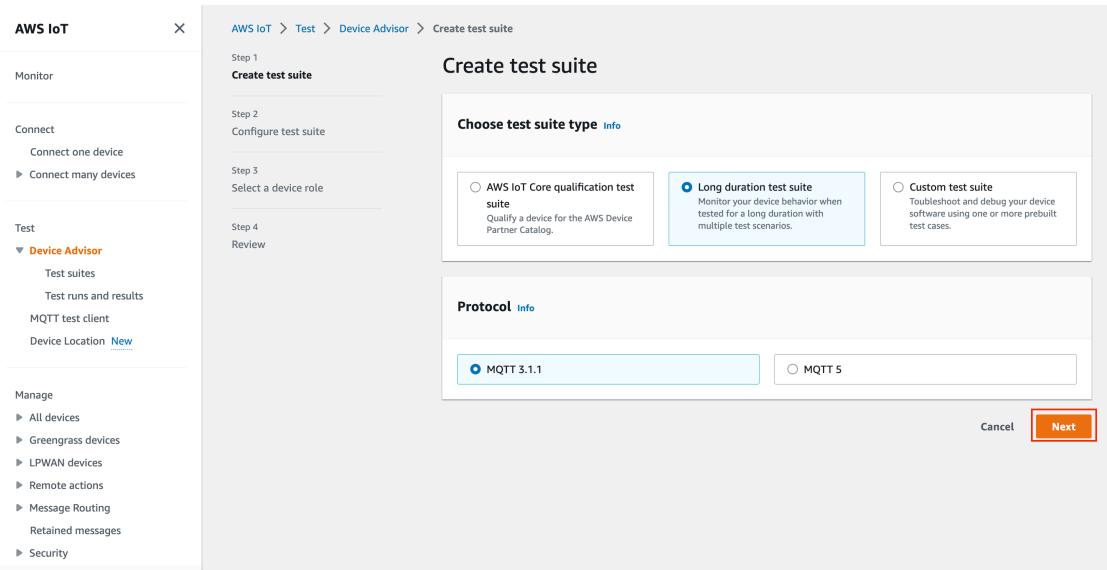
The screenshot shows the 'Create a test suite' page. It features three main sections: 'AWS IoT Core qualification test suite' (with a 'Create qualification test suite' button), 'Long duration test suite' (with a 'Create long duration test suite' button highlighted with a red box), and 'Custom test suite' (with a 'Create custom test suite' button). Below these, the 'Test suites' section shows a table with one row: 'No test suites' and a 'Create test suite' button.

2. Sur la page Créez une suite de tests, sélectionnez Suite de tests de longue durée et choisissez Suivant.

Pour le protocole, choisissez MQTT 3.1.1 ou MQTT 5.

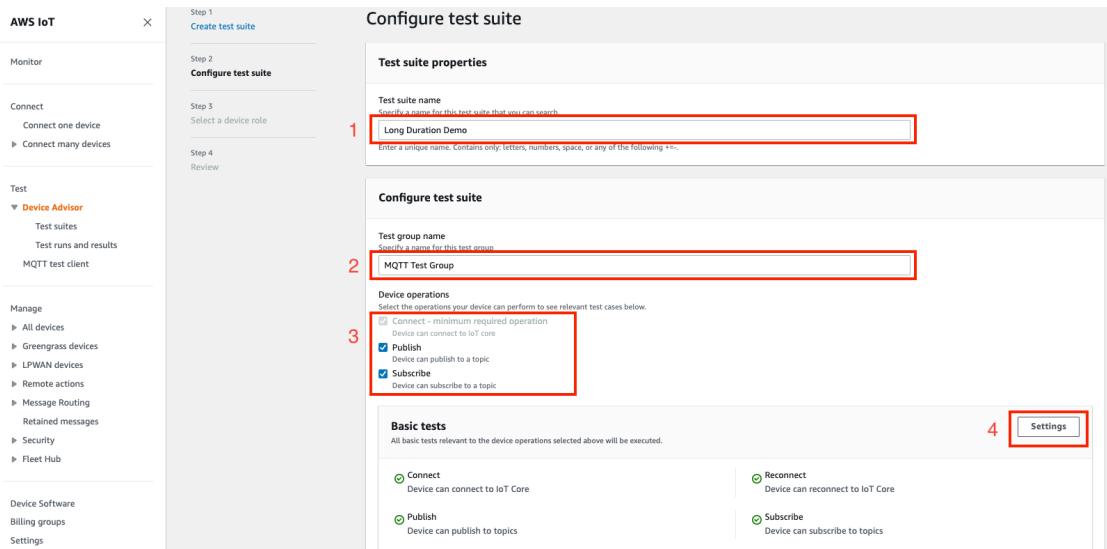
AWS IoT Core Guide du développeur

Flux de travail de console de tests de longue durée

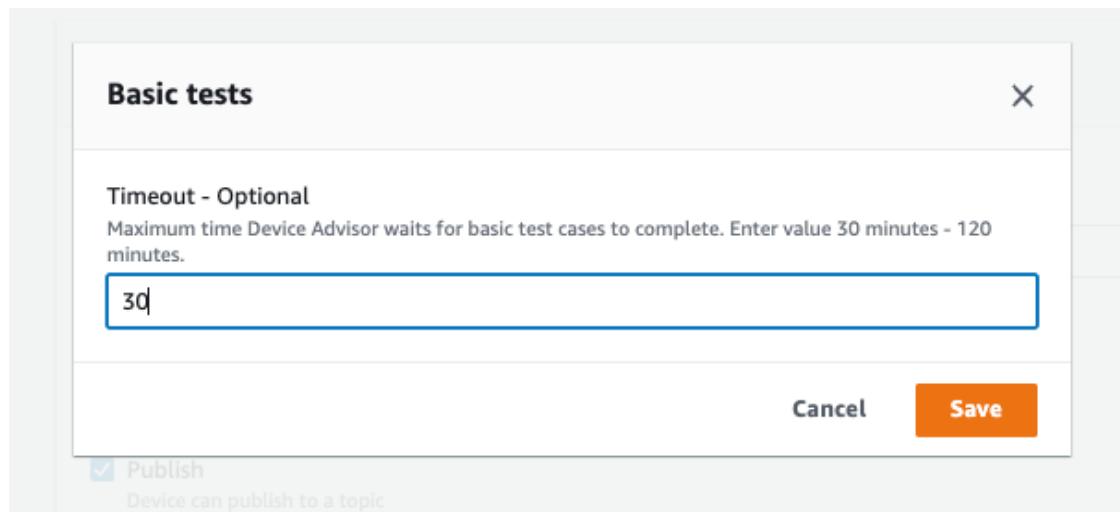


3. Procédez comme suit sur la page Configurer la suite de tests :

- a. Mettez à jour le champ Nom de la suite de tests.
- b. Mettez à jour le champ Nom du groupe de test.
- c. Choisissez les opérations que l'appareil peut effectuer. Cela sélectionnera les tests à exécuter.
- d. Sélectionnez l'option Paramètres.



4. (Facultatif) Entrez la durée maximale pendant laquelle Device Advisor doit attendre pour que les tests de base soient terminés. Sélectionnez Save.



5. Procédez comme suit dans les sections Tests avancés et Paramètres supplémentaires.
 - a. Sélectionnez ou désélectionnez les tests avancés que vous souhaitez exécuter dans le cadre de ce test.
 - b. Modifiez les configurations pour les tests, le cas échéant.
 - c. Configurez le temps d'exécution supplémentaire dans la section Paramètres supplémentaires.
 - d. Choisissez Suivant pour passer à l'étape suivante.

Basic tests
All basic tests relevant to the device operations selected above will be executed.

Test case	Description	Configure
<input checked="" type="checkbox"/> Connect	Device can connect to IoT Core	-
<input checked="" type="checkbox"/> Publish	Device can publish to topics	-
<input checked="" type="checkbox"/> Reconnect	Device can reconnect to IoT Core	-
<input checked="" type="checkbox"/> Subscribe	Device can subscribe to topics	-

Advanced tests
In addition, you can select and configure any advanced tests that you would like to execute

Test case	Description	Configure
<input checked="" type="checkbox"/> Test case	Device can connect to IoT Core	-
<input checked="" type="checkbox"/> Return PUBACK on QoS1 subscription	Device can return a PUBACK message for a message published to a subscribed QoS1 topic	-
<input checked="" type="checkbox"/> Receive large payload	Device can receive the large payload message	-
<input checked="" type="checkbox"/> Persistent session	Device can reconnect, receive stored messages and maintain a persistent session	-
<input checked="" type="checkbox"/> Keep Alive	Device can disconnect and reconnect to keep alive	-
<input checked="" type="checkbox"/> Intermittent connectivity	Device reconnects when disconnected at random intervals	-
<input checked="" type="checkbox"/> Reconnect backoff	Device has a backoff mechanism when disconnected	Edit
<input checked="" type="checkbox"/> Long server disconnect	Device reconnects when disconnected for long period	Edit

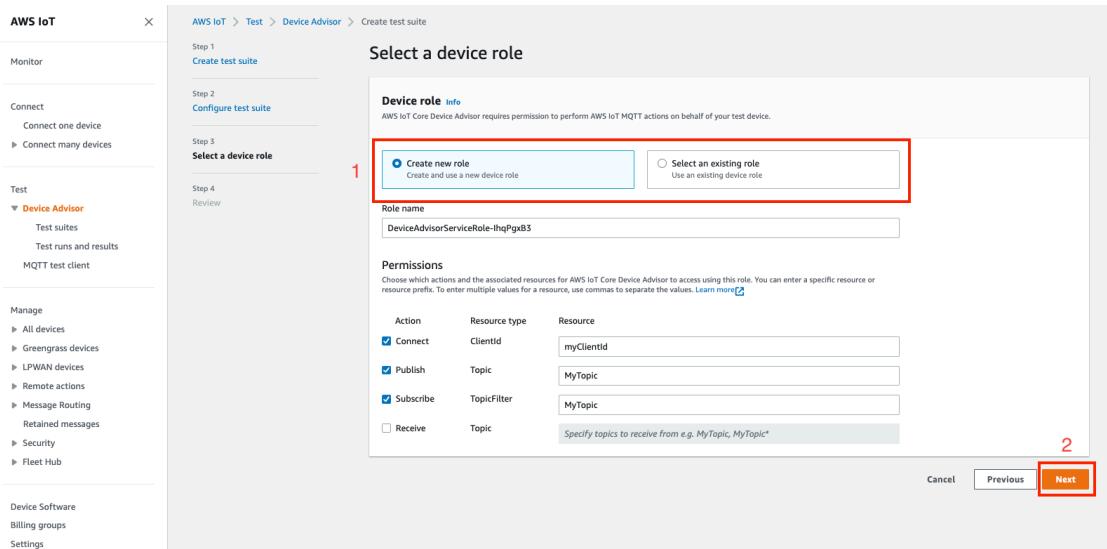
Additional settings
Additional execution time - Optional
Maximum time Device Advisor waits after completing all our test cases, before ending the test session. Enter value 0 - 120 minutes.

Cancel Previous Next

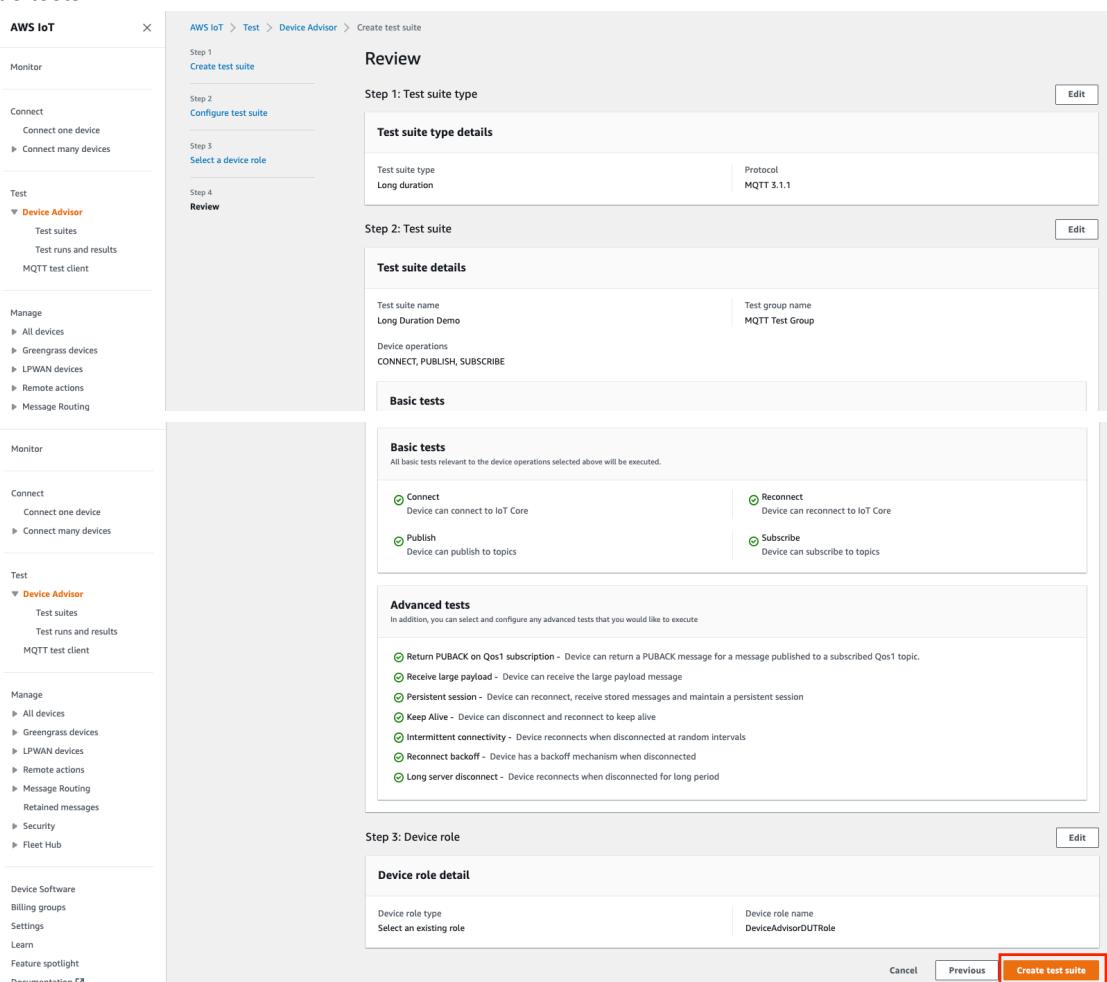
6. Au cours de cette étape, créez un nouveau rôle ou sélectionnez un rôle existant. Consultez [Créez un rôle IAM à utiliser comme rôle de votre appareil \(p. 1168\)](#) pour plus de détails.

AWS IoT Core Guide du développeur

Flux de travail de console de tests de longue durée



7. Passez en revue toutes les configurations créées jusqu'à cette étape et sélectionnez Crée une suite de tests.



8. La suite de tests créée se trouve dans la section Suites de tests. Sélectionnez la suite pour afficher les détails.

AWS IoT Core Guide du développeur

Flux de travail de console de tests de longue durée

AWS IoT Core qualification test suite
Qualify your device for inclusion in the AWS Partner Device Catalog.
[Create qualification test suite](#)

Long duration test suite
Monitor your device behavior when tested for a long duration with multiple test scenarios.
[Create long duration test suite](#)

Custom test suite
Troubleshoot and debug your device software using one or more prebuilt test cases.
[Create custom test suite](#)

Name	Test Type	Protocol	Date created
Long Duration Demo	Long duration	MQTT 3.1.1	October 12, 2022, 11:10:53 (UTC-0700)

9. Pour exécuter la suite de tests créée, sélectionnez Actions puis Exécuter la suite de tests.

Test suite details

Suite definition ARN: arn:aws:iotdeviceadvisor:ap-northeast-1:50723790144:suitedefinition/jl7u6uvtzki
Suite version: v1

Created: October 12, 2022, 11:10:53 (UTC-0700)
Test type: Long duration

Activity Log

No test suite activities

Test suite summary
A summary of the tests to be run in the test suite, organized by groups.

Test suite details

Test suite name: Long Duration Demo
Test group name: MQTT Test Group

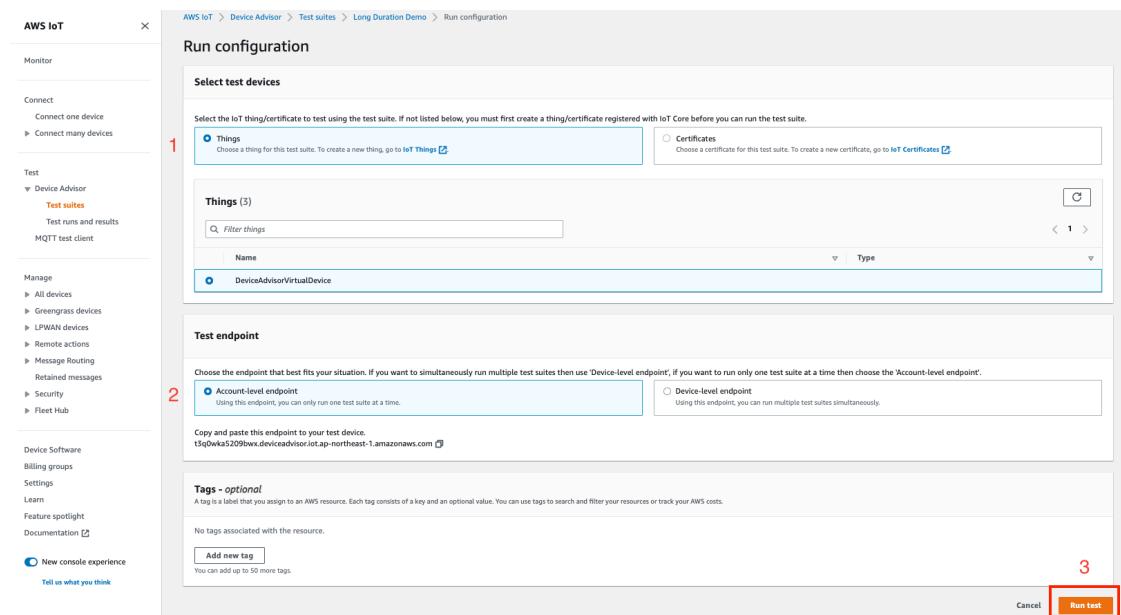
Device operations: CONNECT, PUBLISH, SUBSCRIBE

10. Choisissez les options de configuration sur la page Exécuter la configuration.

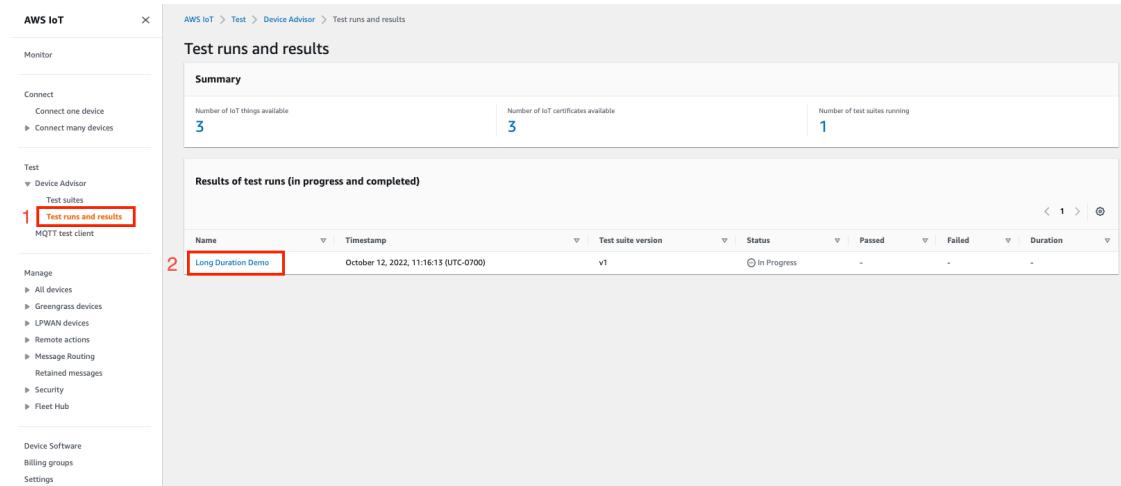
- a. Sélectionnez les objets ou le certificat sur lesquels effectuer le test.
- b. Sélectionnez le point de terminaison au niveau du compte ou le point de terminaison au niveau de l'appareil.
- c. Choisissez Exécuter le test pour l'exécuter.

AWS IoT Core Guide du développeur

Flux de travail de console de tests de longue durée



- Pour afficher les résultats de l'exécution de la suite de tests, sélectionnez Essais et résultats dans le volet de navigation de gauche. Choisissez la suite de tests exécutée pour afficher les détails des résultats.



- L'étape précédente fait apparaître la page de résumé du test. Tous les détails du test sont affichés sur cette page. Lorsque la console vous invite à démarrer la connexion de l'appareil, connectez votre appareil au point de terminaison fourni. La progression des tests est visible sur cette page.

AWS IoT Core Guide du développeur

Points de terminaison VPC Device Advisor () AWS PrivateLink

The screenshot shows the AWS IoT Device Advisor Test suites interface. On the left, a sidebar lists navigation options like Monitor, Connect, Test, and Manage. Under the Test section, 'Test suites' is selected, showing 'Test runs and results' and 'MQTT test client'. The main content area displays a test run titled 'Long Duration Demo' from October 12, 2022, at 11:16:14 (UTC-0700). A message box at the top says 'Connect your device now' and provides instructions to validate the device for MQTT Long duration. Below this is an 'Activity log details' table with columns for Device, Suite version, Created, and Status. The 'MQTT Test Group' section contains two tables: 'Basic tests' and 'Advanced tests', each listing various test cases with their status (e.g., In Progress, Pending). To the right, a separate window titled 'Test log summary' shows a table of log entries with columns for Timestamp and Message.

13. Le test de longue durée fournit un résumé supplémentaire du journal des tests sur le panneau latéral qui affiche tous les événements importants survenus entre l'appareil et le courtier en temps quasi réel. Pour consulter des journaux plus détaillés, cliquez sur Journal des cas de test.

This screenshot is identical to the one above, showing the AWS IoT Device Advisor Test suites interface for the 'Long Duration Demo' test run. It displays the same navigation sidebar, test suite details, activity logs, and MQTT test group tables. To the right, a 'Test log summary' window is open, showing a table of log entries with columns for Timestamp and Message.

Points de terminaison VPC Device Advisor () AWS PrivateLink

Vous pouvez établir une connexion privée entre votre VPC et le point de terminaison de AWS IoT Core Device Advisor test (plan de données) en créant un point de terminaison de VPC d'interface. Vous pouvez utiliser ce point de terminaison pour valider AWS IoT les appareils afin de garantir une connectivité fiable et sécurisée AWS IoT Core avant de les déployer en production. [Les tests prédéfinis de Device Advisor vous](#)

[aident à valider le logiciel de votre appareil par rapport aux meilleures pratiques d'utilisation de TLS, MQTT, Device Shadow et Jobs. AWS IoT](#)

[AWSPrivateLink](#)alimente les points de terminaison d'interface utilisés avec vos appareils IoT. Ce service vous aide à accéder au point de terminaison de AWS IoT Core Device Advisor test en privé, sans passerelle Internet, périphérique NAT, connexion VPN ou AWS Direct Connect connexion. Les instances de votre VPC qui envoient des paquets TCP et MQTT ne requièrent pas d'adresses IP publiques pour communiquer avec AWS IoT Core Device Advisor les points de terminaison de test. Le trafic entre votre VPC et AWS IoT Core Device Advisor ne part pas. AWS Cloud Toutes les communications TLS et MQTT entre les appareils IoT et les scénarios de test de Device Advisor restent dans les limites des ressources de votre Compte AWS

Chaque point de terminaison d'interface est représenté par une ou plusieurs [interfaces réseau Elastic](#) dans vos sous-réseaux.

Pour en savoir plus sur l'utilisation des points de terminaison d'interface VPC, consultez les points de terminaison de [l'interface VPC \(AWS PrivateLink\)](#) dans le guide de l'utilisateur d'Amazon VPC.

Considérations relatives aux points de terminaison de VPC AWS IoT Core Device Advisor

Passez en revue les [propriétés et les limites des points de terminaison d'interface](#) dans le Guide de l'utilisateur d'Amazon VPC avant de configurer les points de terminaison d'interface VPC. Tenez compte des éléments suivants avant de continuer :

- AWS IoT Core Device Advisor prend actuellement en charge les appels vers le point de terminaison de test de Device Advisor (plan de données) depuis votre VPC. Un courtier de messages utilise les communications par plan de données pour envoyer et recevoir des données. Il le fait à l'aide de paquets TLS et MQTT. Points de terminaison VPC pour AWS IoT Core Device Advisor connecter votre AWS IoT appareil aux points de test de Device Advisor. Les [actions de l'API du plan de contrôle](#) ne sont pas utilisées par ce point de terminaison VPC. Pour créer ou exécuter une suite de tests ou d'autres API du plan de contrôle, utilisez la console, un AWS SDK ou une interface de ligne de AWS commande via l'Internet public.
- Les points de terminaison VPC suivants Régions AWS prennent en charge : AWS IoT Core Device Advisor
 - USA Est (Virginie du Nord)
 - USA Ouest (Oregon)
 - Asie Pacifique (Tokyo)
 - Europe (Irlande)
- Device Advisor prend en charge le protocole MQTT avec des certificats client X.509 et des certificats de serveur RSA.
- Les [stratégies relatives aux points de terminaison d'un VPC](#) ne sont pas prises en charge pour le moment.
- Consultez les [conditions requises](#) pour les points de terminaison du VPC pour obtenir des instructions sur la façon de [créer des ressources](#) qui connectent les points de terminaison du VPC. Vous devez créer un VPC et des sous-réseaux privés pour utiliser les points de terminaison du AWS IoT Core Device Advisor VPC.
- Vos ressources AWS PrivateLink font l'objet de quotas. Pour plus d'informations, consultez [AWS PrivateLinkQuotas](#).
- Les points de terminaison VPC ne prennent en charge que le trafic IPv4.

Créer un point de terminaison de VPC d'interface pour AWS IoT Core Device Advisor

Pour commencer à utiliser les points de terminaison VPC, [créez un point de terminaison VPC d'interface](#). Ensuite, sélectionnez AWS IoT Core Device Advisor commeService AWS. Si vous utilisez le AWS CLI, appelez [describe-vpc-endpoint-services](#) pour vérifier qu'il AWS IoT Core Device Advisor est présent dans une zone de disponibilité de votre Région AWS. Vérifiez que le groupe de sécurité attaché au point de terminaison autorise la [communication par protocole TCP](#) pour le trafic MQTT et TLS. Par exemple, dans la région USA Est (Virginie du Nord), utilisez la commande suivante :

```
aws ec2 describe-vpc-endpoint-services --service-name com.amazonaws.us-east-1.deviceadvisor.iot
```

Notez que vous pouvez créer un point de terminaison d'un VPC en AWS IoT Core utilisant le nom de service suivant :

- com.amazonaws. région .deviceadvisor.iot

Par défaut, le DNS privé est activé pour le point de terminaison. Cela garantit que l'utilisation du point de terminaison de test par défaut reste au sein de vos sous-réseaux privés. Pour obtenir votre point de terminaison au niveau de votre compte ou de votre appareil, utilisez la console AWS CLI ou un AWS SDK. Par exemple, si vous exécutez [get-endpoint](#) dans un sous-réseau public ou sur Internet public, vous pouvez obtenir votre point de terminaison et l'utiliser pour vous connecter à Device Advisor. Pour plus d'informations, consultez [Accès à un service via un point de terminaison d'interface](#) dans le Guide de l'utilisateur Amazon VPC.

Pour connecter les clients MQTT aux interfaces de point de terminaison du VPC, le AWS PrivateLink service crée des enregistrements DNS dans une zone hébergée privée rattachée à votre VPC. Ces enregistrements DNS dirigent les demandes de l'AWS IoT appareil vers le point de terminaison du VPC.

Contrôle de l'accès aux points de AWS IoT Core Device Advisor terminaison d'un VPC

Vous pouvez restreindre l'accès des appareils AWS IoT Core Device Advisor et autoriser l'accès uniquement via les points de terminaison du VPC à l'aide des clés contextuelles [conditionnelles](#) du VPC. AWS IoT Core prend en charge les clés de contexte liées au VPC suivantes :

- [SourceVpc](#)
- [SourceVpce](#)
- [Adresse IP source VPC](#)

Note

AWS IoT Core Device Advisor ne prend pas en charge les [politiques relatives aux points de terminaison VPC](#) pour le moment.

La politique suivante autorise la connexion à AWS IoT Core Device Advisor l'aide d'un ID client correspondant au nom de l'objet. Il publie également sur n'importe quel sujet préfixé par le nom de l'objet. La politique dépend de la connexion de l'appareil à un point de terminaison VPC avec un ID de point de terminaison VPC particulier. Cette politique refuse les tentatives de connexion à votre point de terminaison de AWS IoT Core Device Advisor test public.

```
{
```

```
"Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
        "Action": [
          "iot:Connect"
        ],
        "Resource": [
          "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"
        ],
        "Condition": {
          "StringEquals": {
            "aws:SourceVpce": "vpce-1a2b3c4d"
          }
        }
      },
      {
        "Effect": "Allow",
          "Action": [
            "iot:Publish"
          ],
          "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
          ]
        }
      ]
    }
  ]
```

Scénarios de test de Device Advisor

Device Advisor propose des tests prédéfinis dans cinq catégories.

- [TLS \(p. 1204\)](#)
- [MQTT \(p. 1208\)](#)
- [Ombre \(p. 1217\)](#)
- [Exécution du Job \(p. 1219\)](#)
- [Autorisations et politiques \(p. 1220\)](#)
- [Tests de longue durée \(p. 1221\)](#)

Device Advisor teste des scénarios afin de les qualifier pour le programme de qualification des AWS appareils.

Votre appareil doit réussir les tests suivants pour être qualifié conformément au [programme de qualification des AWS appareils](#).

- [Certificat de serveur de nom d'objet incorrect TLS \(p. 1207\)](#) (« Nom commun du sujet (CN) /Nom alternatif du sujet (SAN) incorrect »)
- Certificat de [serveur TLS non sécurisé](#) (« Non signé par une autorité de certification (p. 1207) reconnue »)

- Connect [TLS \(« Connect \(p. 1204\) TLS »\)](#)
- [MQTT Connect \(p. 1208\)](#) (« L'appareil envoie CONNECT à AWS IoT Core (Happy case) »)
- [S'abonner à MQTT \(p. 1214\)](#) (« Je peux m'abonner (Happy Case) »)
- [MQTT Publish \(p. 1212\)](#) (« QoS0 (Happy Case) »)

TLS

Utilisez ces tests pour déterminer si le protocole de sécurité de la couche transport (TLS) entre vos appareils AWS IoT est sécurisé.

Note

Device Advisor prend désormais en charge TLS 1.3.

Chemin heureux

Connect TLS

Vérifie si l'appareil testé peut effectuer l'établissement de la connexion TLS vers AWS IoT. Ce test ne valide pas l'implémentation MQTT de l'appareil client.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Pour de meilleurs résultats, nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_tls_connect_test",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", //in seconds
    },
    "test": {
      "id": "TLS_Connect",
      "version": "0.0.0"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : l'appareil testé a établi une liaison TLS avec AWS IoT
- Réussite avec avertissements — L'appareil testé a établi une connexion TLS avec AWS IoT, mais des messages d'avertissement TLS ont été envoyés par l'appareil ou AWS IoT
- Échec : l'appareil testé n'a pas réussi à établir une connexion TLS avec AWS IoT en raison d'une erreur d'établissement de connexion.

TLS reçoit des fragments de taille maximale

Ce scénario de test confirme que votre appareil peut recevoir et traiter des fragments de taille TLS maximale. Votre appareil de test doit s'abonner à une rubrique préconfigurée avec QoS 1 pour recevoir une charge utile importante. Vous pouvez personnaliser la charge utile avec la configuration \${payload}.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Pour de meilleurs résultats, nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "TLS Receive Maximum Size Fragments",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", //in seconds
      "PAYLOAD_FORMAT": "{\"message\": \"$payload\"}", // A string with a placeholder
      ${payload}, or leave it empty to receive a plain string.
      "TRIGGER_TOPIC": "test_1" // A topic to which a device will subscribe, and to
      which a test case will publish a large payload.
    },
    "test": {
      "id": "TLS_Receive_Maximum_Size_Fragments",
      "version": "0.0.0"
    }
  }
]
```

Suites de chiffrement

Support des appareils TLS pour les suites de AWS IoT chiffrement recommandées

Vérifie que les suites de chiffrement figurant dans le message d'accueil du client TLS envoyé par l'appareil testé contiennent les suites de chiffrement recommandées AWS IoT. (p. 409) Il fournit des informations supplémentaires sur les suites de chiffrement prises en charge par l'appareil.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_tls_support_awst_iot_cipher_suites_test",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", // in seconds
    },
    "test": {
      "id": "TLS_Support_AWS_IoT_Cipher_Suites",
      "version": "0.0.0"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : les suites de chiffrement du périphérique testées contiennent au moins l'une des suites de AWS IoT chiffrement recommandées et ne contiennent aucune suite de chiffrement non prise en charge.

- Passe avec avertissements — Les suites de chiffrement de l'appareil contiennent au moins une suite de AWS IoT chiffrement mais :
 1. Il ne contient aucune des suites de chiffrement recommandées
 2. Il contient des suites de chiffrement qui ne sont pas prises en charge par AWS IoT.
- Nous vous suggérons de vérifier que toutes les suites de chiffrement non prises en charge sont sûres.
- Échec : le périphérique soumis aux suites de chiffrement testées ne contient aucune des suites de chiffrement AWS IoT prises en charge.

Certificat de serveur de plus grande taille

Certificat de serveur TLS de grande taille

Les validateurs sur votre appareil peuvent terminer l'établissement de la liaison TLS AWS IoT lorsqu'il reçoit et traite un certificat de serveur de plus grande taille. La taille du certificat de serveur (en octets) utilisé pour ce test est supérieure à celle actuellement utilisée dans le scénario de test TLS Connect et dans IoT Core de 20 %. Dans ce cas de test, AWS IoT teste l'espace tampon de votre appareil pour le protocole TLS. Si l'espace tampon est suffisamment important, l'établissement de connexion TLS s'effectue sans erreur. Ce test ne valide pas l'implémentation MQTT de l'appareil. Le scénario de test est terminé une fois le processus d'établissement de connexion TLS terminé.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Pour de meilleurs résultats, nous recommandons un délai d'attente de 2 minutes. Si ce scénario de test échoue mais que celui de TLS Connect réussit, nous vous recommandons d'augmenter la limite d'espace tampon de votre appareil pour le protocole TLS. L'augmentation de la limite d'espace tampon permet à votre appareil de traiter un certificat de serveur de plus grande taille au cas où la taille augmenterait.

```
"tests": [
  {
    "name": "my_tls_large_size_server_cert_test",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", // in seconds
    },
    "test": {
      "id": "TLS_Large_Size_Server_Cert",
      "version": "0.0.0"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : l'appareil testé a terminé l'établissement de la connexion TLS avec AWS IoT
- Réussite avec avertissements — L'appareil testé a terminé l'établissement de la connexion TLS avec AWS IoT, mais des messages d'avertissement TLS proviennent soit de l'appareil, soit AWS IoT
- Échec : le périphérique testé n'a pas réussi à terminer l'établissement de connexion TLS en AWS IoT raison d'une erreur au cours du processus d'établissement de connexion.

Certificat de serveur TLS non sécurisé

Non signé par une autorité de certification reconnue

Vérifie que l'appareil testé ferme la connexion s'il reçoit un certificat de serveur sans signature valide de la part de l'autorité de certification ATS. Un appareil ne doit se connecter qu'à un point de terminaison qui présente un certificat valide.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente 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"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : le périphérique testé a fermé la connexion.
- Échec — L'appareil testé a réussi à établir une liaison TLS avec AWS IoT

Certificat du serveur TLS avec nom d'objet incorrect /Nom commun du sujet (CN) incorrect/Nom alternatif du sujet (SAN)

Vérifie que l'appareil testé ferme la connexion s'il reçoit un certificat de serveur pour un nom de domaine différent de celui demandé.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente 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"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : le périphérique testé a fermé la connexion.
- Échec — L'appareil testé a terminé l'établissement de connexion TLS avec AWS IoT

Certificat de serveur TLS expiré

Certificat de serveur expiré

Vérifie que l'appareil testé ferme la connexion s'il reçoit un certificat de serveur expiré.

Example Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_tls_expired_cert_test",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", //in seconds
    },
    "test": {
      "id": "TLS_Expired_Server_Cert",
      "version": "0.0.0"
    }
  }
]
```

Example Sorties du scénario de test :

- Réussite : l'appareil en cours de test refuse de terminer l'établissement de la connexion TLS avec AWS IoT. L'appareil envoie un message d'alerte TLS avant de fermer la connexion.
- Réussite avec avertissements — L'appareil testé refuse de terminer l'établissement de la connexion TLS avec AWS IoT. Toutefois, il n'envoie pas de message d'alerte TLS avant de fermer la connexion.
- Échec : le périphérique testé termine l'établissement de connexion TLS avec AWS IoT

MQTT

CONNECTEZ, DÉCONNECTEZ et RECONNECTEZ

« L'appareil envoie CONNECT à AWS IoT Core (Happy case) »

Vérifie que l'appareil testé envoie une demande CONNECT.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
```

```
{
    "name": "my_mqtt_connect_test",
    "configuration": {
        // optional:
        "EXECUTION_TIMEOUT": "300", // in seconds
    },
    "test": {
        "id": "MQTT_Connect",
        "version": "0.0.0"
    }
}
```

« Le périphérique peut renvoyer PUBACK vers une rubrique arbitraire pour QoS1»

Ce scénario de test vérifiera si l'appareil (client) peut renvoyer un message PUBACK s'il a reçu un message de publication du courtier après s'être abonné à une rubrique avec QoS1.

Le contenu et la taille de la charge utile sont configurables pour ce scénario de test. Si la taille de la charge utile est configurée, Device Advisor remplacera la valeur du contenu de la charge utile et enverra une charge utile prédefinie à l'appareil avec la taille souhaitée. La taille de la charge utile est une valeur comprise entre 0 et 128 Ko et ne peut pas dépasser 128 Ko. AWS IoT Core rejette les demandes de publication et de connexion supérieures à 128 Ko, comme indiqué sur la page des [limites et quotas du courtier de AWS IoT Core messages et du protocole](#).

Définition du scénario de test de l'API :

Note

`EXECUTION_TIMEOUT` dispose d'une valeur par défaut. Nous recommandons un délai d'attente de 2 minutes. `PAYLOAD_SIZE` peut être configuré sur une valeur comprise entre 0 et 128 kilo-octets. La définition d'une taille de charge utile remplace le contenu de la charge utile car Device Advisor renverra une charge utile prédefinie avec la taille donnée à l'appareil.

```
"tests": [
{
    "name": "my_mqtt_client_puback_qos1",
    "configuration": {
        // optional: "TRIGGER_TOPIC": "myTopic",
        "EXECUTION_TIMEOUT": "300", // in seconds
        "PAYLOAD_FOR_PUBLISH_VALIDATION": "custom payload",
        "PAYLOAD_SIZE": "100" // in kilobytes
    },
    "test": {
        "id": "MQTT_Client_Puback_QoS1",
        "version": "0.0.0"
    }
}]
```

« Rétentatives de connexion de l'appareil avec interruption de l'instabilité : aucune réponse CONNACK »

Vérifie que l'appareil testé utilise la fonction d'atténuation de gigue appropriée lorsqu'il se reconnecte au moins cinq fois avec le courtier. Le broker enregistre l'horodatage de la demande CONNECT de l'appareil testé, valide le paquet, fait une pause 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 vers l'appareil en cours de test.

Le processus précédent est à nouveau exécuté. Au total, ce scénario de test nécessite que l'appareil se connecte au moins 12 fois au total. Les horodatages collectés sont utilisés pour valider que l'appareil testé utilise la fonction Jitter Backoff. Si le périphérique testé présente un délai de décélération strictement exponentiel, ce scénario de test sera réussi avec des avertissements.

Nous recommandons la mise en œuvre du mécanisme [Exponential Backoff And Jitter](#) sur l'appareil testé pour réussir ce scénario de test.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente 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"
    }
  }
]
```

« Rétentatives de connexion de l'appareil avec interruption exponentielle : aucune réponse CONNACK »

Vérifie que l'appareil testé utilise le décalage exponentiel approprié lors de la reconnexion avec le courtier au moins cinq fois. Le broker enregistre l'horodatage de la demande CONNECT de l'appareil testé, valide le paquet, fait une pause sans envoyer de CONNACK à l'appareil client et attend que le périphérique testé renvoie la demande. Les horodatages collectés sont utilisés pour valider qu'un décalage exponentiel est utilisé par le dispositif testé.

Nous recommandons la mise en œuvre du mécanisme [Exponential Backoff And Jitter](#) sur l'appareil testé pour réussir ce scénario de test.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente 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"
    }
  }
]
```

« Reconnexion de l'appareil avec fonction Jitter Backoff : après la déconnexion du serveur »

Vérifie si un appareil en cours de test utilise les instabilités et les ralentissements nécessaires lors de sa reconnexion après avoir été déconnecté du serveur. Device Advisor déconnecte l'appareil du serveur au moins cinq fois et observe le comportement de l'appareil lors de la reconnexion MQTT. Device Advisor enregistre l'horodatage de la demande CONNECT pour le périphérique testé, valide

le paquet, fait une pause sans envoyer de CONNACK à l'appareil client et attend que le périphérique testé renvoie la demande. Les horodatages collectés sont utilisés pour valider que l'appareil testé utilise des instabilités et des blocages lors de sa reconnexion. Si l'appareil testé présente un ralentissement strictement exponentiel ou s'il ne met pas en œuvre un mécanisme d'annulation de gigue approprié, ce scénario de test sera réussi avec des avertissements. Si l'appareil testé a mis en œuvre un mécanisme d'arrêt linéaire ou constant, le test échouera.

Pour réussir ce scénario de test, nous vous recommandons d'implémenter le mécanisme [Exponential Backoff And Jitter](#) sur l'appareil testé.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 4 minutes.

Le nombre de tentatives de reconnexion à valider pour le backoff peut être modifié en spécifiant le. RECONNECTION_ATTEMPTS Le nombre doit être compris entre 5 et 10. La valeur par défaut est 5.

```
"tests": [
  {
    "name": "my_mqtt_reconnect_backoff_retries_on_server_disconnect",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", // in seconds
      "RECONNECTION_ATTEMPTS": 5
    },
    "test": {
      "id": "MQTT_Reconnect_Backoff_Retries_On_Server_Disconnect",
      "version": "0.0.0"
    }
  }
]
```

« Reconnexion de l'appareil avec fonction Jitter Backoff - En cas de connexion instable »

Valide si un appareil en cours de test utilise les instabilités et les ralentissements nécessaires lors de sa reconnexion sur une connexion instable. Device Advisor déconnecte l'appareil du serveur après cinq connexions réussies et observe le comportement de l'appareil lors de la reconnexion MQTT. Device Advisor enregistre l'horodatage de la demande CONNECT pour le périphérique testé, valide le paquet, renvoie CONNACK, se déconnecte, enregistre l'horodatage de la déconnexion et attend que le périphérique testé renvoie la demande. Les horodatages collectés sont utilisés pour valider que l'appareil testé utilise des instabilités et des ralentissements lors de la reconnexion après des connexions réussies mais instables. Si l'appareil testé présente un ralentissement strictement exponentiel ou s'il ne met pas en œuvre un mécanisme d'annulation de gigue approprié, ce scénario de test sera réussi avec des avertissements. Si l'appareil testé a mis en œuvre un mécanisme d'arrêt linéaire ou constant, le test échouera.

Pour réussir ce scénario de test, nous vous recommandons d'implémenter le mécanisme [Exponential Backoff And Jitter](#) sur l'appareil testé.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 4 minutes.

Le nombre de tentatives de reconnexion à valider pour le backoff peut être modifié en spécifiant le. RECONNECTION_ATTEMPTS Le nombre doit être compris entre 5 et 10. La valeur par défaut est 5.

```
"tests": [
  {
    "name": "my_mqtt_reconnect_backoff_retries_on_unstable_connection",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", // in seconds
      "RECONNECTION_ATTEMPTS": 5
    },
    "test": {
      "id": "MQTT_Reconnect_Backoff_Retries_On_Unstable_Connection",
      "version": "0.0.0"
    }
  }
]
```

Publier

« QoS0 (Happy Case) »

Vérifie que le périphérique testé publie un message avec QoS0 et QoS1. Vous pouvez également valider le sujet du message et la charge utile en spécifiant la valeur du sujet et la charge utile dans les paramètres de test.

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_mqtt_publish_test",
    "configuration": {
      // 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"
    }
  }
]
```

« Nouvelle tentative de publication avec QoS1 - Pas de PUBACK »

Vérifie que l'appareil testé republie un message envoyé avec QoS1, si le courtier n'envoie pas PUBACK. Vous pouvez également valider le sujet du message en spécifiant ce sujet dans les paramètres du test. L'appareil client ne doit pas se déconnecter avant de republier le message. Ce test permet également de vérifier que le message republié possède le même identifiant de paquet que l'original. Pendant l'exécution du test, si l'appareil perd la connexion et se reconnecte, le scénario de test se réinitialisera sans échec et l'appareil doit recommencer les étapes du scénario de test.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Il est recommandé de le faire pendant au moins 4 minutes.

```
"tests": [
  {
    "name": "my_mqtt_publish_retry_test",
    "configuration": {
      // 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"
    }
  }
]
```

« Publier les messages conservés »

Vérifie que l'appareil testé publie un message avec la valeur retainFlag true. Vous pouvez valider le sujet et la charge utile du message en définissant la valeur du sujet et la charge utile dans les paramètres de test. Si le message retainFlag envoyé dans le paquet PUBLISH n'est pas défini sur true, le scénario de test échouera.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 2 minutes. Pour exécuter ce scénario de test, ajoutez l'iot:RetainPublishaction dans le [rôle de votre appareil](#).

```
"tests": [
  {
    "name": "my_mqtt_publish_retained_messages_test",
    "configuration": {
      // optional:
      "EXECUTION_TIMEOUT": "300", // in seconds
      "TOPIC_FOR_PUBLISH_RETAINED_VALIDATION": "my_TOPIC_FOR_PUBLISH_RETAINED_VALIDATION",
      "PAYLOAD_FOR_PUBLISH_RETAINED_VALIDATION": "my_PAYLOAD_FOR_PUBLISH_RETAINED_VALIDATION",
    },
    "test": {
      "id": "MQTT_Publish_Retained_Messages",
      "version": "0.0.0"
    }
  }
]
```

« Publier avec la propriété de l'utilisateur »

Vérifie que l'appareil testé publie un message avec la propriété utilisateur correcte. Vous pouvez valider la propriété utilisateur en définissant la paire nom-valeur dans les paramètres de test. Si la propriété utilisateur n'est pas fournie ou ne correspond pas, le scénario de test échoue.

Définition du scénario de test de l'API :

Note

Il s'agit d'un cas de test réservé au MQTT5.

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_mqtt_user_property_test",
    "test": {
      "USER_PROPERTIES": [
        {"name": "name1", "value": "value1"},
        {"name": "name2", "value": "value2"}
      ],
      "EXECUTION_TIMEOUT": "300", // in seconds
    },
    "test": {
      "id": "MQTT_Publish_User_Property",
      "version": "0.0.0"
    }
  }
]
```

S'abonner

« Je peux m'abonner (Happy Case) »

Vérifie que l'appareil testé est abonné aux rubriques MQTT. Vous pouvez également valider la rubrique à laquelle l'appareil testé est abonné en spécifiant cette rubrique dans les paramètres du test.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 2 minutes.

```
"tests": [
  {
    "name": "my_mqtt_subscribe_test",
    "configuration": {
      // 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"
      }
    }
]
```

« S'abonner, réessayer, pas de SUBACK »

Vérifie que l'appareil en cours de test réessaie un abonnement qui a échoué aux rubriques MQTT. Le serveur attend alors et n'envoie pas de SUBACK. Si l'appareil client ne réessaie pas l'abonnement, le test échoue. L'appareil client doit réessayer l'abonnement qui a échoué avec le même identifiant de paquet. Vous pouvez également valider la rubrique à laquelle l'appareil testé est abonné en spécifiant cette rubrique dans les paramètres du test. Pendant l'exécution du test, si l'appareil perd la connexion et se reconnecte, le scénario de test se réinitialisera sans échec et l'appareil doit recommencer les étapes du scénario de test.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 4 minutes.

```
"tests": [
  {
    "name": "my_mqtt_subscribe_retry_test",
    "configuration": {
      "EXECUTION_TIMEOUT": "300", // in seconds
      // optional:
      "TOPIC_LIST_FOR_SUBSCRIPTION_VALIDATION_ID": [
        "my_TOPIC_FOR_PUBLISH_VALIDATION_a", "my_TOPIC_FOR_PUBLISH_VALIDATION_b"
      ],
      "test": {
        "id": "MQTT_Subscribe_Retry_No_Suback",
        "version": "0.0.0"
      }
    }
]
```

Keep-Alive

« Matt No Ack PingResp »

Ce scénario de test permet de vérifier si l'appareil testé se déconnecte lorsqu'il ne reçoit pas de réponse ping. Dans le cadre de ce scénario de test, Device Advisor bloque les réponses envoyées AWS IoT Core pour les demandes de publication, d'abonnement et de ping. Il vérifie également si l'appareil testé déconnecte la connexion MQTT.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente supérieur à 1,5 fois la keepAliveTime valeur.

```
"tests": [
  {
    "name": "Mqtt_No_Ack_PingResp",
    "configuration": {
      //optional:
      "EXECUTION_TIMEOUT": "306", // in seconds
    },
    "test": {
      "id": "MQTT_No_Ack_PingResp",
      "version": "0.0.0"
    }
  }
]
```

Session persistante

« Session persistante (Happy Case) »

Ce scénario de test valide le comportement de l'appareil lorsqu'il est déconnecté d'une session persistante. Le scénario de test vérifie si l'appareil peut se reconnecter, reprendre les abonnements

à ses sujets déclencheurs sans se réabonner explicitement, recevoir les messages stockés dans les rubriques et fonctionner comme prévu pendant une session persistante. Lorsque ce scénario de test est réussi, cela indique que l'appareil client est capable de maintenir une session persistante avec le AWS IoT Core courtier de la manière attendue. Pour plus d'informations sur les sessions AWS IoT persistantes, consultez la section [Utilisation des sessions persistantes MQTT](#).

Dans ce cas de test, l'appareil client doit se CONNECTER AWS IoT Core avec l'indicateur de session propre défini sur false, puis s'abonner à un sujet déclencheur. Une fois l'abonnement réussi, l'appareil sera déconnecté par AWS IoT Core Device Advisor. Lorsque l'appareil est déconnecté, une charge utile de message QoS 1 sera stockée dans cette rubrique. Device Advisor autorisera ensuite l'appareil client à se reconnecter au point de terminaison de test. À ce stade, étant donné qu'il existe une session persistante, l'appareil client devrait reprendre ses abonnements aux rubriques sans envoyer de paquets SUBSCRIBE supplémentaires et recevoir le message QoS 1 du courtier. Après la reconnexion, si l'appareil client se réabonne à son sujet déclencheur en envoyant un paquet SUBSCRIBE supplémentaire et/ou si le client ne reçoit pas le message stocké à partir du sujet déclencheur, le scénario de test échouera.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'expiration d'au moins 4 minutes. Lors de la première connexion, l'appareil client doit s'abonner explicitement à un service TRIGGER_TOPIC auquel il n'était pas abonné auparavant. Pour réussir le test, l'appareil client doit s'abonner avec succès TRIGGER_TOPIC avec un QoS 1. Après la reconnexion, l'appareil client est censé comprendre qu'il existe une session persistante active ; il doit donc accepter le message enregistré envoyé par le sujet déclencheur et renvoyer PUBACK pour ce message spécifique.

```
"tests": [
  {
    "name": "my_mqtt_persistent_session_happy_case",
    "configuration": {
      //required:
      "TRIGGER_TOPIC": "myTrigger/topic",
      // optional:
      // if Payload not provided, a string will be stored in the trigger topic to be
      // sent back to the client device
      "PAYLOAD": "The message which should be received from AWS IoT Broker after re-
      connecting to a persistent session from the specified trigger topic.",
      "EXECUTION_TIMEOUT": "300" // in seconds
    },
    "test": {
      "id": "MQTT_Persistent_Session_Happy_Case",
      "version": "0.0.0"
    }
  }
]
```

« Session persistante - Expiration de la session »

Ce scénario de test permet de valider le comportement de l'appareil lorsqu'un appareil déconnecté se reconnecte à une session persistante expirée. Une fois la session expirée, nous nous attendons à ce que l'appareil se réabonne aux sujets auxquels il était précédemment abonné en envoyant explicitement un nouveau paquet SUBSCRIBE.

Lors de la première connexion, nous nous attendons à ce que l'appareil de test se CONNECTE au courtier AWS IoT, car son CleanSession indicateur est défini sur false pour lancer une session persistante. L'appareil doit ensuite s'abonner à un sujet déclencheur. L'appareil est ensuite déconnecté par AWS IoT Core Device Advisor, après un abonnement réussi et le lancement d'une session

persistante. Après la déconnexion, AWS IoT Core Device Advisor permet à l'appareil de test de se reconnecter au point de terminaison de test. À ce stade, lorsque l'appareil de test envoie un autre paquet CONNECT, AWS IoT Core Device Advisor renvoie un paquet CONNACK qui indique que la session persistante a expiré. Le périphérique de test doit interpréter correctement ce paquet et il est censé se réabonner au même sujet déclencheur lorsque la session persistante est terminée. Si l'appareil de test ne se réabonne pas à son déclencheur de sujet, le scénario de test échoue. Pour que le test soit réussi, l'appareil doit comprendre que la session persistante est terminée et renvoyer un nouveau paquet SUBSCRIBE pour le même sujet déclencheur lors de la deuxième connexion.

Si ce scénario de test réussit pour un appareil de test, cela indique que le périphérique est capable de gérer la reconnexion à l'expiration d'une session persistante de la manière attendue.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'expiration d'au moins 4 minutes. L'appareil de test doit s'abonner explicitement à un TRIGGER_TOPIC, auquel il n'était pas abonné auparavant. Pour réussir le test, l'appareil de test doit envoyer un paquet CONNECT avec l'CleanSession indicateur défini sur false et s'abonner avec succès à un sujet déclencheur avec une QoS 1. Une fois la connexion établie, AWS IoT Core Device Advisor déconnecte l'appareil. Après la déconnexion, AWS IoT Core Device Advisor autorise l'appareil à se reconnecter, et l'appareil devrait s'y réabonner TRIGGER_TOPIC car AWS IoT Core Device Advisor aurait mis fin à la session persistante.

```
"tests": [
  {
    "name": "my_expired_persistent_session_test",
    "configuration": {
      //required:
      "TRIGGER_TOPIC": "myTrigger/topic",
      // optional:
      "EXECUTION_TIMEOUT": "300" // in seconds
    },
    "test": {
      "id": "MQTT_Expired_Persistent_Session",
      "version": "0.0.0"
    }
  }
]
```

Shadow

Utilisez ces tests pour vérifier que vos appareils testés utilisent correctement le service AWS IoT Device Shadow. Pour en savoir plus, consultez [Service AWS IoT Device Shadow \(p. 690\)](#). Si ces scénarios de test sont configurés dans votre suite de tests, il est nécessaire de fournir un élément lors du démarrage de l'exécution de la suite.

Le MQTT over n'WebSocket test pas pris en charge pour le moment.

Publier

« L'appareil publie son état après la connexion (Happy case) »

Vérifie si un appareil peut publier son état après s'être connecté à AWS IoT Core

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente 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"
    }
  }
]
```

Ils REPORTED_STATE peuvent être fournis pour une validation supplémentaire de l'état d'ombre exact de votre appareil, une fois qu'il s'est connecté. Par défaut, ce scénario de test valide l'état de publication de votre appareil.

Si ce n'**SHADOW_NAME** est pas le cas, le scénario de test recherche les messages publiés avec des préfixes de rubrique de type ombre anonyme (classique) par défaut. Indiquez un nom d'ombre si votre appareil utilise le type d'ombre indiqué. Pour plus d'informations, consultez [la section Utilisation des ombres sur les appareils](#).

Mettre à jour

« L'appareil met à jour l'état signalé vers l'état souhaité (cas heureux) »

Vérifie 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 possède déjà une ombre avant d'exécuter le test, assurez-vous que l'état souhaité configuré pour le scénario 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 consultant le ClientToken champ du document Shadow tel qu'il seraDeviceAdvisorShadowTestCaseSetup.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur par défaut. Nous recommandons un délai d'attente 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"
    }
]
]
```

DESIRED_STATEII doit comporter au moins un attribut et une valeur associée.

Si ce n'SHADOW_NAMEest pas le cas, le scénario de test recherche les messages publiés avec des préfixes de rubrique du type d'ombre anonyme (classique) par défaut. Indiquez un nom d'ombre si votre appareil utilise le type d'ombre indiqué. Pour plus d'informations, consultez [la section Utilisation des ombres sur les appareils](#).

Exécution de Job

« L'appareil peut terminer l'exécution d'une tâche »

Ce scénario de test vous permet de vérifier si votre appareil est en mesure de recevoir des mises à jour à l'aide de AWS IoT Jobs et de publier l'état des mises à jour réussies. Pour plus d'informations sur les AWS IoT tâches, consultez la section [Tâches](#).

Pour exécuter correctement ce scénario de test, vous devez attribuer un [rôle dans l'appareil](#) à deux AWS rubriques réservées. Pour vous abonner aux messages relatifs à l'activité professionnelle, utilisez les rubriques Notify et Notify-next. Le rôle de votre appareil doit autoriser l'action PUBLIER pour les sujets suivants :

- \$aws/things/thingName/jobs/jobId/get
- \$aws/things/thingName/jobs/jobId/update

Il est recommandé d'accorder des actions SUBSCRIBE et RECEIVE pour les sujets suivants :

- \$aws/things/thingName/jobs/get/accepted
- \$aws/things/thingName/jobs/jobId/get/rejected
- \$aws/things/thingName/jobs/jobId/update/accepted
- \$aws/things/thingName/jobs/jobId/update/rejected

Il est recommandé d'autoriser l'action SUBSCRIBE pour le sujet suivant :

- \$aws/things/thingName/jobs/notify-next

Pour plus d'informations sur ces rubriques réservées, consultez la section Rubriques réservées pour [AWS IoTJobs](#).

Le MQTT over n'WebSocket est pas pris en charge pour le moment.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous recommandons un délai d'attente de 3 minutes. En fonction du document de AWS IoT Job ou de la source fournis, ajustez la valeur du délai d'expiration (par exemple, si l'exécution d'une tâche prend beaucoup de temps, définissez une valeur de délai d'expiration plus longue pour le scénario de test). Pour exécuter le test, un document de AWS IoT Job valide ou un ID de tâche déjà existant

est requis. Un document de AWS IoT Job peut être fourni sous forme de document JSON ou de lien S3. Si un document de travail est fourni, la fourniture d'un identifiant de poste est facultative. Si un identifiant de Job est fourni, Device Advisor utilisera cet identifiant lors de la création de la AWS IoT tâche en votre nom. Si le document de travail n'est pas fourni, vous pouvez fournir un identifiant existant qui se trouve dans la même région que celle où vous exécutez le scénario de test. Dans ce cas, Device Advisor utilisera ce AWS IoT Job lors de l'exécution du scénario de test.

```

"tests": [
  {
    "name": "my_job_execution",
    "configuration": {
      // optional:
      // Test case will create a job task by using either JOB_DOCUMENT or
      JOB_DOCUMENT_SOURCE.
      // If you manage the job task on your own, leave it empty and provide the
      JOB_JOBID (self-managed job task).
      // JOB_DOCUMENT is a JSON formatted string
      "JOB_DOCUMENT": "{\n        \"operation\": \"reboot\", \n        \"files\": {\n            \"fileName\": \"install.py\", \n            \"url\": \"${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket-\nname/key}\\\" \n        }\n    }",
      // JOB_DOCUMENT_SOURCE is an S3 link to the job document. It will be used only
      if JOB_DOCUMENT is not provided.
      "JOB_DOCUMENT_SOURCE": "https://s3.amazonaws.com/bucket-name/key",
      // JOB_JOBID is mandatory, only if neither document nor document source is
      provided. (Test case needs to know the self-managed job task id).
      "JOB_JOBID": "String",
      // JOB_PRESIGN_ROLE_ARN is used for the presign Url, which will replace the
      placeholder in the JOB_DOCUMENT field
      "JOB_PRESIGN_ROLE_ARN": "String",
      // Presigned Url expiration time. It must be between 60 and 3600 seconds, with
      the default value being 3600.
      "JOB_PRESIGN_EXPIRES_IN_SEC": "Long"
      "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 travail, consultez le [document de travail](#).

Autorisations et politiques

Vous pouvez utiliser les tests suivants pour déterminer si les politiques associées aux certificats de vos appareils respectent les meilleures pratiques standard.

Le MQTT over n'WebSocket est pas pris en charge pour le moment.

« Les politiques relatives aux certificats d'appareils ne contiennent pas de caractères génériques »

Vérifie si les politiques d'autorisation associées à un appareil respectent les meilleures pratiques et n'accordent pas à l'appareil plus d'autorisations que nécessaire.

Définition du scénario de test de l'API :

Note

EXECUTION_TIMEOUT dispose d'une valeur de par défaut. Nous vous recommandons de définir un délai d'expiration d'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"
    }
  }
]
```

Tests de longue durée

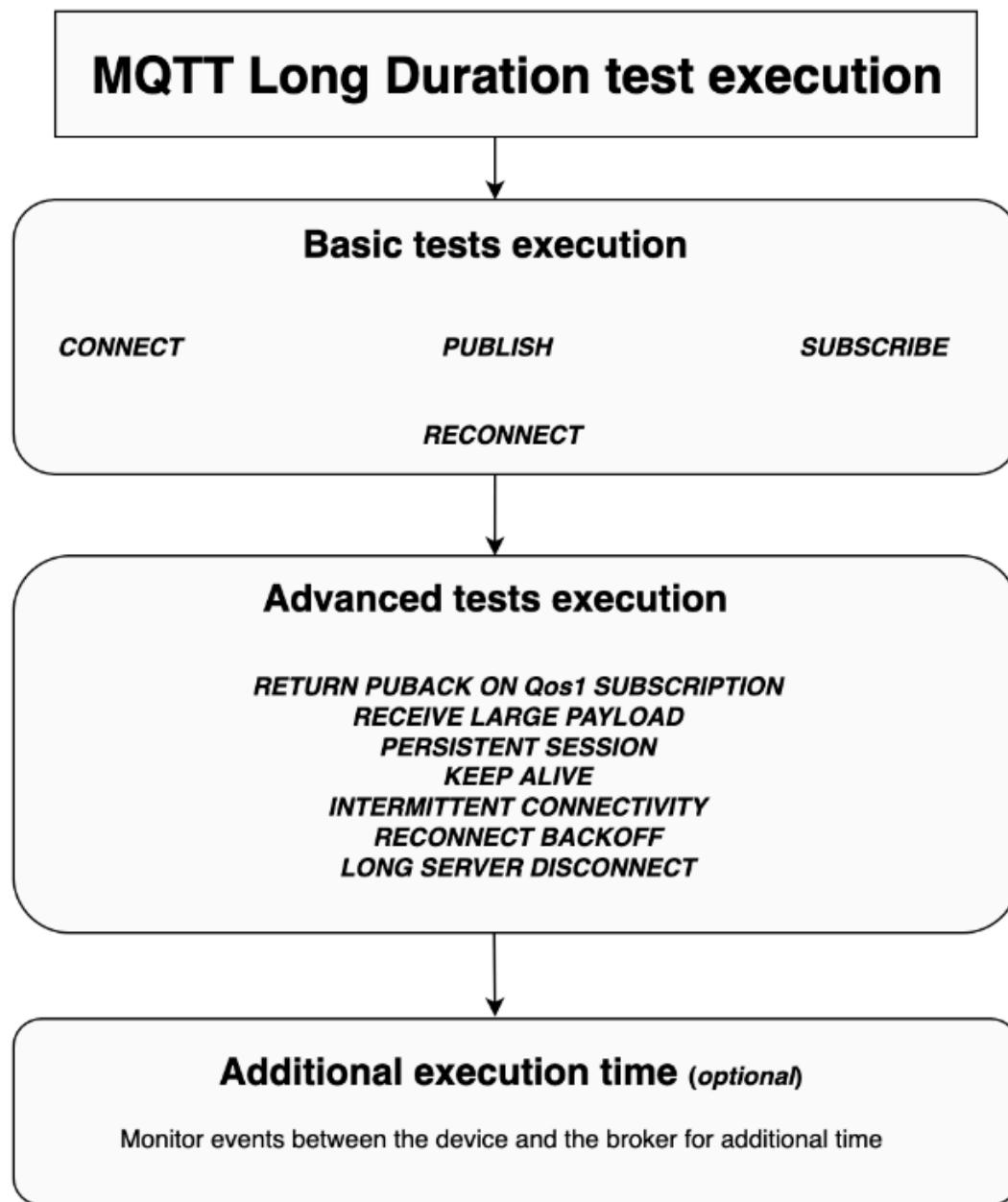
Les tests de longue durée sont une nouvelle suite de tests qui surveille le comportement d'un appareil lorsqu'il fonctionne sur de longues périodes. Contrairement aux tests individuels qui se concentrent sur des comportements spécifiques d'un appareil, le test de longue durée examine le comportement de l'appareil dans divers scénarios réels tout au long de la durée de vie de l'appareil. Device Advisor orchestre les tests dans l'ordre le plus efficace possible. Le test génère des résultats et des journaux, y compris un journal récapitulatif contenant des mesures utiles sur les performances de l'appareil pendant le test.

Boîtier de test de longue durée MQTT

Dans le scénario de test de longue durée MQTT, le comportement de l'appareil est initialement observé dans des scénarios optimistes tels que MQTT Connect, Subscribe, Publish et Reconnect. Ensuite, le périphérique est observé dans de multiples scénarios de défaillance complexes tels que l'interruption de la connexion MQTT, la déconnexion prolongée du serveur et la connectivité intermittente.

Flux d'exécution de scénarios de test de longue durée MQTT

L'exécution d'un scénario de test MQTT de longue durée comporte trois phases :



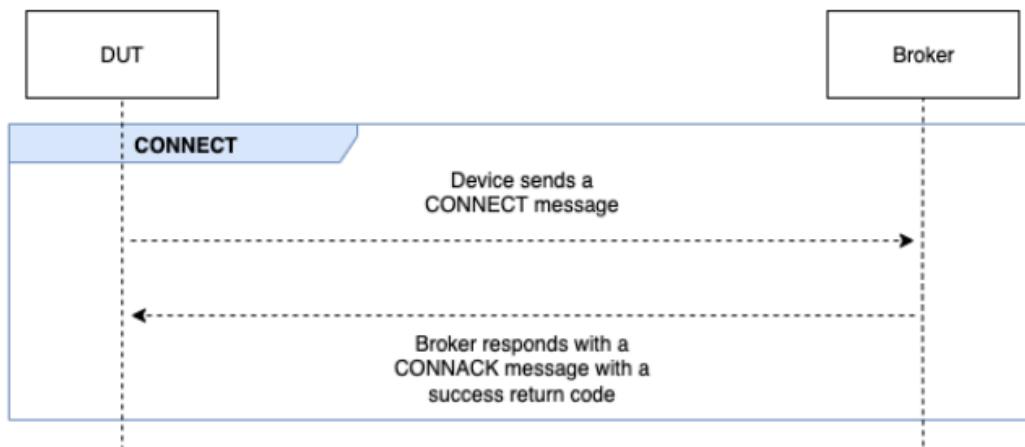
Exécution de tests de base

Dans cette phase, le scénario de test exécute des tests simples en parallèle. Le test permet de vérifier si l'appareil possède les opérations sélectionnées dans la configuration.

L'ensemble des tests de base peut inclure les éléments suivants, en fonction des opérations sélectionnées :

CONNECT

Ce scénario permet de vérifier si l'appareil est capable d'établir une connexion réussie avec le courtier.

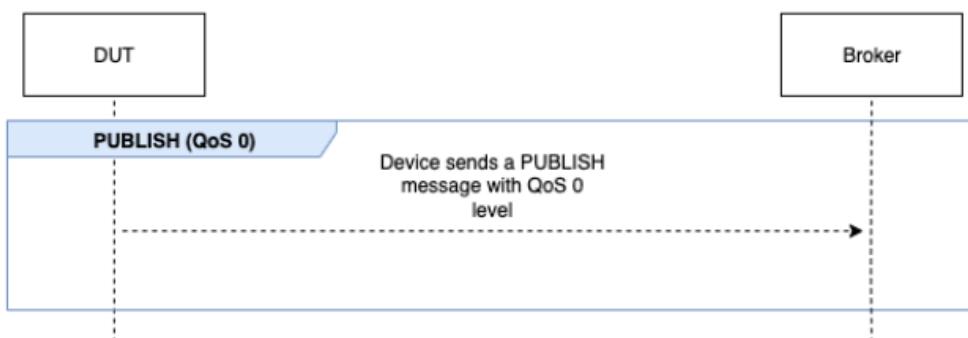


PUBLIER

Ce scénario permet de vérifier si l'appareil publie avec succès auprès du courtier.

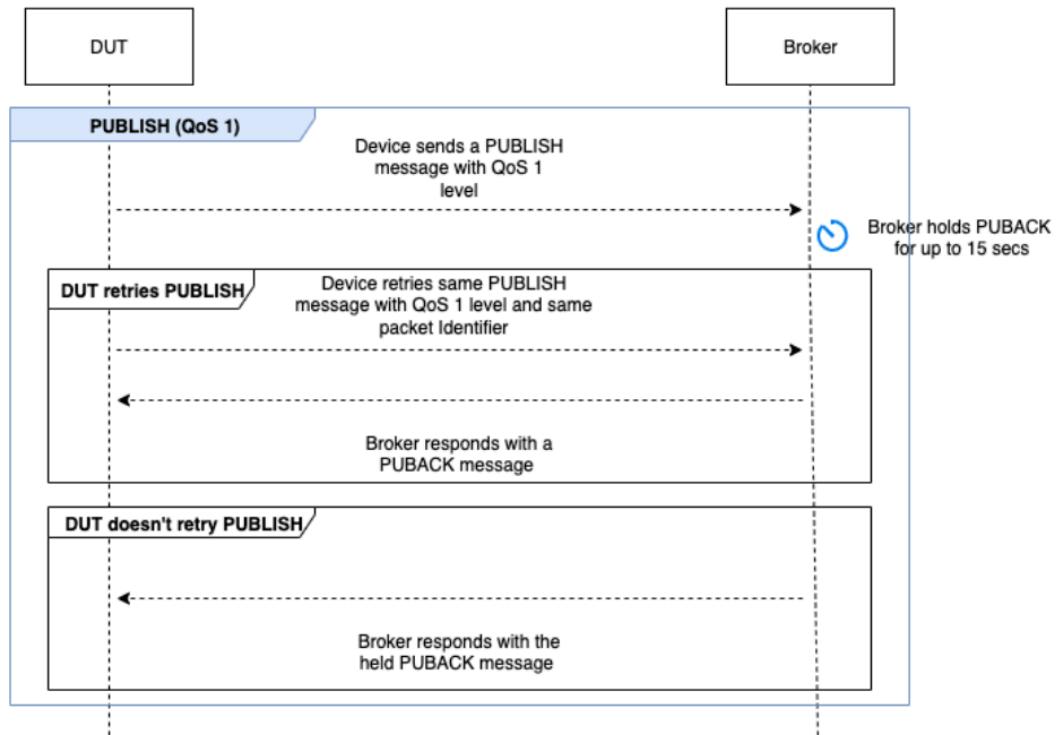
QoS 0

Ce scénario de test permet de vérifier si l'appareil envoie correctement un PUBLISH message au broker lors d'une publication avec QoS 0. Le test n'attend pas que le PUBACK message soit reçu par l'appareil.



QoS 1

Dans ce cas de test, l'appareil devrait envoyer deux PUBLISH messages au courtier avec QoS 1. Après le premier PUBLISH message, le courtier attend jusqu'à 15 secondes avant de répondre. L'appareil doit réessayer le PUBLISH message d'origine avec le même identifiant de paquet dans la fenêtre de 15 secondes. Si tel est le cas, le courtier répond par un PUBACK message et le test est validé. Si l'appareil ne réessaie pas PUBLISH, l'original PUBACK est envoyé à l'appareil et le test est marqué comme Réussi avec des avertissements, ainsi qu'un message système. Pendant l'exécution du test, si l'appareil perd la connexion et se reconnecte, le scénario de test se réinitialisera sans échec et l'appareil doit recommencer les étapes du scénario de test.

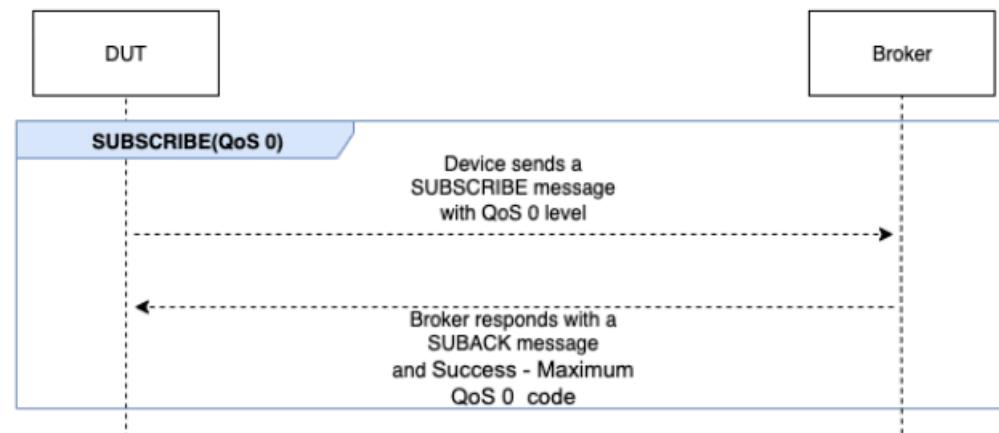


SOUSCRIRE

Ce scénario permet de vérifier si l'appareil s'abonne avec succès auprès du courtier.

QoS 0

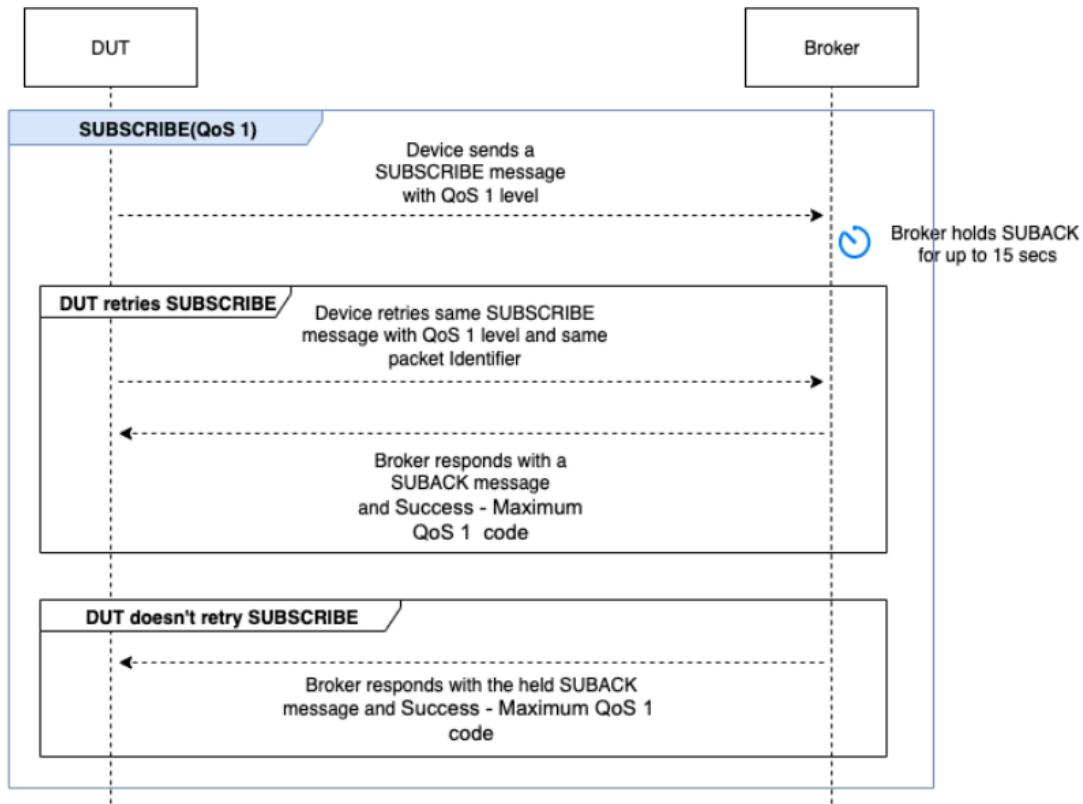
Ce scénario de test permet de vérifier si l'appareil envoie avec succès un SUBSCRIBE message au courtier lors d'un abonnement avec QoS 0. Le test n'attend pas que l'appareil reçoive un message SUBACK.



QoS 1

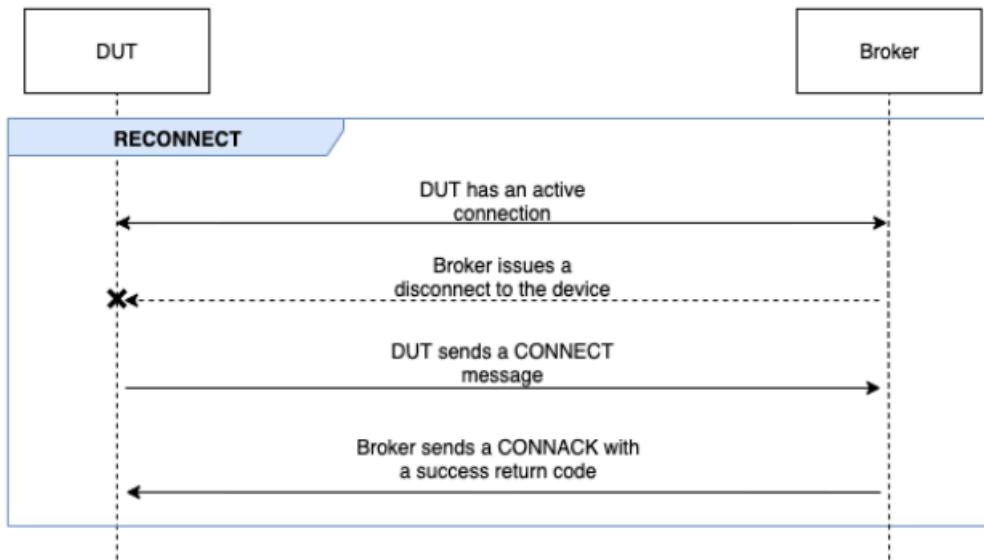
Dans ce cas de test, l'appareil devrait envoyer deux SUBSCRIBE messages au courtier avec QoS 1. Après le premier SUBSCRIBE message, le courtier attend jusqu'à 15 secondes avant de répondre. L'appareil

doit réessayer le SUBSCRIBE message d'origine avec le même identifiant de paquet dans la fenêtre de 15 secondes. Si tel est le cas, le courtier répond par un SUBACK message et le test est validé. Si l'appareil ne réessaie pasSUBSCRIBE, l'original SUBACK est envoyé à l'appareil et le test est marqué comme Réussi avec des avertissements, ainsi qu'un message système. Pendant l'exécution du test, si l'appareil perd la connexion et se reconnecte, le scénario de test se réinitialisera sans échec et l'appareil doit recommencer les étapes du scénario de test.



RECONNECTER

Ce scénario permet de vérifier si l'appareil se reconnecte avec succès au broker après avoir été déconnecté d'une connexion réussie. Device Advisor ne déconnectera pas l'appareil s'il s'est connecté plusieurs fois au cours de la suite de tests. Au lieu de cela, il marquera le test comme étant réussi.



Exécution de tests avancés

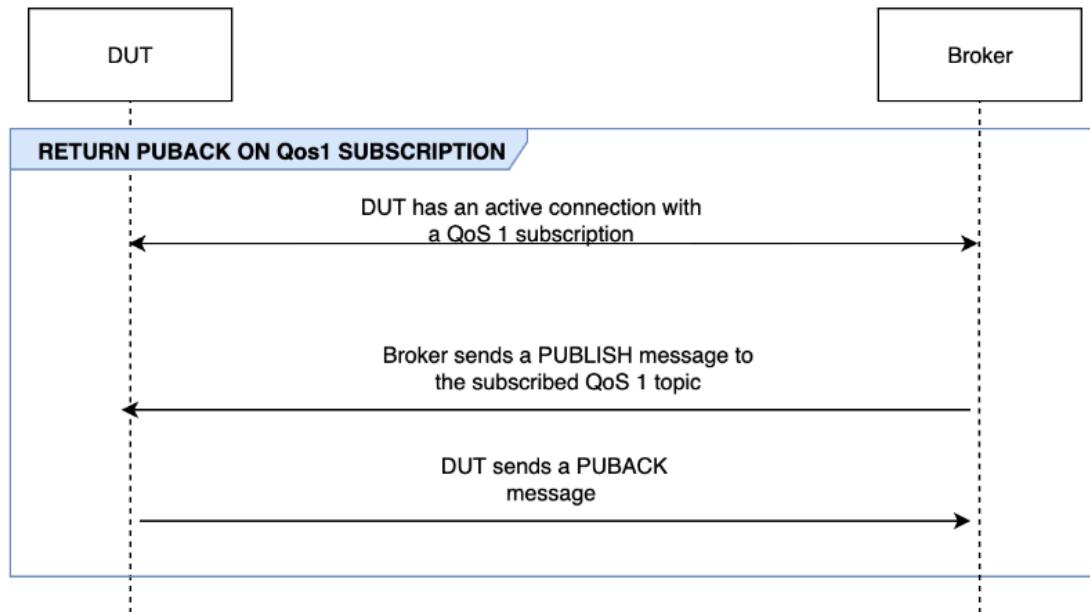
Au cours de cette phase, le scénario de test exécute des tests plus complexes en série pour valider si l'appareil suit les meilleures pratiques. Ces tests avancés peuvent être sélectionnés et peuvent être désactivés s'ils ne sont pas nécessaires. Chaque test avancé possède sa propre valeur de délai d'attente en fonction des exigences du scénario.

RETOUR DE L'ABONNEMENT À QoS 1

Note

Sélectionnez ce scénario uniquement si votre appareil est capable d'exécuter des abonnements QoS 1.

Ce scénario valide si, une fois que l'appareil s'est abonné à une rubrique et a reçu un PUBLISH message du courtier, il renvoie un PUBACK message.

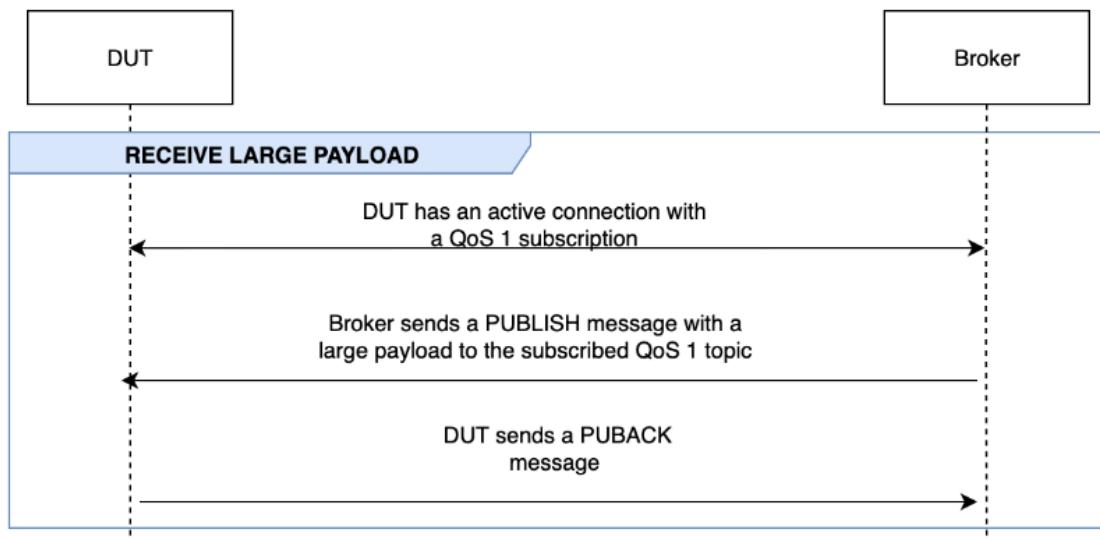


RECEVOIR UNE CHARGE UTILE IMPORTANTE

Note

Sélectionnez ce scénario uniquement si votre appareil est capable d'exécuter des abonnements QoS 1.

Ce scénario vérifie si l'appareil répond par un PUBACK message après avoir reçu un PUBLISH message du courtier concernant un sujet de QoS 1 comportant une charge utile importante. Le format de la charge utile attendue peut être configuré à l'aide de l'`LONG_PAYLOAD_FORMAT` option.



SESSION PERSISTANTE

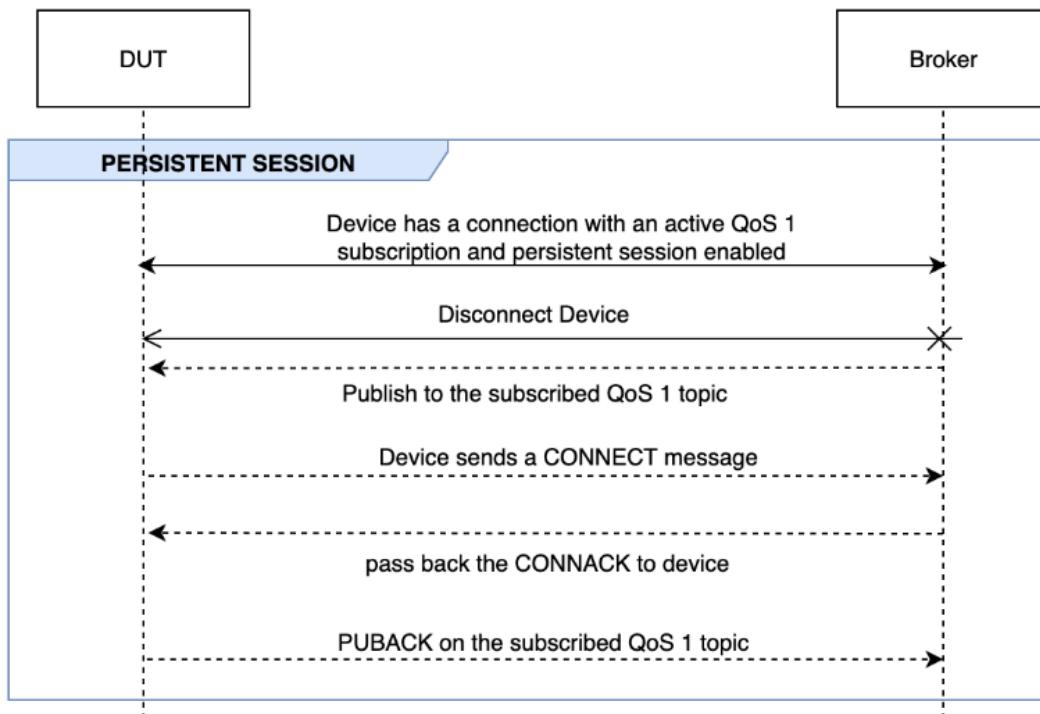
Note

Sélectionnez ce scénario uniquement si votre appareil est capable d'exécuter des abonnements QoS 1 et de maintenir une session persistante.

Ce scénario valide le comportement de l'appareil lors du maintien de sessions persistantes. Le test est validé lorsque les conditions suivantes sont réunies :

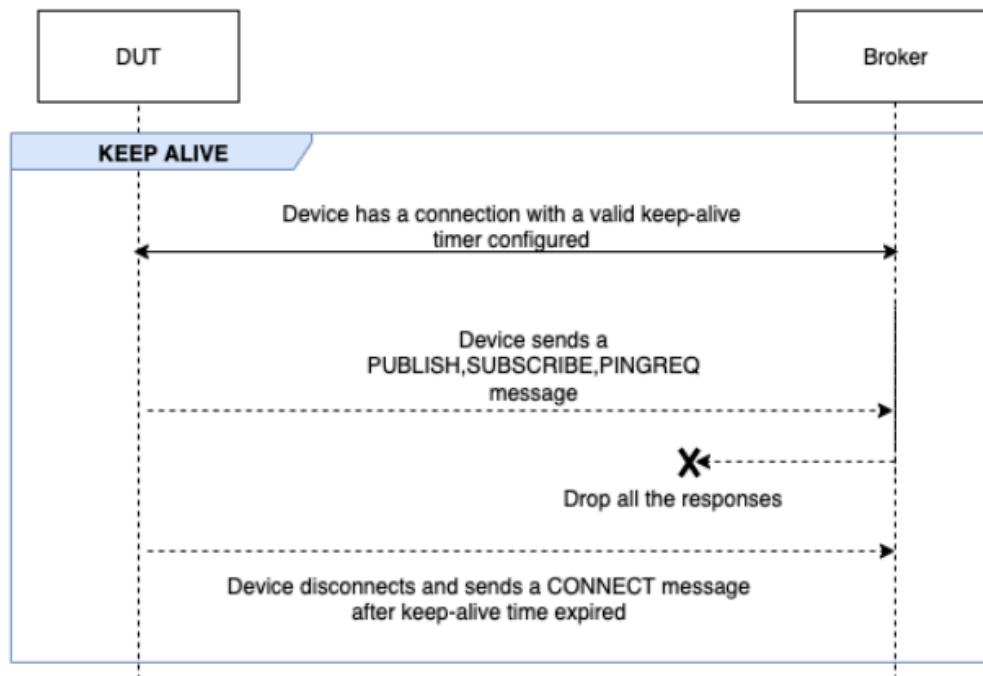
- L'appareil se connecte au courtier avec un abonnement QoS 1 actif et des sessions persistantes activées.
- L'appareil se déconnecte correctement du courtier pendant la session.
- L'appareil se reconnecte au courtier et reprend les abonnements à ses sujets déclencheurs sans se réabonner explicitement à ces sujets.
- L'appareil reçoit correctement les messages stockés par le courtier pour les rubriques auxquelles il est abonné et s'exécute comme prévu.

Pour plus d'informations sur les sessions AWS IoT persistantes, consultez la section [Utilisation des sessions persistantes MQTT](#).



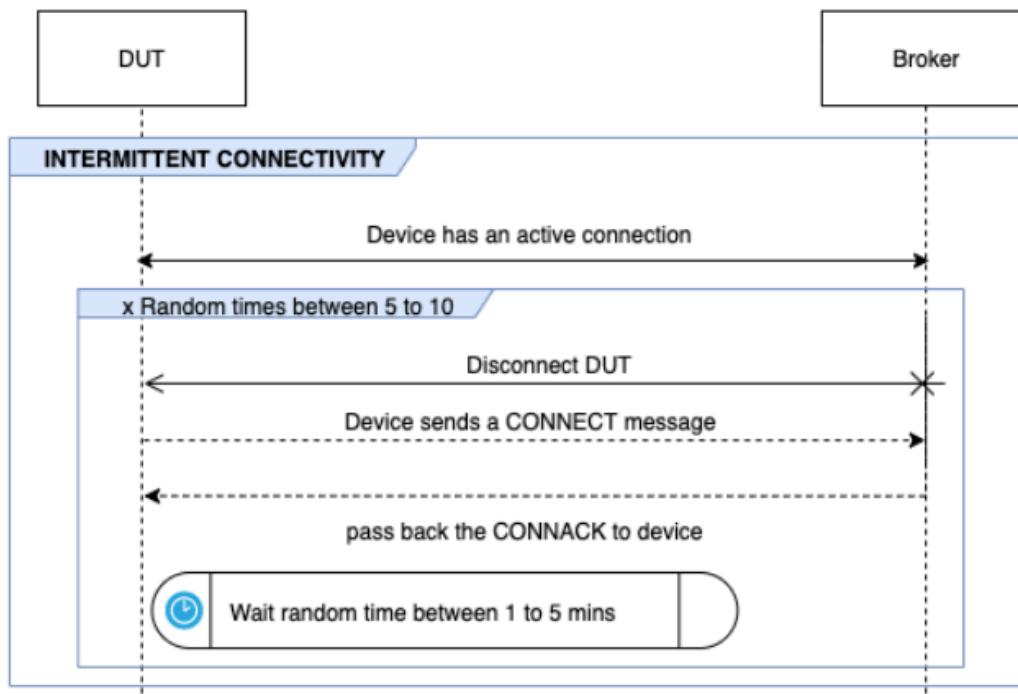
RESTER EN VIE

Ce scénario permet de vérifier si l'appareil se déconnecte correctement s'il n'a pas reçu de réponse ping du courtier. Un temporisateur de maintien en vie valide doit être configuré pour la connexion. Dans le cadre de ce test, le courtier bloque toutes les réponses envoyées et SUBSCRIBE les PINGREQ messages. PUBLISH Il vérifie également si l'appareil testé déconnecte la connexion MQTT.



CONNECTIVITÉ INTERMITTENTE

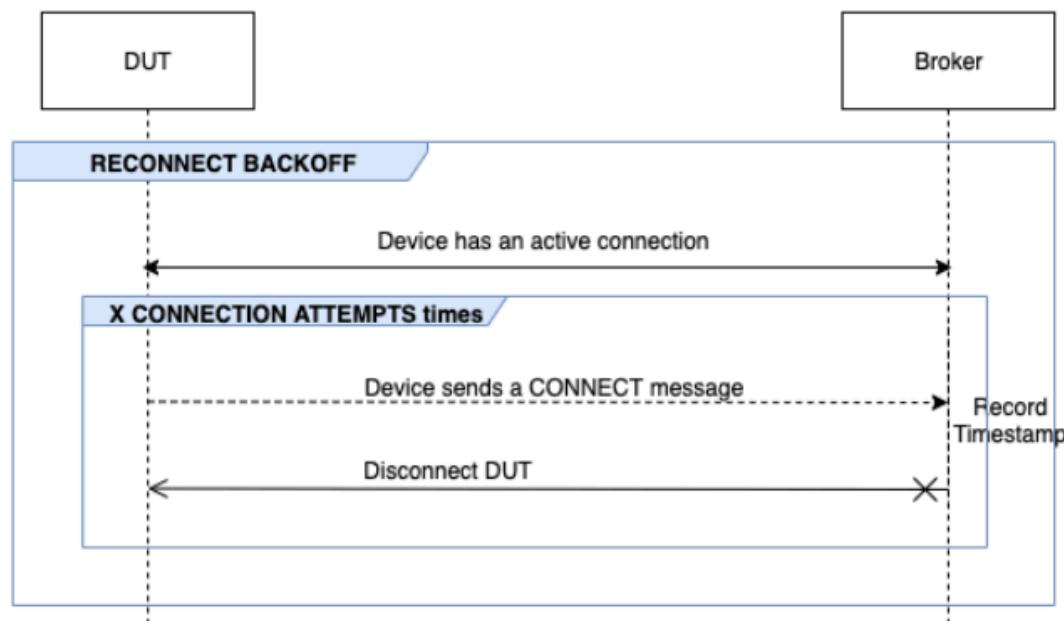
Ce scénario permet de vérifier si l'appareil peut se reconnecter au courtier une fois que celui-ci l'a déconnecté à des intervalles aléatoires pendant une période de temps aléatoire.



RECONNECTER LE BACKOFF

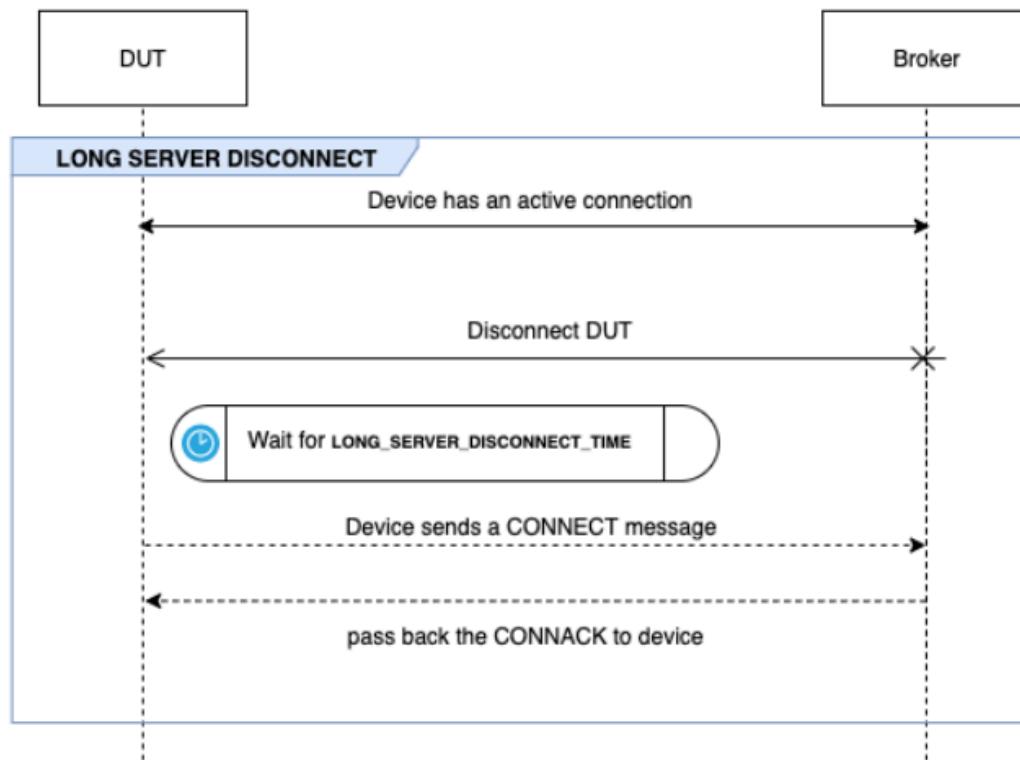
Ce scénario permet de vérifier si l'appareil dispose d'un mécanisme de sauvegarde mis en œuvre lorsque le courtier s'en déconnecte plusieurs fois. Device Advisor indique que le type de décélération est exponentiel, instable, linéaire ou constant. Le nombre de tentatives d'annulation est configurable à l'aide de BACKOFF_CONNECTION_ATTEMPTS cette option. La valeur par défaut est 5. La valeur est configurable entre 5 et 10.

Pour réussir ce test, nous vous recommandons d'implémenter le mécanisme [Exponential Backoff And Jitter](#) sur l'appareil testé.



LONGUE DÉCONNEXION DU SERVEUR

Ce scénario permet de vérifier si l'appareil peut se reconnecter correctement après que le courtier l'ait déconnecté pendant une longue période (jusqu'à 120 minutes). L'heure de déconnexion du serveur peut être configurée à l'aide de l'`LONG_SERVER_DISCONNECT_TIME` option. La valeur par défaut est de 120 minutes. Cette valeur est configurable entre 30 et 120 minutes.



Temps d'exécution supplémentaire

Le temps d'exécution supplémentaire est le temps que le test attend après avoir terminé tous les tests ci-dessus et avant de terminer le scénario de test. Les clients utilisent cette période supplémentaire pour surveiller et enregistrer toutes les communications entre l'appareil et le courtier. Le temps d'exécution supplémentaire peut être configuré à l'aide de l'OPTIONAL_EXECUTION_TIME option. Par défaut, cette option est définie sur 0 minute et peut être comprise entre 0 et 120 minutes.

Options de configuration des tests de longue durée MQTT

Toutes les options de configuration fournies pour le test de longue durée du MQTT sont facultatives. Les options suivantes sont disponibles :

OPÉRATIONS

La liste des opérations effectuées par le périphérique, telles que CONNECT, PUBLISH et SUBSCRIBE. Le scénario de test exécute des scénarios basés sur les opérations spécifiées. Les opérations qui ne sont pas spécifiées sont considérées comme valides.

```
{  
    "OPERATIONS": ["PUBLISH", "SUBSCRIBE"]  
    //by default the test assumes device can CONNECT  
}
```

SCÉNARIOS

Sur la base des opérations sélectionnées, le scénario de test exécute des scénarios pour valider le comportement de l'appareil. Il existe deux types de scénarios :

- Les scénarios de base sont des tests simples qui permettent de vérifier si le périphérique peut effectuer les opérations sélectionnées ci-dessus dans le cadre de la configuration. Ils sont présélectionnés en fonction des opérations spécifiées dans la configuration. Aucune saisie supplémentaire n'est requise dans la configuration.
- Les scénarios avancés sont des scénarios plus complexes qui sont exécutés sur l'appareil afin de vérifier si celui-ci suit les meilleures pratiques dans des conditions réelles. Ils sont facultatifs et peuvent être transmis sous forme de tableau de scénarios à l'entrée de configuration de la suite de tests.

```
{
    "SCENARIOS": [
        // list of advanced scenarios
        "PUBACK_QOS_1",
        "RECEIVE_LARGE_PAYLOAD",
        "PERSISTENT_SESSION",
        "KEEP_ALIVE",
        "INTERMITTENT_CONNECTIVITY",
        "RECONNECT_BACK_OFF",
        "LONG_SERVER_DISCONNECT"
    ]
}
```

DÉLAI D'EXPIRATION DES TESTS DE BASE :

Durée maximale pendant laquelle le scénario de test attendra la fin de tous les tests de base. La valeur par défaut est de 60 minutes. Cette valeur est configurable entre 30 et 120 minutes.

LONGUE DURÉE DE DÉCONNEXION DU SERVEUR :

Le temps nécessaire au scénario de test pour déconnecter et reconnecter le périphérique pendant le test de déconnexion prolongée du serveur. La valeur par défaut est de 60 minutes. Cette valeur est configurable entre 30 et 120 minutes.

DELE_D'EXECUTION_DEL'EXECUTION SUPPLÉMENTAIRE :

La configuration de cette option fournit une fenêtre de temps après la fin de tous les tests, afin de surveiller les événements entre l'appareil et le courtier. La valeur par défaut est de 0 minutes. Cette valeur est configurable entre 0 et 120 minutes.

BACKOFF_CONNECTION_ATTEMPTS :

Cette option configure le nombre de fois que l'appareil est déconnecté par le scénario de test. Ceci est utilisé par le test Reconnect Backoff. La valeur par défaut est de 5 tentatives. Cette valeur est configurable entre 5 et 10.

FORMAT DE CHARGE UTILE LONGUE :

Format de la charge utile du message que l'appareil attend lorsque le scénario de test est publié dans une rubrique QoS 1 à laquelle l'appareil a souscrit.

Définition du scénario de test de l'API :

```
{
    "tests": [
        {
            "name": "my_mqtt_long_duration_test",
            "configuration": {
                // optional
                "OPERATIONS": ["PUBLISH", "SUBSCRIBE"],
                "SCENARIOS": [
                    "LONG_SERVER_DISCONNECT",
                    "RECONNECT_BACK_OFF",

```

```
        "KEEP_ALIVE",
        "RECEIVE_LARGE_PAYLOAD",
        "INTERMITTENT_CONNECTIVITY",
        "PERSISTENT_SESSION",
    ],
    "BASIC_TESTS_EXECUTION_TIMEOUT": 60, // in minutes (60 minutes by default)
    "LONG_SERVER_DISCONNECT_TIME": 60, // in minutes (120 minutes by default)
    "ADDITIONAL_EXECUTION_TIME": 60, // in minutes (0 minutes by default)
    "BACKOFF_CONNECTION_ATTEMPTS": "5",
    "LONG_PAYLOAD_FORMAT":{"message":"${payload}"}}"
},
"test":{
    "id":"MQTT_Long_Duration",
    "version":"0.0.0"
}
}
]
```

Journal récapitulatif des cas de test de longue durée MQTT

Le scénario de test de longue durée MQTT dure plus longtemps que les scénarios de test classiques. Un journal récapitulatif distinct est fourni, qui répertorie les événements importants tels que les connexions des appareils, la publication et l'abonnement pendant l'exécution. Les détails incluent ce qui a été testé, ce qui n'a pas été testé et ce qui a échoué. À la fin du journal, le test inclut un résumé de tous les événements survenus pendant l'exécution du scénario de test. Cela comprend :

- Le minuteur Keep Alive est configuré sur l'appareil.
- Indicateur de session persistante configuré sur l'appareil.
- Le nombre de connexions de périphériques pendant le test.
- Le type d'interruption de la reconnexion de l'appareil, s'il est validé pour le test d'interruption de reconnexion.
- Les sujets sur lesquels l'appareil a publié, lors de l'exécution du scénario de test.
- Les sujets auxquels l'appareil s'est abonné lors de l'exécution du scénario de test.

AWS IoT CoreEmplacement de l'appareil

Avant d'utiliser la fonction de localisation de l'AWS IoT Coreappareil, consultez les conditions générales de cette fonctionnalité. Notez que vous AWS pouvez transmettre les paramètres de votre demande de recherche de géolocalisation, tels que les données de localisation utilisées pour effectuer les recherches, et d'autres informations au fournisseur de données tiers que vous avez choisi, qui peut ne pas être celui Région AWS que vous utilisez actuellement. Pour plus d'informations, consultez [Conditions de service AWS](#).

Utilisez AWS IoT Core Device Location pour tester la localisation de vos appareils IoT à l'aide de solveurs tiers. Les solveurs sont des algorithmes fournis par des fournisseurs tiers qui résolvent les données de mesure et estiment la position de votre appareil. En identifiant l'emplacement de vos appareils, vous pouvez les suivre et les corriger sur le terrain afin de résoudre d'éventuels problèmes.

Les données de mesure collectées à partir de diverses sources sont résolues et les informations de géolocalisation sont signalées sous forme de charge utile [GeoJSON](#). Le format GeoJSON est un format utilisé pour coder des structures de données géographiques. La charge utile contient les coordonnées de latitude et de longitude de la position de votre appareil, qui sont basées sur le système de coordonnées du [Système géodésique mondial \(WGS84\)](#).

Rubriques

- [Types de mesures et solveurs \(p. 1234\)](#)
- [Comment fonctionne la localisation de l'AWS IoT Coreappareil \(p. 1235\)](#)
- [Comment utiliser la localisation de AWS IoT Core l'appareil \(p. 1236\)](#)
- [Résolution de la localisation des appareils IoT \(p. 1237\)](#)
- [Résolution de la localisation de l'appareil à l'aide des rubriques MQTT AWS IoT Core Device Location \(p. 1241\)](#)
- [Solveurs de localisation et charge utile de l'appareil \(p. 1246\)](#)

Types de mesures et solveurs

AWS IoT CoreDevice Location s'associe à des fournisseurs tiers pour résoudre les données de mesure et fournir une estimation de la position de l'appareil. Le tableau suivant présente les types de mesures et les solveurs de localisation tiers, ainsi que des informations sur les appareils pris en charge. Pour plus d'informations sur les périphériques LoRa WAN et la configuration de leur emplacement, reportez-vous à la section [Configurer la position des ressources sans fil avec AWS IoT for LoRa WAN \(p. 1300\)](#).

Types de mesures et solveurs

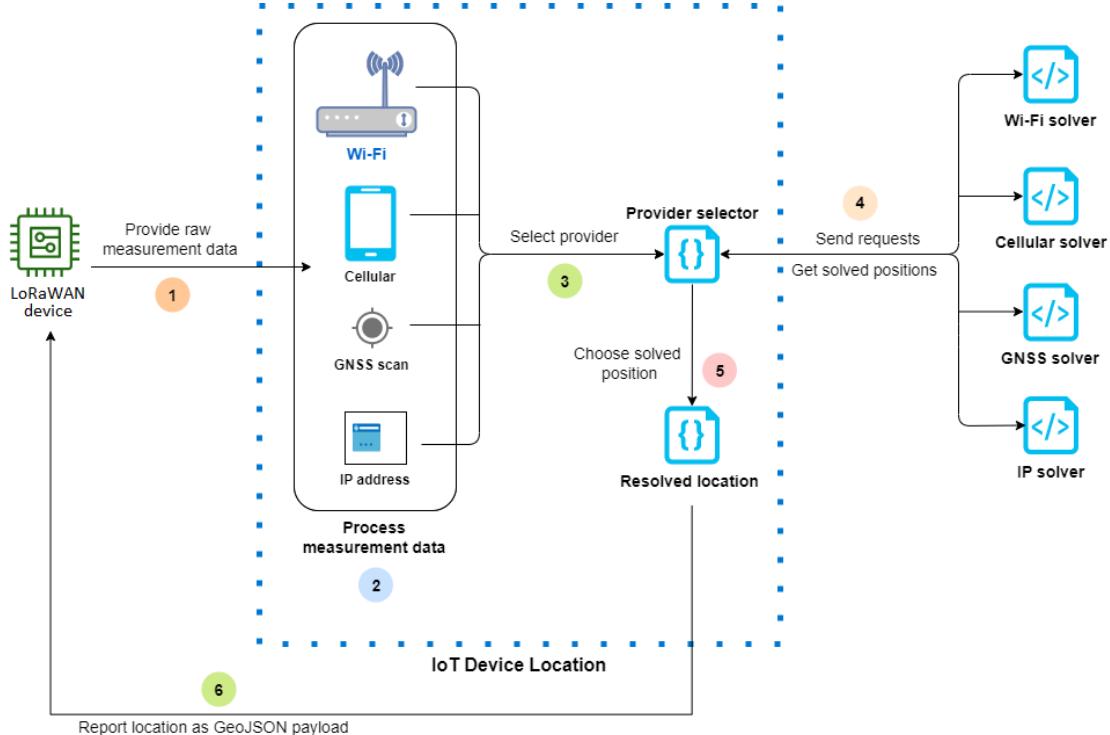
Type de mesure	Solveurs tiers	Appareils pris en charge
Points d'accès Wi-Fi	Solveur basé sur Wi-Fi	Appareils IoT généraux et appareils LoRa WAN
Tours radio cellulaires : données GSM, LTE, CDMA, SCDMA, WCDMA et TD-SCDMA	Solveur basé sur une base de données	Appareils IoT généraux et appareils LoRa WAN

Type de mesure	Solveurs tiers	Appareils pris en charge
Adresse IP	Solveur de recherche inversée IP	Appareils IoT généraux
Données de numérisation GNSS (messages NAV)	solveur GNSS	Appareils IoT généraux et appareils LoRa WAN

Pour plus d'informations sur les solveurs de localisation et des exemples illustrant la charge utile de l'appareil pour les différents types de mesures, consultez. [Solveurs de localisation et charge utile de l'appareil \(p. 1246\)](#)

Comment fonctionne la localisation de l'AWS IoT Coreappareil

Le schéma suivant montre comment AWS IoT Core Device Location collecte les données de mesure et résout les informations de localisation de vos appareils.



Les étapes suivantes montrent comment fonctionne la localisation de l'AWS IoT Coreappareil.

1. Recevoir des données de mesure

Les données de mesure brutes relatives à la position de votre appareil sont d'abord envoyées depuis celui-ci. Les données de mesure sont spécifiées sous forme de charge utile JSON.

2. Données de mesure des processus

Les données de mesure sont traitées et AWS IoT Core Device Location choisit les données de mesure à utiliser, qui peuvent être des informations Wi-Fi, cellulaires, de numérisation GNSS ou d'adresse IP.

3. Choisissez un solveur

Le solveur tiers est choisi en fonction des données de mesure. Par exemple, si les données de mesure contiennent des informations sur le Wi-Fi et l'adresse IP, il choisit le solveur Wi-Fi et le solveur de recherche inversée IP.

4. Obtenir un emplacement résolu

Une demande d'API est envoyée aux fournisseurs de solutions pour leur demander de résoudre l'emplacement. AWS IoT Core Device Location obtient ensuite les informations de géolocalisation estimées des solveurs.

5. Choisissez un emplacement résolu

Les informations de localisation résolues et leur précision sont comparées, et AWS IoT Core Device Location choisit les résultats de géolocalisation avec la plus grande précision.

6. Informations de localisation de sortie

Les informations de géolocalisation vous sont envoyées sous forme de charge utile GeoJSON. La charge utile contient les coordonnées géographiques WGS84, les informations de précision, les niveaux de confiance et l'horodatage auquel la position résolue a été obtenue.

Comment utiliser la localisation de AWS IoT Core l'appareil

Les étapes suivantes montrent comment utiliser la localisation de AWS IoT Core l'appareil.

1. Fournir des données de mesure

Spécifiez les données de mesure brutes liées à l'emplacement de votre appareil sous forme de charge utile JSON. Pour récupérer les données de mesure de la charge utile, accédez aux journaux de votre appareil ou utilisez les CloudWatch journaux et copiez les informations relatives aux données de charge utile. La charge utile JSON doit contenir un ou plusieurs types de mesures de données. Pour des exemples illustrant le format de charge utile de différents solveurs, reportez-vous à la section [Solveurs de localisation et charge utile de l'appareil \(p. 1246\)](#).

2. Résoudre les informations de localisation

À l'aide de la page [Emplacement de l'appareil](#) de la AWS IoT console ou de l'opération [GetPositionEstimate](#) API, transmettez les données de mesure de la charge utile et déterminez l'emplacement de l'appareil. AWS IoT Core Device Location choisit ensuite le solveur le plus précis et indique la position du périphérique. Pour plus d'informations, veuillez consulter [Résolution de la localisation des appareils IoT \(p. 1237\)](#).

3. Copier les informations de localisation

Vérifiez les informations de géolocalisation qui ont été résolues par AWS IoT Core Device Location et signalées sous forme de charge utile GeoJSON. Vous pouvez copier la charge utile pour l'utiliser avec vos applications et d'autres applications AWS. Par exemple, vous pouvez envoyer vos données de localisation géographique à Amazon Location Service à l'aide de l'action relative à la [Emplacement \(p. 591\)](#) AWS IoT règle.

Les rubriques suivantes montrent comment utiliser la localisation de l'AWS IoT Core appareil et des exemples de charge utile de la localisation des appareils.

- [Résolution de la localisation des appareils IoT \(p. 1237\)](#)
- [Solveurs de localisation et charge utile de l'appareil \(p. 1246\)](#)

Résolution de la localisation des appareils IoT

Utilisez AWS IoT Core Device Location pour décoder les données de mesure de vos appareils et résolvez la localisation de l'appareil à l'aide de solveurs tiers. L'emplacement résolu est généré sous forme de charge utile GeoJSON avec les coordonnées géographiques et les informations de précision. Vous pouvez résoudre l'emplacement de votre appareil à partir de la AWS IoT console, de l'AWS IoT WirelessAPI ou AWS CLI.

Rubriques

- [Résolution de la localisation de l'appareil \(console\) \(p. 1237\)](#)
- [Résolution de la localisation de l'appareil \(API\) \(p. 1239\)](#)
- [Résolution des erreurs lors de la résolution de l'emplacement \(p. 1240\)](#)

Résolution de la localisation de l'appareil (console)

Pour résoudre l'emplacement de l'appareil (console)

1. Accédez à la page [Emplacement de l'appareil](#) dans la AWS IoT console.
2. Obtenez les données de mesure de la charge utile à partir des journaux de votre appareil ou des CloudWatch journaux, et saisissez-les dans la section Entrer les données de charge utile au format JSON.

Le code suivant montre un exemple de charge utile JSON. La charge utile contient des données de mesure cellulaires et Wi-Fi. Si votre charge utile contient d'autres types de données de mesure, le solveur le plus précis sera utilisé. Pour plus d'informations, consultez[the section called “Solveurs de localisation et charge utile de l'appareil” \(p. 1246\)](#).

Note

La charge utile JSON doit contenir au moins un type de données de mesure.

```
{  
    "Timestamp": 1664313161,  
    "Ip": {  
        "IpAddress": "54.240.198.35"  
    },  
    "WiFiAccessPoints": [ {  
        "MacAddress": "A0:EC:F9:1E:32:C1",  
        "Rss": -77  
    } ],  
    "CellTowers": {  
        "Gsm": [ {  
            "Mcc": 262,  
            "Mnc": 1,  
            "Lac": 5126,  
            "GeranCid": 16504,  
            "GsmLocalId": {  
                "Bsic": 6,  
                "Bcch": 82  
            },  
            "GsmTimingAdvance": 1,  
            "RxLevel": -110,  
            "GsmNmz": [ {  
                "Bsic": 7,  
                "Bcch": 85,  
                "RxLevel": -100,  
                "GlobalIdentity": {  
                    "Lac": 5126,  
                    "CellId": 16504,  
                    "Bsic": 6,  
                    "Bcch": 82  
                }  
            } ]  
        } ]  
    }  
}
```

```
        "Lac": 1,
        "GeranCid": 1
    }
}
},
" Wcdma": [
    "Mcc": 262,
    "Mnc": 7,
    "Lac": 65535,
    "UtranCid": 14674663,
    "WcdmaNmr": [
        "Uarfcndl": 10786,
        "UtranCid": 14674663,
        "Psc": 149
    },
    {
        "Uarfcndl": 10762,
        "UtranCid": 14674663,
        "Psc": 211
    }
]
},
" Lte": [
    "Mcc": 262,
    "Mnc": 2,
    "EutranCid": 2898945,
    "Rsrp": -50,
    "Rsrq": -5,
    "LteNmr": [
        "Earfcn": 6300,
        "Pci": 237,
        "Rsrp": -60,
        "Rsrq": -6,
        "EutranCid": 2898945
    },
    {
        "Earfcn": 6300,
        "Pci": 442,
        "Rsrp": -70,
        "Rsrq": -7,
        "EutranCid": 2898945
    }
]
}
]
```

3. Pour résoudre les informations de localisation, choisissez Résoudre.

Les informations de localisation sont de type blob et sont renvoyées sous forme de charge utile utilisant le format GeoJSON, qui est un format utilisé pour coder les structures de données géographiques. La charge utile contient :

- Les coordonnées géographiques WGS84, qui incluent les informations de latitude et de longitude. Il peut également inclure des informations d'altitude.
 - Type d'informations de localisation signalées, par exemple Point. Un type d'emplacement de point représente l'emplacement sous la forme d'une latitude et d'une longitude WGS84, codées sous la forme d'un point [GeoJSON](#).
 - Les informations de précision horizontale et verticale, qui indiquent la différence entre les informations de localisation estimées par les solveurs et la position réelle du périphérique.
 - Le niveau de confiance, qui indique l'incertitude liée à la réponse d'estimation de l'emplacement. La valeur par défaut est 0,68, ce qui indique une probabilité de 68 % que la position réelle de l'appareil se trouve dans le rayon d'incertitude de la position estimée.

- La ville, l'État, le pays et le code postal où se trouve l'appareil. Ces informations ne seront communiquées que lorsque le solveur de recherche inversée IP est utilisé.
- Les informations d'horodatage, qui correspondent à la date et à l'heure auxquelles la localisation a été résolue. Il utilise le format d'horodatage Unix.

Le code suivant montre un exemple de charge utile GeoJSON renvoyé en résolvant l'emplacement.

Note

Si AWS IoT Core Device Location signale des erreurs lors de la tentative de résolution de l'emplacement, vous pouvez résoudre les erreurs et résoudre le problème de localisation. Pour plus d'informations, veuillez consulter [Résolution des erreurs lors de la résolution de l'emplacement \(p. 1240\)](#).

```
{  
    "coordinates": [  
        13.376076698303223,  
        52.51823043823242  
    ],  
    "type": "Point",  
    "properties": {  
        "verticalAccuracy": 45,  
        "verticalConfidenceLevel": 0.68,  
        "horizontalAccuracy": 303,  
        "horizontalConfidenceLevel": 0.68,  
        "country": "USA",  
        "state": "CA",  
        "city": "Sunnyvalue",  
        "postalCode": "91234",  
        "timestamp": "2022-11-18T12:23:58.189Z"  
    }  
}
```

4. Accédez à la section Emplacement des ressources et vérifiez les informations de géolocalisation signalées par AWS IoT Core Device Location. Vous pouvez copier la charge utile pour l'utiliser avec d'autres applications et Service AWS s. Par exemple, vous pouvez utiliser le [Emplacement \(p. 591\)](#) pour envoyer vos données de localisation géographique à Amazon Location Service.

Résolution de la localisation de l'appareil (API)

Pour résoudre l'emplacement de l'appareil à l'aide de l'AWS IoT WirelessAPI, utilisez l'opération [GetPositionEstimate](#)API ou la commande [get-position-estimate](#)CLI. Spécifiez les données de mesure de la charge utile en entrée et exécutez l'opération d'API pour déterminer l'emplacement de l'appareil.

Note

L'opération de l'GetPositionEstimateAPI ne stocke aucune information sur l'appareil ou son état et ne peut pas être utilisée pour récupérer des données de localisation historiques. Il effectue une opération unique qui résout les données de mesure et produit la position estimée. Pour récupérer les informations de localisation, vous devez spécifier les informations de charge utile chaque fois que vous effectuez cette opération d'API.

La commande suivante montre comment résoudre l'emplacement à l'aide de cette opération d'API.

Note

Lorsque vous exécutez la commande [get-position-estimate](#) CLI, vous devez spécifier le fichier JSON de sortie comme première entrée. Ce fichier JSON stockera les

informations de localisation estimées obtenues en réponse de la CLI au format GeoJSON.
Par exemple, la commande suivante enregistre les informations de localisation dans le fichier *locationout.json*.

```
aws iotwireless get-position-estimate locationout.json \
--ip IpAddress="54.240.198.35" \
--wi-fi-access-points \
    MacAddress="A0:EC:F9:1E:32:C1",Rss=-75 \
    MacAddress="A0:EC:F9:15:72:5E",Rss=-67
```

Cet exemple inclut à la fois les points d'accès Wi-Fi et l'adresse IP comme types de mesure. AWS IoT Core Device Location choisit entre le solveur Wi-Fi et le solveur de recherche inversée IP, et sélectionne le solveur avec la plus grande précision.

L'emplacement résolu est renvoyé sous la forme d'une charge utile utilisant le format GeoJSON, qui est un format utilisé pour coder les structures de données géographiques. Il est ensuite stocké dans le fichier *locationout.json*. La charge utile contient les coordonnées de latitude et de longitude WGS84, les informations de précision et de niveau de confiance, le type de données de localisation et l'horodatage auquel la position a été déterminée.

```
{
  "coordinates": [
    13.37704086303711,
    52.51865005493164
  ],
  "type": "Point",
  "properties": {
    "verticalAccuracy": 707,
    "verticalConfidenceLevel": 0.68,
    "horizontalAccuracy": 389,
    "horizontalConfidenceLevel": 0.68,
    "country": "USA",
    "state": "CA",
    "city": "Sunnyvale",
    "postalCode": "91234",
    "timestamp": "2022-11-18T14:03:57.391Z"
  }
}
```

Résolution des erreurs lors de la résolution de l'emplacement

Lorsque vous tentez de résoudre l'emplacement, l'un des codes d'erreur suivants peut s'afficher. AWS IoT Core L'emplacement de l'appareil peut générer une erreur lors de l'utilisation de l'opération GetPositionEstimate API, ou bien faire référence au numéro de ligne correspondant à l'erreur dans la AWS IoT console.

- Erreur

Cette erreur indique que le format JSON de la charge utile de l'appareil ne peut pas être validé par AWS IoT Core Device Location. L'erreur peut se produire pour les raisons suivantes :

- Les données de mesure JSON ne sont pas formatées correctement.
- La charge utile contient uniquement les informations d'horodatage.
- Les paramètres des données de mesure, tels que l'adresse IP, ne sont pas valides.

Pour résoudre cette erreur, vérifiez si votre fichier JSON est correctement formaté et s'il contient des données provenant d'un ou de plusieurs types de mesures en entrée. Si l'adresse IP n'est pas valide,

pour plus d'informations sur la manière de fournir une adresse IP valide afin de résoudre l'erreur, consultez [Solveur de recherche inversée IP \(p. 1251\)](#).

- Erreur

Cette erreur indique que vous n'êtes pas autorisé à effectuer l'opération d'API ou à utiliser la AWS IoT console pour récupérer l'emplacement de l'appareil. Pour résoudre cette erreur, vérifiez que vous disposez des autorisations requises pour effectuer cette action. Cette erreur peut se produire si votre AWS Management Console session ou votre jeton de AWS CLI session a expiré. Pour résoudre cette erreur, actualisez le jeton de session pour utiliser le AWS CLI, ou déconnectez-vous du, AWS Management Console puis connectez-vous à l'aide de vos informations d'identification.

- Erreur

Cette erreur indique qu'aucune information de localisation n'a été trouvée ou résolue par AWS IoT Core Device Location. L'erreur peut se produire en raison de cas tels que des données insuffisantes dans la saisie des données de mesure. Par exemple :

- L'adresse MAC ou les informations de la tour de téléphonie mobile ne sont pas suffisantes.
- L'adresse IP n'est pas disponible pour rechercher et récupérer l'emplacement.
- La charge utile du GNSS n'est pas suffisante.

Pour résoudre l'erreur dans de tels cas, vérifiez si vos données de mesure contiennent suffisamment d'informations pour déterminer la position de l'appareil.

- Erreur

Cette erreur indique qu'une exception interne au serveur s'est produite lorsque AWS IoT Core Device Location a tenté de résoudre l'emplacement. Pour tenter de corriger cette erreur, actualisez la session et réessayez d'envoyer les données de mesure à résoudre.

Résolution de la localisation de l'appareil à l'aide des rubriques MQTT AWS IoT Core Device Location

Vous pouvez utiliser des rubriques MQTT réservées pour obtenir les dernières informations de localisation de vos appareils grâce à la fonction de localisation des AWS IoT Core appareils.

La publication et l'abonnement à des sujets relatifs à la localisation des appareils nécessitent une autorisation par sujet. AWS IoT se réserve le droit d'ajouter de nouveaux sujets à la structure de sujets existante. C'est pour cette raison que nous vous recommandons d'éviter les abonnements de type générique aux sujets de la localisation des appareils. Par exemple, évitez de vous abonner à des filtres thématiques, \$aws/device/location/*customer_device_id*/get_position_estimate/# car le nombre de sujets correspondant à ce filtre de sujets peut augmenter à mesure que de nouveaux sujets relatifs à AWS IoT la localisation des appareils sont introduits.

Format de localisation de l'appareil (rubriques MQTT)

Les rubriques réservées à la localisation de l'AWS IoT Core appareil utilisent le préfixe suivant :

\$aws/device/location/{*customer_device_id*}/

Pour créer un sujet complet, remplacez-le *customer_device_id* d'abord par votre identifiant unique que vous utilisez pour identifier votre appareil. Nous vous recommandons de spécifier leWirelessDeviceId, par exemple dans le cas des appareils LoRa WAN et Sidewalk, et *thingName* si votre appareil est enregistré en tant qu'AWS IoT objet. Vous ajoutez ensuite le sujet avec le talon du

sujet, `get_position_estimate/accepted` comme `get_position_estimate` indiqué dans la section suivante.

Note

Ces noms ne `{customer_device_id}` peuvent contenir que des lettres minuscules, des chiffres et des traits d'union. Lorsque vous vous abonnez à des rubriques relatives à la localisation des appareils, vous pouvez utiliser le signe plus (+) comme caractère générique pour qu'il corresponde `{customer_device_id}` à n'importe quel identifiant d'appareil.

Les rubriques réservées utilisées pour interagir avec la localisation de l'AWS IoT Core appareil sont les suivantes.

Rubriques MQTT relatives à la localisation des appareils

Sujet	Opérations autorisées	Description
<code>\$aws/device_location/ customer_device_id / get_position_estimate</code>	Publier	Un appareil publie des informations sur cette rubrique pour obtenir les données de mesure brutes numérisées à résoudre par AWS IoT Core Device Location.
<code>\$aws/device_location/ customer_device_id / get_position_estimate/ accepted</code>	S'abonner	AWS IoT Core La localisation de l'appareil publie les informations de localisation dans cette rubrique lorsqu'elle parvient à résoudre l'emplacement de l'appareil.
<code>\$aws/device_location/ customer_device_id / get_position_estimate/ rejected</code>	S'abonner	AWS IoT Core La localisation de l'appareil publie les informations d'erreur dans cette rubrique lorsqu'elle ne parvient pas à résoudre l'emplacement de l'appareil.

Politique relative à la localisation des appareils (sujets MQTT)

Pour recevoir des messages relatifs aux rubriques relatives à la localisation de l'appareil, votre appareil doit utiliser une politique lui permettant de se connecter à la passerelle de l'AWS IoT appareil et de s'abonner aux rubriques MQTT.

Vous trouverez ci-dessous un exemple de la politique requise pour recevoir des messages concernant les différents sujets.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Receive"
      ],
      "Resource": [
        "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/accepted"
      ]
    }
  ]
}
```

```
        ],
        "Resource": [
            "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/accepted",
            "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/rejected"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Subscribe"
        ],
        "Resource": [
            "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/accepted",
            "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/rejected"
        ]
    }
}
```

Rubriques relatives à la localisation des appareils et charge utile

La section suivante présente les rubriques relatives à la localisation des AWS IoT Core appareils, le format de la charge utile des messages et un exemple de politique pour chaque rubrique.

Rubriques

- [/get_position_estimate \(p. 1243\)](#)
- [/get_position_estimate/accepted \(p. 1244\)](#)
- [/get_position_estimate/rejeté \(p. 1245\)](#)

/get_position_estimate

Publiez un message sur cette rubrique pour obtenir les données de mesure brutes de l'appareil à résoudre en fonction de la localisation de AWS IoT Core l'appareil.

```
$aws/device_location/customer_device_id/get_position_estimate/
```

AWS IoT CoreDevice Location répond en publiant sur [/get_position_estimate/accepted \(p. 1244\)](#) ou [/get_position_estimate/rejeté \(p. 1245\)](#).

Charge utile des messages

Le format de la charge utile du message suit une structure similaire à celle du corps de la demande d'opération de l'AWS IoT WirelessAPI. [GetPositionEstimate](#) Il contient :

- **Timestamp**Chaîne facultative qui correspond à la date et à l'heure auxquelles la localisation a été résolue.
- **MessageId**Chaîne facultative, qui peut être utilisée pour mapper la demande à la réponse. Si vous spécifiez cette chaîne, le message publié dans les `get_position_estimate/rejected` rubriques `get_position_estimate/accepted` ou contiendra cette chaîne**MessageId**.
- Les données de mesure de l'appareil qui contient un ou plusieurs des types de mesure suivants :
 - [WiFiAccessPoint](#)

- [CellTowers](#)
- [IpAddress](#)
- [Gnss](#)

Voici un exemple de charge utile de la charge utile des messages.

```
{  
    "Timestamp": "1664313161",  
    "MessageId": "ABCD1",  
    "WiFiAccessPoints": [  
        {  
            "MacAddress": "A0:EC:F9:1E:32:C1"  
            "Rss": -66  
        }  
    ],  
    "Ip": {  
        "IpAddress": "54.192.168.0"  
    },  
    "Gnss": {  
        "Payload":  
        "0178b6a3a652c324a880d2e927446d528fd113fa0770a84e25c27a0fd051358961a21d00a2fa2f0c577e2063b5b3429ef8703  
        "CaptureTime": 1330798212.8811495  
    }  
}
```

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate"  
            ]  
        }  
    ]  
}
```

/get_position_estimate/accepted

AWS IoT CoreDevice Location publie une réponse à cette rubrique lorsque vous renvoyez les informations de localisation résolues pour votre appareil. Les informations de localisation sont renvoyées au format [GeoJSON](#).

```
$aws/device_location/customer_device_id/get_position_estimate/accepted
```

Ce qui suit montre la charge utile du message et un exemple de politique.

Charge utile des des messages

Voici un exemple de la charge utile des messages au format GeoJSON. Si vous avez indiqué un MessageId lors de la fourniture de vos données de mesure brutes et que AWS IoT Core Device Location

a correctement résolu les informations de localisation, les mêmes MessageId informations seront renvoyées dans la charge utile du message.

```
{  
    "coordinates": [  
        13.37704086303711,  
        52.51865005493164  
    ],  
    "type": "Point",  
    "properties": {  
        "verticalAccuracy": 707,  
        "verticalConfidenceLevel": 0.68,  
        "horizontalAccuracy": 389,  
        "horizontalConfidenceLevel": 0.68,  
        "country": "USA",  
        "state": "CA",  
        "city": "Sunnyvalue",  
        "postalCode": "91234",  
        "timestamp": "2022-11-18T14:03:57.391Z",  
        "messageId": "ABCD1"  
    }  
}
```

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/accepted"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/accepted"  
            ]  
        }  
    ]  
}
```

/get_position_estimate/rejeté

AWS IoT CoreLa localisation de l'appareil publie une réponse d'erreur à cette rubrique lorsqu'elle ne parvient pas à résoudre l'emplacement de l'appareil.

```
$aws/device_location/customer_device_id/get_position_estimate/rejected
```

Ce qui suit montre la charge utile du message et un exemple de politique. Pour plus d'informations sur les erreurs, reportez-vous à la section [Résolution des erreurs lors de la résolution de l'emplacement \(p. 1240\)](#).

Charge utile des messages

Voici un exemple de la charge utile du message qui fournit le code et le message d'erreur indiquant pourquoi AWS IoT Core Device Location n'a pas réussi à résoudre les informations de localisation. Si vous avez indiqué un MessageId lors de la fourniture de vos données de mesure brutes et que AWS IoT Core Device Location ne parvient pas à résoudre les informations de localisation, les mêmes MessageId informations seront renvoyées dans la charge utile du message.

```
{  
    "errorCode": 500,  
    "errorMessage": "Failed to get position estimate due to internal server error",  
    " messageId": "ABCD1"  
}
```

Exemple de stratégie

Voici un exemple de document de stratégie requise :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/rejected"  
            ]  
        },  
        {  
            "Action": [  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:topic/$aws/device_location/customer_device_id/get_position_estimate/rejected"  
            ]  
        }  
    ]  
}
```

Solveurs de localisation et charge utile de l'appareil

Les solveurs de localisation sont des algorithmes qui peuvent être utilisés pour résoudre la localisation de vos appareils IoT. AWS IoT Core Device Location prend en charge les solveurs de localisation suivants. Vous verrez des exemples du format de charge utile JSON pour ces types de mesures, les appareils pris en charge par le solveur et la manière dont la localisation est résolue.

Pour déterminer l'emplacement de l'appareil, spécifiez un ou plusieurs de ces types de données de mesure. Un seul emplacement résolu sera renvoyé pour toutes les données de mesure combinées.

Rubriques

- [Solveur basé sur Wi-Fi \(p. 1247\)](#)
- [Solveur basé sur la technologie cellulaire \(p. 1247\)](#)

- [Solveur de recherche inversée IP \(p. 1251\)](#)
- [solveur GNSS \(p. 1251\)](#)

Solveur basé sur Wi-Fi

Utilisez le solveur Wi-Fi pour résoudre l'emplacement à l'aide des informations numérisées provenant des points d'accès Wi-Fi. Le solveur prend en charge la technologie WLAN et peut être utilisé pour calculer l'emplacement des appareils pour les appareils IoT généraux et les appareils sans fil LoRa WAN.

Les appareils LoRa WAN doivent être équipés du chipset LoRa Edge, qui peut décoder les informations de numérisation Wi-Fi entrantes. LoRaEdge est une plateforme à très faible consommation qui intègre un LoRa émetteur-récepteur longue portée, un scanner GNSS multi-constellation et un scanner MAC Wi-Fi passif ciblant les applications de géolocalisation. Lorsqu'un message de liaison montante est reçu de l'appareil, les données de numérisation Wi-Fi sont envoyées à l'emplacement de l'AWS IoT Core appareil et l'emplacement est estimé en fonction des résultats de l'analyse Wi-Fi. Les informations décodées sont ensuite transmises au solveur basé sur le Wi-Fi pour récupérer les informations de localisation.

Exemple de charge utile d'un solveur basé sur le Wi-Fi

Le code suivant montre un exemple de charge utile JSON provenant de l'appareil qui contient les données de mesure. Lorsque AWS IoT Core Device Location reçoit ces données en entrée, il envoie une requête HTTP au fournisseur du solveur pour résoudre les informations de localisation. Pour récupérer les informations, spécifiez les valeurs de l'adresse MAC et du RSS (puissance du signal reçu). Pour ce faire, fournissez la charge utile JSON à l'aide de ce format ou utilisez le paramètre [WiFiAccessPointsobject](#) de l'opération [GetPositionEstimateAPI](#).

```
{  
    "Timestamp": 1664313161,      // optional  
    "WiFiAccessPoints": [  
        {  
            "MacAddress": "A0:EC:F9:1E:32:C1",  // required  
            "Rss": -75                         // required  
        }  
    ]  
}
```

Solveur basé sur la technologie cellulaire

Vous pouvez utiliser le solveur basé sur la technologie cellulaire pour résoudre la position à l'aide des données de mesure obtenues à partir des tours radio cellulaires. Le solveur prend en charge les technologies suivantes. Une seule information de localisation résolue est obtenue, même si vous incluez des données de mesure issues de l'une ou de l'ensemble de ces technologies.

- GSM
- CDMA
- WCDMA
- TD-SCDMA
- LTE

Exemples de charge utile de solveur basé sur la technologie cellulaire

Le code suivant montre des exemples de charge utile JSON provenant de l'appareil qui contient des données de mesure cellulaires. Lorsque AWS IoT Core Device Location reçoit ces données en entrée, il

envoie une requête HTTP au fournisseur du solveur pour résoudre les informations de localisation. Pour récupérer les informations, vous devez soit fournir la charge utile JSON en utilisant ce format dans la console, soit spécifier des valeurs pour le [CellTowers](#) paramètre de l'opération d'[GetPositionEstimate](#) API. Vous pouvez fournir les données de mesure en spécifiant les valeurs des paramètres à l'aide de l'une ou de l'ensemble de ces technologies cellulaires.

LTE (évolution à long terme)

Lorsque vous utilisez ces données de mesure, vous devez spécifier des informations telles que le réseau et le code de pays du réseau mobile, ainsi que des paramètres supplémentaires facultatifs, notamment des informations sur l'identifiant local. Le code suivant montre un exemple du format de charge utile. Pour plus d'informations sur ces paramètres, voir [Objet LTE](#).

```
{
    "Timestamp": 1664313161,           // optional
    "CellTowers": {
        "Lte": [
            {
                "Mcc": int,           // required
                "Mnc": int,           // required
                "EutranCid": int,     // required
                "Tac": int,           // optional
                "LteLocalId": {
                    "Pci": int,         // required
                    "Earfcn": int,      // required
                },
                "LteTimingAdvance": int, // optional
                "Rsrp": int,           // optional
                "Rsrq": float,          // optional
                "NrCapable": boolean, // optional
                "LteNmr": [
                    {
                        "Pci": int,       // required
                        "Earfcn": int,     // required
                        "EutranCid": int, // required
                        "Rsrp": int,       // optional
                        "Rsrq": float       // optional
                    }
                ]
            }
        ]
    }
}
```

GSM (Système mondial pour les communications mobiles)

Lorsque vous utilisez ces données de mesure, vous devez spécifier des informations telles que le code du réseau et du pays du réseau mobile, les informations de la station de base et des paramètres supplémentaires facultatifs. Le code suivant montre un exemple du format de charge utile. Pour plus d'informations sur ces paramètres, voir [Objet GSM](#).

```
{
    "Timestamp": 1664313161,           // optional
    "CellTowers": {
        "Gsm": [
            {
                "Mcc": int,           // required
                "Mnc": int,           // required
                "Lac": int,           // required
                "GeranCid": int,      // required
                "GsmLocalId": {
                    "Bsic": int,         // required
                    "Bcch": int,         // required
                }
            }
        ]
    }
}
```

```

    },
    "GsmTimingAdvance": int,           // optional
    "RxLevel": int,                  // optional
    "GsmNmr": [
        {
            "Bsic": int,             // required
            "Bcch": int,             // required
            "RxLevel": int,          // optional
            "GlobalIdentity": {
                "Lac": int,           // required
                "GeranCid": int     // required
            }
        }
    ]
}
]
}
}

```

CDMA (accès multiple par division de code)

Lorsque vous utilisez ces données de mesure, vous devez spécifier des informations telles que la puissance du signal et les informations d'identification, les informations de la station de base et des paramètres supplémentaires facultatifs. Le code suivant montre un exemple du format de charge utile. Pour plus d'informations sur ces paramètres, voir [Objet CDMA](#).

```

{
    "Timestamp": 1664313161,           // optional
    "CellTowers": [
        "Cdma": [
            {
                "SystemId": int,      // required
                "NetworkId": int,    // required
                "BaseStationId": int, // required
                "RegistrationZone": int, // optional
                "CdmaLocalId": [
                    "PnOffset": int,   // required
                    "CdmaChannel": int // required
                ],
                "PilotPower": int,    // optional
                "BaseLat": float,    // optional
                "BaseLng": float,    // optional
                "CdmaNmr": [
                    {
                        "PnOffset": int,   // required
                        "CdmaChannel": int, // required
                        "PilotPower": int, // optional
                        "BaseStationId": int // optional
                    }
                ]
            }
        ]
    }
}

```

WCDMA (accès multiple par division de code à large bande)

Lorsque vous utilisez ces données de mesure, vous devez spécifier des informations telles que le code du réseau et du pays, la puissance du signal et les informations d'identification, les informations de la station de base et des paramètres supplémentaires facultatifs. Le code suivant montre un exemple du format de charge utile. Pour plus d'informations sur ces paramètres, voir [Objet CDMA](#).

```

{
    "Timestamp": 1664313161,           // optional
}

```

```

"CellTowers": [
    "Wcdma": [
        {
            "Mcc": int,                                // required
            "Mnc": int,                                // required
            "UtranCid": int,                            // required
            "Lac": int,                                 // optional
            "WcdmaLocalId": [
                "Uarfcndl": int,                         // required
                "Psc": int,                               // required
            ],
            "Rscp": int,                                // optional
            "Pathloss": int,                            // optional
            "WcdmaNmr": [
                {
                    "Uarfcndl": int,                     // required
                    "Psc": int,                           // required
                    "UtranCid": int,                      // required
                    "Rscp": int,                           // optional
                    "Pathloss": int,                      // optional
                }
            ]
        ]
    ]
}

```

TD-SCDMA (accès multiple par division de code synchrone par division temporelle)

Lorsque vous utilisez ces données de mesure, vous devez spécifier des informations telles que le code du réseau et du pays, la puissance du signal et les informations d'identification, les informations de la station de base et des paramètres supplémentaires facultatifs. Le code suivant montre un exemple du format de charge utile. Pour plus d'informations sur ces paramètres, voir [Objet CDMA](#).

```

{
    "Timestamp": 1664313161,                  // optional
    "CellTowers": [
        "Tdscdma": [
            {
                "Mcc": int,                                // required
                "Mnc": int,                                // required
                "UtranCid": int,                            // required
                "Lac": int,                                 // optional
                "TdscdmaLocalId": [
                    "Uarfcn": int,                          // required
                    "CellParams": int,                      // required
                ],
                "TdscdmaTimingAdvance": int,             // optional
                "Rscp": int,                                // optional
                "Pathloss": int,                            // optional
                "TdscdmaNmr": [
                    {
                        "Uarfcn": int,                      // required
                        "CellParams": int,                  // required
                        "UtranCid": int,                      // optional
                        "Rscp": int,                           // optional
                        "Pathloss": int,                      // optional
                    }
                ]
            ]
        ]
    ]
}

```

Solveur de recherche inversée IP

Vous pouvez utiliser le solveur de recherche inversée IP pour résoudre l'emplacement en utilisant l'adresse IP comme entrée. Le solveur peut obtenir les informations de localisation à partir des appareils qui ont été approvisionnés. AWS IoT Spécifiez les informations d'adresse IP à l'aide d'un format qui est soit le modèle standard IPv4 ou IPv6, soit le modèle compressé hexadécimal IPv6. Vous obtenez ensuite l'estimation de l'emplacement résolu, y compris des informations supplémentaires telles que la ville et le pays où se trouve l'appareil.

Note

En utilisant la recherche IP inversée, vous acceptez de ne pas l'utiliser dans le but d'identifier ou de localiser un domicile ou une adresse postale spécifique.

Exemple de charge utile d'un solveur de recherche inversée IP

Le code suivant montre un exemple de charge utile JSON provenant de l'appareil qui contient les données de mesure. Lorsque AWS IoT Core Device Location reçoit les informations d'adresse IP contenues dans les données de mesure, elle recherche ces informations dans la base de données du fournisseur du solveur, qui est ensuite utilisée pour résoudre les informations de localisation. Pour récupérer les informations, fournissez la charge utile JSON à l'aide de ce format ou spécifiez des valeurs pour le paramètre `Ip` de l'opération [GetPositionEstimateAPI](#).

Note

Lorsque ce solveur est utilisé, la ville, l'État, le pays et le code postal où se trouve l'appareil sont également indiqués en plus des coordonnées. Pour voir un exemple, consultez [Résolution de la localisation de l'appareil \(console\) \(p. 1237\)](#).

```
{  
    "Timestamp": 1664313161,  
    "Ip": {  
        "IpAddress": "54.240.198.35"  
    }  
}
```

solveur GNSS

Utilisez le solveur GNSS (Global Navigation Satellite System) pour récupérer l'emplacement de l'appareil à l'aide des informations contenues dans les messages de résultats du scan GNSS ou les messages NAV. Vous pouvez éventuellement fournir des informations d'assistance GNSS supplémentaires, ce qui réduit le nombre de variables que le solveur doit utiliser pour rechercher des signaux. En fournissant ces informations d'assistance, qui incluent la position, l'altitude, le temps de capture et les informations de précision, le solveur peut facilement identifier les satellites en vue et calculer la position du dispositif.

Ce solveur peut être utilisé avec des périphériques LoRa WAN et d'autres appareils qui ont été provisionnés avec AWS IoT. Pour les appareils IoT généraux, si les appareils prennent en charge l'estimation de l'emplacement à l'aide du GNSS, lorsque les informations de numérisation GNSS sont reçues de l'appareil, les émetteurs-récepteurs résolvent les informations de localisation. Pour les appareils LoRa WAN, les appareils doivent être équipés du chipset LoRa Edge. Lorsqu'un message de liaison montante est reçu du dispositif, les données de numérisation GNSS sont envoyées à AWS IoT Core for LoRaWAN et la position est estimée sur la base des résultats de numérisation des émetteurs-récepteurs.

Exemple de charge utile du solveur GNSS

Le code suivant montre un exemple de charge utile JSON provenant de l'appareil qui contient les données de mesure. Lorsque AWS IoT Core Device Location reçoit les informations de numérisation GNSS

contenant la charge utile contenue dans les données de mesure, il utilise les émetteurs-récepteurs et toutes les informations d'assistance supplémentaires incluses pour rechercher des signaux et résoudre les informations de localisation. Pour récupérer les informations, fournissez la charge utile JSON à l'aide de ce format ou spécifiez des valeurs pour le paramètre [Gnss](#) de l'opération [GetPositionEstimateAPI](#).

Note

Pour que AWS IoT Core Device Location puisse résoudre l'emplacement de l'appareil, vous devez supprimer l'octet de destination de la charge utile.

```
{  
    "Timestamp": 1664313161,                      // optional  
    "Gnss": {  
        "AssistAltitude": number,                  // optional  
        "AssistPosition": [ number ],              // optional  
        "CaptureTime": number,                    // optional  
        "CaptureTimeAccuracy": number,           // optional  
        "Payload": string,                      // required  
        "Use2DSolver": boolean                   // optional  
    }  
}
```

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 plus d'informations sur le AWS IoT Events service qui vous permet de créer des détecteurs pour surveiller vos appareils afin de détecter les pannes ou les changements de fonctionnement et de déclencher des actions lorsqu'elles se produisent, consultez [AWS IoT Events](#).

Comment les messages d'événements sont générés

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.

Politique de réception des messages relatifs aux événements

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/*"  
            ]  
        }  
    ]  
}
```

Activer les événements pour AWS IoT

Pour que les abonnés aux rubriques réservées puissent recevoir des messages, vous devez activer les messages d'événements à partir du ou à l'aide de l'API AWS Management Console ou de l'interface de ligne de commande. Pour plus d'informations sur les messages d'événement gérés par les différentes options, consultez le [Tableau des paramètres de configuration des AWS IoT événements \(p. 1254\)](#).

- Pour activer les messages d'événements, accédez à l'onglet [Paramètres](#) de la AWS IoT console puis, dans la section Messages basés sur les événements, choisissez Gérer les événements. Vous pouvez spécifier les événements que vous voulez gérer.
- Pour contrôler les types d'événements publiés à l'aide de l'API ou de la CLI,appelez [l'UpdateEventConfigurations API](#) ou utilisez la update-event-configurations commande CLI. Par exemple :

```
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 actuelle des événements en appelant [l'DescribeEventConfigurations API](#) ou en utilisant la commande describe-event-configurations CLI. Par exemple : .

```
aws iot describe-event-configurations
```

Tableau des paramètres de configuration des AWS IoT événements

Catégorie d'événement (AWS IoTConsole : Paramètres : messages basés sur des événements)	eventConfigurations valeur clé (AWS CLI/API)	Objet du message d'événement
(Ne peut être configuré qu'à l'aide de l'AWS CLI/API)	CA_CERTIFICATE	\$aws/events/certificates/registered/ <i>caCertificateId</i>
(Ne peut être configuré qu'à l'aide de l'AWS CLI/API)	CERTIFICATE	\$aws/events/presence/connected/ <i>clientId</i>
(Ne peut être configuré qu'à l'aide de l'AWS CLI/API)	CERTIFICATE	\$aws/events/presence/disconnected/ <i>clientId</i>
(Ne peut être configuré qu'à l'aide de l'AWS CLI/API)	CERTIFICATE	\$aws/events/subscriptions/subscribed/ <i>clientId</i>
(Ne peut être configuré qu'à l'aide de l'AWS CLI/API)	CERTIFICATE	\$aws/events/subscriptions/unsubscribed/ <i>clientId</i>
Tâche terminée, annulée	JOB	\$aws/events/job/ <i>jobID</i> /canceled
Tâche terminée, annulée	JOB	\$aws/events/job/ <i>jobID</i> /cancellation_in_progress
Tâche terminée, annulée	JOB	\$aws/events/job/ <i>jobID</i> /completed

Catégorie d'événement (AWS IoTConsole : Paramètres : messages basés sur des événements)	eventConfigurations valeur clé (AWS CLI/API)	Objet du message d'événement
Tâche terminée, annulée	JOB	\$aws/events/job/ <i>jobID</i> /deleted
Tâche terminée, annulée	JOB	\$aws/events/job/ <i>jobID</i> /deletion_in_progress
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /canceled
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /deleted
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /failed
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /rejected
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /removed
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /succeeded
Exécution de la Job : réussite, échec, rejet, annulation, suppression	JOB_EXECUTION	\$aws/events/jobExecution/ <i>jobID</i> /timed_out
Objet : créé, mis à jour, supprimé	THING	\$aws/events/thing/ <i>thingName</i> /created
Objet : créé, mis à jour, supprimé	THING	\$aws/events/thing/ <i>thingName</i> /updated
Objet : créé, mis à jour, supprimé	THING	\$aws/events/thing/ <i>thingName</i> /deleted
Groupe d'objets : ajouté, supprimé	THING_GROUP	\$aws/events/thingGroup/ <i>thingGroupName</i> /created
Groupe d'objets : ajouté, supprimé	THING_GROUP	\$aws/events/thingGroup/ <i>thingGroupName</i> /updated

Catégorie d'événement (AWS IoTConsole : Paramètres : messages basés sur des événements)	eventConfigurations valeur clé (AWS CLI/API)	Objet du message d'événement
Groupe d'objets : ajouté, supprimé	THING_GROUP	\$aws/events/ thingGroup/ <i>thingGroupName</i> /deleted
Hiérarchie des groupes d'objets : ajoutée, supprimée	THING_GROUP_HIERARCHY	\$aws/events/ thingGroupHierarchy/ thingGroup/ <i>parentThingGroupName</i> / childThingGroup/ <i>childThingGroupName</i> / added
Hiérarchie des groupes d'objets : ajoutée, supprimée	THING_GROUP_HIERARCHY	\$aws/events/ thingGroupHierarchy/ thingGroup/ <i>parentThingGroupName</i> / childThingGroup/ <i>childThingGroupName</i> / removed
Appartenance à un groupe d'objets : ajoutée, supprimée	THING_GROUP_MEMBERSHIP	\$aws/events/ thingGroupMembership/ thingGroup/ <i>thingGroupName</i> / thing/ <i>thingName</i> /added
Appartenance à un groupe d'objets : ajoutée, supprimée	THING_GROUP_MEMBERSHIP	\$aws/events/ thingGroupMembership/ thingGroup/ <i>thingGroupName</i> / thing/ <i>thingName</i> /removed
Type d'objet : créé, mis à jour, supprimé	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / created
Type d'objet : créé, mis à jour, supprimé	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / updated
Type d'objet : créé, mis à jour, supprimé	THING_TYPE	\$aws/events/ thingType/ <i>thingTypeName</i> / deleted

Catégorie d'événement (AWS IoTConsole : Paramètres : messages basés sur des événements)	eventConfigurations valeur clé (AWS CLI/API)	Objet du message d'événement
Association de types d'objets : ajoutée, supprimée	THING_TYPE_ASSOCIATION	\$aws/events/thingTypeAssociation/thing/ <i>thingName</i> /thingType/ <i>thingTypeName</i> /added \$aws/events/thingTypeAssociation/thing/ <i>thingName</i> /thingType/ <i>thingTypeName</i> /removed

Événements de registre

Le registre peut publier des messages d'événement lorsque des objets, des types d'objets et des groupes d'objets sont créés, mis à jour ou supprimés. Ces événements ne sont toutefois pas disponibles par défaut. Pour plus d'informations sur la façon d'activer ces événements, consultez [Activer les événements pour AWS IoT \(p. 1253\)](#).

Le registre peut fournir les types d'événements suivants :

- [Évènements d'objets \(p. 1257\)](#)
- [Événements de type objet \(p. 1258\)](#)
- [Événements liés à des groupes d'objets \(p. 1260\)](#)

Évènements d'objets

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
- SUPPRIMÉE

accountId

Votre Compte AWS carte d'identité.

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 est renvoyé, null.

attributs

Un ensemble de paires nom-valeur associées à l'objet.

Événements de type objet

Événements liés au type d'objet :

- [Type d'objet créé/obsoète/dont l'obsolescence est annulée/supprimé \(p. 1258\)](#)
- [Type d'objet associé à un objet/dissocié d'un objet \(p. 1260\)](#)

Type d'objet créé/obsoète/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
- SUPPRIMÉE

accountId

Votre Compte AWS carte d'identité.

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 est renvoyé, false.

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*/thingType/*typeName*/added
- \$aws/events/thingTypeAssociation/thing/*thingName*/thingType/*typeName*/removed

Voici un exemple de added charge utile. Les charges utiles pour les removed messages sont similaires.

```
{  
    "eventId" : "87f8e095-531c-47b3-aab5-5171364d138d",  
    "eventType" : "THING_TYPE_ASSOCIATION_EVENT",  
    "operation" : "ADDED",  
    "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 :

- AJOUTÉE
- ENLEVÉ

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.

Événements liés à des groupes d'objets

Événements liés aux groupes d'objets :

- [Type d'objet créé/mis à jour/supprimé \(p. 1261\)](#)
- [Objet ajouté à un groupe d'objets/retiré d'un groupe d'objets \(p. 1262\)](#)
- [Groupe d'objets ajouté à un groupe d'objets/retiré d'un groupe d'objets \(p. 1263\)](#)

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. Les charges utiles created et les deleted messages sont similaires.

```
{
  "eventType": "THING_GROUP_EVENT",
  "eventId": "8b9ea8626aea1e42100f3f32b975899",
  "timestamp": 1603995417409,
  "operation": "UPDATED",
  "accountId": "571EXAMPLE833",
  "thingGroupId": "8757eec8-bb37-4cca-a6fa-403b003d139f",
  "thingGroupName": "Tg_level15",
  "versionNumber": 3,
  "parentGroupName": "Tg_level4",
  "parentGroupId": "5fce366a-7875-4c0e-870b-79d8d1dce119",
  "description": "New description for Tg_level15",
  "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
- SUPPRIMÉE

accountId

Votre Compte AWS carte d'identité.

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. Il existe un élément pour chaque groupe d'objets parent, en commençant par le groupe d'objets racine et en continuant jusqu'au groupe d'objets parent.

Chaque entrée contient le `groupArn` et du groupe d'objets `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

Votre Compte AWS carte d'identité.

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-489fcda41",  
    "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

Votre Compte AWS carte d'identité.

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 sur des sujets réservés sur le protocole MQTT lorsque des tâches sont en attente, terminées ou annulées, et lorsqu'un appareil signale un succès ou un échec lors de l'exécution d'une tâche. Les appareils ou les applications de gestion et de surveillance peuvent suivre l'état des tâches en s'abonnant à ces rubriques.

Comment activer les événements d'emploi

Les messages de réponse du service AWS IoT Jobs ne passent pas par le gestionnaire de messages et d'autres clients ou règles ne peuvent pas s'y abonner. Pour vous abonner aux messages relatifs à l'activité professionnelle, utilisez les rubriques `notify` et `notify-next`. Pour plus d'informations sur les sujets relatifs aux emplois, consultez [Rubriques de tâche \(p. 124\)](#).

Pour être informé des mises à jour des tâches, activez ces événements de tâches à l'aide de la AWS Management Console, ou à l'aide de l'API ou de l'interface de ligne de commande. Pour plus d'informations, veuillez consulter [Activer les événements pour AWS IoT \(p. 1253\)](#).

Comment fonctionnent les événements liés à l'emploi

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 des appareils" \(p. 89\)](#).

Types d'événements d'emploi

Voici les différents types d'événements d'emploi :

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 `canceled` message 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 `deleted` message 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"  
}
```

État du terminal d'exécution des Job

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 peut publier des événements du cycle de vie sur les sujets du MQTT. Ces événements sont disponibles par défaut et ne peuvent pas être désactivés.

Note

Les messages de cycle de vie peuvent être envoyés dans le désordre. Vous pouvez recevoir des messages en double.

Événements de connexion/déconnexion

Note

Grâce à AWS IoT l'indexation du parc Device Management, vous pouvez rechercher des objets, exécuter des requêtes agrégées et créer des groupes dynamiques en fonction des événements de connexion/déconnexion d'objets. Pour de plus amples informations, veuillez consulter [Indexation du parc](#).

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 gestionnaire 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. Pour ce faire, le client peut envoyer un paquet de DISCONNECT contrôle MQTT ou Close frame s'il utilise une WebSocket connexion.

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 WebSocket a été connecté depuis plus longtemps que sa time-to-live valeur.

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, il s'agit des 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. Pour les sessions persistantes, le numéro de version est réinitialisé à 0 lorsqu'un client a été déconnecté plus longtemps que la durée configurée time-to-live (TTL) pour la session persistante.

Un message de connexion présente la structure suivante.

```
{  
    "clientId": "186b5",  
    "timestamp": 1573002230757,  
    "eventType": "connected",  
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",  
    "principalIdentifier": "12345678901234567890123456789012",  
    "ipAddress": "192.0.2.0",  
    "versionNumber": 0  
}
```

Un message de déconnexion présente la structure suivante.

```
{  
    "clientId": "186b5",  
    "timestamp": 1573002340451,  
    "eventType": "disconnected",  
    "sessionIdentifier": "a4666d2a7d844ae4ac5d7b38c9cb7967",  
    "principalIdentifier": "12345678901234567890123456789012",  
    "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 qu'il est traité (par Lambda ou un autre service), vous pouvez d'abord 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
```

or

```
$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/ABCDEFGHIJKLMNPQRSTUVWXYZ:some-user/  
ABCDEFGHIJKLMNPQRSTUVWXYZ: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, il s'agit des 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 +/- 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 Corepour LoRa WAN

AWS IoT Corefor LoRa WAN est un serveur réseau LoRa WAN (LNS) entièrement géré qui assure la gestion des passerelles à l'aide des fonctionnalités du serveur de configuration et de mise à jour (CUPS) et des mises à jour du microprogramme en direct (FUOTA). Vous pouvez remplacer votre LNS privé par unAWS IoT Core réseau LoRa étendu et y connecter vos périphériques et passerelles de réseau étendu (LoRaWAN) àAWS IoT Core. Vous réduirez ainsi la maintenance, les coûts opérationnels, le temps de configuration et les frais généraux.

Note

AWS IoT Corefor LoRa WAN ne prend en charge que le format d'adresse IPv4. Il ne prend pas en charge IPv6 (IPv6 IPv6 Configuration IPv6). Pour plus amples informations, consultez [Service AWSs qui prennent en charge IPv6](#).

Introduction

Les périphériques WAN sont des appareils à longue portée, à faible consommation d'énergie et alimentés par batterie qui utilisent le protocole LoRa WAN pour fonctionner dans un spectre radio sans licence. LoRaLe WAN est un protocole de communication LPWAN (Low Power Wide Area Network) basé sur LoRa. LoRa est le protocole de couche physique qui permet une communication à faible consommation d'énergie sur une zone étendue entre les appareils.

Vous pouvez intégrer vos appareils LoRa WAN de la même manière que vous le feriez pour d'autres appareils IoT. Pour connecter vos appareils LoRa WAN àAWS IoT, vous devez utiliser une passerelle LoRa WAN. La passerelle agit comme un pont permettant de connecter votre appareil auAWS IoT Core LoRa réseau WAN et d'échanger des messages. AWS IoT Corefor LoRa WAN utilise le moteur deAWS IoT règles pour acheminer les messages de vos périphériques LoRa WAN vers d'autresAWS IoT services.

Pour réduire les efforts de développement et intégrer rapidement vos appareils auAWS IoT Core LoRa WAN, nous vous recommandons d'utiliser des terminaux LoRa certifiés WAN. Pour plus amples informations, consultez la page de [présentation du produitAWS IoT Core pour le réseau de LoRa réseau de réseau de réseau de réseau de réseau](#). Pour plus d'informations sur la certification LoRa WAN de vos appareils, consultez la section [Certification des produits LoRa WAN](#).

Comment utiliserAWS IoT Core pour le LoRa WAN

Vous pouvez rapidement intégrer vos périphériques et passerelles LoRa WAN à votre LoRa réseau WANAWS IoT Core à l'aide de la console ou de l'APIAWS IoT sans fil.

Utilisation de la console

Pour intégrer vos périphériques et passerelles LoRa WAN à l'aide duAWS Management Console, connectez-vous à la page dédiée au WAN sur laAWS IoT consoleAWS Management Console et accédez [AWS IoT Core à la page dédiée au LoRa réseau WAN](#). Vous pouvez ensuite utiliser la section Intro pour ajouter vos passerelles et appareils àAWS IoT Core for LoRa WAN. Pour plus d'informations, veuillez

consulter [Utilisation de la console pour intégrer votre appareil et votre passerelle vers AWS IoT Core for LoRa WAN \(p. 1277\)](#).

Utilisation de l'API ou de l'interface de ligne de commande

Vous pouvez intégrer à la fois des appareils LoRa WAN et Sidewalk à l'aide de l'API [AWS IoT sans fil](#). L'API AWS IoT sans fil sur laquelle AWS IoT Core repose le LoRa WAN est prise en charge par le AWS SDK. Pour plus amples informations, consultez [Kits SDK](#).

Vous pouvez utiliser le AWS CLI pour exécuter des commandes d'intégration et de gestion de vos passerelles et appareils LoRa WAN. Pour plus amples informations, consultez [Référence de ligne de ligne de AWS IoT](#).

AWS IoT Core pour les régions LoRa WAN et les points de terminaison

AWS IoT Core for LoRa WAN prend en charge les points de terminaison des API du plan de contrôle et du plan de données qui sont spécifiques à votre Région AWS. Les points de terminaison de l'API Data Plane sont spécifiques à votre Compte AWS et Région AWS. Pour plus d'informations sur les AWS IoT Core points de terminaison LoRa WAN, consultez la section [AWS IoT Core consacrée aux points de terminaison LoRa WAN](#) dans la Référence AWS générale.

Pour une communication plus sécurisée entre vos appareils AWS IoT, vous pouvez connecter vos appareils AWS IoT Core à la LoRa technologie WAN via AWS PrivateLink votre cloud privé virtuel (VPC) au lieu de vous connecter via Internet public. Pour plus d'informations, veuillez consulter [Connexion AWS IoT Core for LoRa WAN via un point de terminaison d'interface VPC \(p. 1308\)](#).

AWS IoT Core for LoRa WAN dispose de quotas qui s'appliquent aux données des appareils transmises entre les appareils et au TPS maximal pour les opérations de l'API AWS IoT sans fil. Pour plus d'informations, reportez-vous [AWS IoT Core à la section LoRa consacrée aux quotas WAN](#) dans la Référence AWS générale.

AWS IoT Core pour la tarification du LoRa WAN

Lorsque vous vous inscrivez à AWS, vous pouvez démarrer gratuitement avec la AWS IoT Core technologie pour le LoRa réseau étendu gratuitement en bénéficiant de l'[AWSS offre gratuite](#).

Pour plus amples informations sur la présentation générale des produits, consultez [AWS IoT Core Tarification](#).

Qu'est-ce que AWS IoT Core le LoRa WAN ?

AWS IoT Core for LoRa WAN remplace un serveur réseau LoRa WAN (LNS) privé en connectant vos périphériques et passerelles LoRa WAN à AWS. À l'aide du moteur de AWS IoT règles, vous pouvez router les messages reçus depuis des périphériques LoRa WAN, où ils peuvent être formatés et envoyés à d'autres AWS IoT services. Pour sécuriser les communications des appareils avec AWS IoT, AWS IoT Core for LoRa WAN utilise des certificats X.509.

AWS IoT Core for LoRa WAN gère les politiques relatives aux services et aux appareils AWS IoT Core nécessaires à la communication avec les passerelles et les appareils LoRa WAN. AWS IoT Core for LoRa

WAN gère également les destinations qui décrivent les AWS IoT règles qui envoient les données des appareils à d'autres services.

Avec AWS IoT Core for LoRa WAN, vous pouvez :

- Intégrez et connectez des périphériques et des passerelles LoRa WAN AWS IoT sans avoir à configurer et à gérer un LNS privé.
- Connectez des appareils LoRa WAN conformes aux spécifications LoRa WAN 1.0.x ou 1.1 normalisées par LoRa Alliance. Ces appareils peuvent fonctionner en mode classe A, classe B ou classe C.
- Utilisez des passerelles LoRa WAN compatibles avec LoRa Basics Station version 2.0.4 ou ultérieure. Toutes les passerelles qualifiées AWS IoT Core pour le LoRa WAN exécutent une version compatible de LoRa Basics Station.
- Surveillez la force du signal, la bande passante et le facteur de propagation en utilisant AWS IoT Core le débit de données adaptatif du LoRa WAN, et optimisez le débit de données si nécessaire.
- Mettez à jour le micrologiciel des passerelles LoRa WAN à l'aide du service CUPS et le micrologiciel des périphériques LoRa WAN à l'aide des mises à jour du microprogramme par voie aérienne (FUOTA).

Rubriques

- [Qu'est-ce que le LoRa WAN ? \(p. 1274\)](#)
- [Comment AWS IoT Core fonctionne le LoRa WAN \(p. 1275\)](#)

Qu'est-ce que le LoRa WAN ?

L'[LoRa Alliance](#) décrit le LoRa WAN comme « un protocole réseau LPWA (Low Power, Wide Area) conçu pour connecter sans fil des « objets » alimentés par batterie à Internet sur des réseaux régionaux, nationaux ou mondiaux, et répond aux exigences clés de l'Internet des objets (IoT) telles que les services de communication bidirectionnelle, de end-to-end sécurité, de mobilité et de localisation ». .

LoRa et LoRa WAN

Le protocole LoRa WAN est un protocole de communication LPWAN (Low Power Wide Area Networking) qui fonctionne sur LoRa. La spécification LoRa WAN est ouverte afin que tout le monde puisse configurer et exploiter un LoRa réseau.

LoRa est une technologie de fréquence audio sans fil qui fonctionne dans un spectre de radiofréquences sans licence. LoRa est un protocole de couche physique qui utilise la modulation à spectre étalé et prend en charge les communications à 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 le rend résistant aux interférences.

Caractéristiques de la technologie LoRa WAN

- Communication à longue portée jusqu'à 10 miles en ligne de visée.
- Longue durée de vie 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 de liaison descendante accrue.
- Faible coût pour les appareils et la maintenance.
- Spectre radioélectrique sans licence, mais des réglementations spécifiques à la région s'appliquent.
- Faible consommation d'énergie, mais sa taille de charge utile est limitée de 51 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.

En savoir plus sur le LoRa réseau WAN

Les liens suivants contiennent des informations utiles sur la technologie LoRa WAN et sur LoRa Basics Station, le logiciel qui s'exécute sur vos passerelles LoRa WAN pour connecter des appareils finaux àAWS IoT Core un LoRa réseau WAN.

- [Les fondamentaux du LoRa WAN](#)

The Things Fundamentals on LoRa WAN contient une vidéo d'introduction qui couvre les principes fondamentaux du LoRa WAN et une série de chapitres qui vous aideront à en savoir plus sur LoRa le LoRa WAN.

- [Qu'est-ce que le LoRa WAN](#)

LoRa Alliance fournit une présentation technique du LoRa LoRa WAN, y compris un résumé des spécifications du LoRa WAN dans différentes régions.

- [LoRa Station Basics](#)

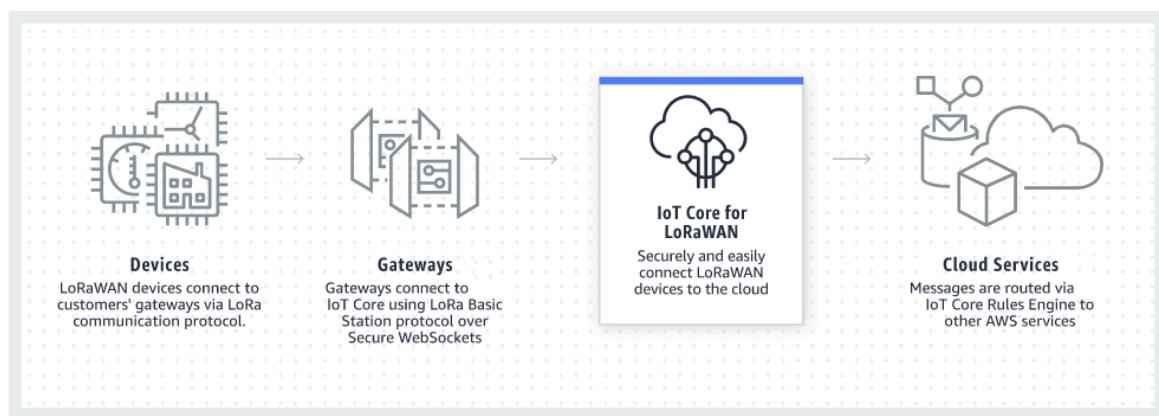
Semtech Corporation fournit des concepts utiles sur LoRa les bases des passerelles et des nœuds de terminaison. LoRa Basics Station, un logiciel open source qui s'exécute sur votre passerelle LoRa WAN, est maintenu et distribué via le [GitHub](#) référentiel de Semtech Corporation. Vous pouvez également en savoir plus sur les protocoles LNS et CUPS qui décrivent comment échanger des données LoRa WAN et effectuer des mises à jour de configuration.

CommentAWS IoT Core fonctionne le LoRa WAN

L'architecture réseau LoRa WAN est déployée selon une topologie en étoile dans laquelle les passerelles relaient les informations entre les terminaux et le serveur réseau LoRa WAN (LNS).

AWS IoT Corefor LoRa WAN vous permet de connecter et de gérer des périphériques LoRa WAN (réseau étendu longue portée à faible consommation) et vous évite de devoir développer et exploiter un LNS. Les périphériques et passerellesLoRa WAN (WAN) à longue portée peuvent se connecter enAWS IoT Core utilisantAWS IoT Core for LoRa WAN.

Ce qui suit montre comment un périphérique LoRa WAN interagit avecAWS IoT Core for LoRa WAN. Il montre également commentAWS IoT Core le LoRa WAN remplace un LNS et communique avecService AWS les autres systèmes duAWS Cloud.



LoRaWAN appareils WAN communiquent avec euxAWS IoT Core via des passerelles LoRa WAN. AWS IoT Corefor LoRa WAN gère les politiques relatives aux services et aux appareils nécessairesAWS IoT Core à la gestion et à la communication avec les passerelles et les appareils LoRa WAN. AWS IoT Corefor LoRa WAN gère également les destinations qui décrivent lesAWS IoT règles qui envoient les données des appareils à d'autres services.

Commencez à utiliser AWS IoT Core for LoRa WAN

1. Sélectionnez les appareils sans fil et les passerelles LoRa WAN dont vous aurez besoin.
Le [AWS Partner Device Catalog](#) contient des passerelles et des kits de développement pouvant être utilisés avec AWS IoT Core for LoRa WAN. Pour plus d'informations, veuillez consulter [Utilisation de passerelles qualifiées figurant dans le catalogue d'appareils AWS partenaires \(p. 1319\)](#).
2. Ajoutez vos appareils sans fil et vos passerelles LoRa WAN à votre LoRa réseau WAN.
[Connexion de passerelles et d'appareils à AWS IoT Core for LoRaWAN \(p. 1276\)](#) vous explique comment décrire vos ressources et ajouter vos périphériques sans fil et vos passerelles LoRa LoRa WAN à AWS IoT Core au WAN. Vous apprendrez également à configurer l'autre AWS IoT Core pour les ressources LoRa WAN dont vous aurez besoin pour gérer ces appareils et envoyer leurs données aux AWS services.
3. Complétez votre solution AWS IoT Core pour LoRa WAN.

Commencez par [notre AWS IoT Core exemple de solution LoRa WAN](#) et personnalisez-le.

AWS IoT Core pour les ressources LoRa WAN

Les ressources suivantes vous aideront à vous familiariser avec la technologie LoRa WAN et AWS IoT Core pour le LoRa WAN.

- [Débuter avec AWS IoT Core for LoRa WAN](#)

La vidéo suivante décrit le fonctionnement du LoRa WAN et vous explique comment AWS IoT Core ajouter des passerelles LoRa WAN à partir du AWS Management Console.

- [AWS IoT Core pour LoRa un atelier WAN](#)

L'atelier couvre les principes fondamentaux de la technologie LoRa WAN et de sa mise en œuvre avec AWS IoT Core for LoRa WAN. Vous pouvez également utiliser l'atelier pour parcourir des ateliers qui montrent comment connecter votre passerelle et votre appareil à AWS IoT Core un LoRa réseau WAN afin de créer un exemple de solution IoT.

Connexion de passerelles et d'appareils à AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN vous permet de connecter et de gérer des périphériques LoRa WAN sans fil (réseau étendu longue portée à faible consommation) et vous évite de devoir développer et exploiter un LNS. Les périphériques et passerelles LoRa WAN (WAN) longue portée peuvent se connecter à AWS IoT Core l'aide de AWS IoT Core for LoRaWAN.

Conventions de dénomination pour vos appareils, passerelles, profils et destinations

Avant de commencer à utiliser AWS IoT Core for LoRaWAN et à créer les ressources, tenez compte de la convention de dénomination de vos appareils, de vos passerelles et de votre destination.

AWS IoT Core for LoRaWAN attribue des identifiants uniques aux ressources que vous créez pour les appareils sans fil, les passerelles et les profils ; toutefois, vous pouvez également attribuer à vos ressources des noms plus descriptifs afin de les identifier plus facilement. Avant d'ajouter des appareils,

des passerelles, des profils et des destinationsAWS IoT Core for LoRaWAN, réfléchissez à la façon dont vous allez les nommer afin de les gérer plus facilement.

Vous pouvez ajouter des balises aux ressources que vous créez. Avant d'ajouter vos périphériques LoRa WAN, réfléchissez à la manière dont vous pourriez utiliser les balises pour identifier et gérer vosAWS IoT Core for LoRaWAN ressources. Les balises peuvent être modifiées une fois que vous les avez ajoutées.

Pour de plus amples informations sur l'attribution de noms et le balisage, veuillez consulter[Décrivez vos ressourcesAWS IoT Core pour le LoRa WAN \(p. 1278\)](#).

Mappage des données de l'appareil vers les données de service

Les données des appareils sans fil LoRa WAN sont souvent codées pour optimiser la bande passante. Ces messages codés arriventAWS IoT Core for LoRaWAN dans un format qui risque de ne pas être facilement utilisé par d'autresAWS services. AWS IoT Core for LoRaWANutilise desAWS IoT règles qui peuvent utiliserAWS Lambda des fonctions pour traiter et décoder les messages de l'appareil dans un format utilisable par d'autresAWS services.

Pour transformer les données de l'appareil et les envoyer à d'autresAWS services, vous devez connaître les informations suivantes :

- Format et contenu des données envoyées par les appareils sans fil.
- Service auquel vous souhaitez envoyer les données.
- Le format requis par ce service.

À l'aide de ces informations, vous pouvez créer laAWS IoT règle qui effectue la conversion et envoie les données converties auxAWS services qui les utiliseront.

Utilisation de la console pour intégrer votre appareil et votre passerelle versAWS IoT Core for LoRaWAN

Vous pouvez utiliser l'interface de console ou l'API pour ajouter votre passerelle LoRa WAN et vos appareils. Si vous utilisez la consoleAWS IoT Core for LoRaWAN pour la première fois, nous vous recommandons d'utiliser la console. L'interface de console est particulièrement pratique lorsque vous gérez quelquesAWS IoT Core for LoRaWAN ressources à la fois. Lorsque vous gérez un grand nombre deAWS IoT Core for LoRaWAN ressources, envisagez de créer des solutions plus automatisées à l'aide de l'AWS IoT WirelessAPI.

La plupart des données que vous saisissez lors de la configurationAWS IoT Core for LoRaWAN des ressources sont fournies par les fournisseurs des appareils et sont spécifiques aux spécifications LoRa WAN qu'ils prennent en charge. Les rubriques suivantes décrivent comment décrire vosAWS IoT Core ressources LoRa WAN et utiliser la console ou l'API pour ajouter vos passerelles et appareils.

Rubriques

- [Décrivez vos ressourcesAWS IoT Core pour le LoRa WAN \(p. 1278\)](#)
- [Intégrez vos passerelles versAWS IoT Core for LoRaWAN \(p. 1279\)](#)
- [Intégrez vos appareils àAWS IoT Core for LoRaWAN \(p. 1288\)](#)

Décrivez vos ressourcesAWS IoT Core pour le LoRa WAN

Si vous l'utilisezAWS IoT Core for LoRaWAN pour la première fois, vous pouvez ajouter votre première passerelle LoRa WAN et votre premier appareil en utilisant la page d'[AWS IoT Core for LoRaWAN](#)introduction de laAWS IoT console.

Avant de commencer à créer les ressources, prenez en compte la convention de dénomination de vos appareils, de vos passerelles et de votre destination. AWS IoT Core for LoRaWANpropose plusieurs options pour identifier les ressources que vous créez. Bien que lesAWS IoT Core for LoRaWAN ressources reçoivent un identifiant unique lors de leur création, cet identifiant n'est pas descriptif et ne peut pas être modifié après la création de la ressource. Vous pouvez également attribuer un nom, ajouter une description et associer des balises et des valeurs de balises à la plupartAWS IoT Core for LoRaWAN des ressources afin de faciliter la sélection, l'identification et la gestion de vosAWS IoT Core for LoRaWAN ressources.

- [Noms des ressources \(p. 1278\)](#)

Pour les passerelles, les appareils et les profils, le nom de la ressource est un champ facultatif que vous pouvez modifier une fois la ressource créée. Le nom apparaît dans les listes affichées sur les pages du centre de ressources.

Pour les destinations, vous devez fournir un nom unique dans votreAWS compte etRégion AWS. Vous ne pouvez pas modifier le nom de la destination après avoir créé la ressource de destination.

Bien qu'un nom puisse comporter 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.

- [Étiquettes de ressources \(p. 1279\)](#)

Les balises sont des paires clé-valeur de métadonnées qui peuvent être attachées auxAWS ressources. Vous choisissez à la fois les clés de balise et les valeurs correspondantes.

Les passerelles, les destinations et les profils peuvent être associés jusqu'à 50 balises. Les appareils ne prennent pas en charge les balises.

Noms des ressources

AWS IoT Core for LoRaWANsupport de ressources pour le nom

Ressource	Nom Field Support
Destination	Le nom est l'ID unique de la ressource et ne peut pas être modifié.
Device	Le nom est un descripteur facultatif de la ressource et peut être modifié.
Passerelle	Le nom est un descripteur facultatif de la ressource et peut être modifié.
Profil	Le nom est un descripteur facultatif de la ressource et peut être modifié.

Le champ de nom apparaît dans les listes de ressources du hub de ressources ; toutefois, l'espace étant limité, seuls les 15 à 30 premiers caractères du nom peuvent être visibles.

Lorsque vous sélectionnez les noms de vos ressources, réfléchissez à la manière dont vous souhaitez qu'elles identifient les ressources et à la manière dont elles seront affichées dans la console.

Description

Les ressources relatives à la destination, à l'appareil et à la passerelle prennent également en charge un champ de description, qui peut accepter jusqu'à 2 048 caractères. Le champ de description apparaît uniquement sur la page détaillée de chaque ressource. Bien que le champ de description puisse contenir de nombreuses informations, étant donné qu'il n'apparaît que sur la page détaillée de la ressource, il n'est pas pratique pour numériser plusieurs ressources.

Étiquettes de ressources

AWS IoT Core for LoRaWAN support de ressources pour les AWS balises

Ressource	AWS support de balises
Destination	Vous pouvez ajouter jusqu'à 50 AWS balises à la ressource.
Device	Cette ressource ne prend pas en charge les AWS balises.
Passerelle	Vous pouvez ajouter jusqu'à 50 AWS balises à la ressource.
Profil	Vous pouvez ajouter jusqu'à 50 AWS balises à la ressource.

Les balises sont des mots ou des expressions qui font office de métadonnées qui vous permettent d'identifier et d'organiser vos AWS ressources. Vous pouvez considérer la clé de balise comme une catégorie d'informations et la valeur de la balise comme une valeur spécifique de cette catégorie.

Par exemple, vous pouvez avoir une valeur de balise de couleur, puis attribuer à certaines ressources une valeur bleue pour cette balise et à d'autres une valeur de rouge. Vous pouvez ainsi utiliser l'[éditeur de balises](#) de la AWS console pour rechercher les ressources dont la valeur de balise de couleur est bleue.

Pour de plus amples informations sur le balisage et les stratégies de balisage, veuillez consulter [Balisage Editor](#).

Intégrer vos passerelles vers AWS IoT Core for LoRaWAN

Si vous utilisez AWS IoT Core for LoRaWAN pour la première fois, vous pouvez ajouter votre première passerelle LoRa WAN et votre premier appareil à l'aide de la console.

Avant d'intégrer votre passerelle

Avant d'intégrer votre passerelle vers AWS IoT Core for LoRaWAN, nous vous recommandons de :

- Utilisez des passerelles qualifiées pour être utilisées avec AWS IoT Core for LoRaWAN. Ces passerelles se connectent à AWS IoT Core sans aucun paramètre de configuration supplémentaire et sont équipées

d'une version compatible du logiciel [LoRa Basics Station](#). Pour plus d'informations, veuillez consulter [Gestion des passerelles avec AWS IoT Core for LoRa WAN \(p. 1319\)](#).

- Tenez compte de la convention de dénomination des ressources que vous créez afin de pouvoir les gérer plus facilement. Pour plus d'informations, veuillez consulter [Décrivez vos ressources AWS IoT Core pour le LoRa WAN \(p. 1278\)](#).
- Préparez à l'avance les paramètres de configuration propres à chaque passerelle, afin de faciliter la saisie des données dans la console. Les paramètres de configuration de la passerelle sans fil qui AWS IoT nécessitent de communiquer avec la passerelle et de la gérer incluent l'EUI de la passerelle et sa bande de LoRa fréquences.

Pour intégrer vos passerelles vers AWS IoT Core for LoRaWAN :

- [Envisagez de sélectionner la bande de fréquences et ajoutez le rôle IAM nécessaire \(p. 1280\)](#)
- [Ajouter une passerelle à AWS IoT Core for LoRaWAN \(p. 1282\)](#)
- [Connectez votre passerelle LoRa WAN et vérifiez son état de connexion \(p. 1287\)](#)

Envisagez de sélectionner la bande de fréquences et ajoutez le rôle IAM nécessaire

Avant d'ajouter votre passerelle à AWS IoT Core for LoRaWAN, nous vous recommandons de prendre en compte la bande de fréquences dans laquelle votre passerelle fonctionnera et d'ajouter le rôle IAM nécessaire à la connexion de votre passerelle AWS IoT Core for LoRaWAN.

Note

Si vous ajoutez votre passerelle à l'aide de la console, cliquez sur **Créer un rôle** dans la console pour créer le rôle IAM nécessaire afin de pouvoir ignorer ces étapes. Vous devez effectuer ces étapes uniquement si vous utilisez l'interface de ligne de commande pour créer la passerelle.

Envisagez de sélectionner des bandes de LoRa fréquences pour vos passerelles et la connexion de vos appareils

AWS IoT Core for 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 appareils physiquement présents dans des 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 votre pays, consultez les [paramètres régionaux du LoRa WAN®](#).

LoRa Alliance publie des spécifications LoRa WAN et des documents sur les paramètres régionaux qui peuvent être téléchargés sur le site Web de LoRa l'Alliance. Les paramètres régionaux de LoRa l'Alliance aident les entreprises à choisir la bande de fréquences à utiliser dans leur région ou leur pays. AWS IoT Core for LoRaWAN de la mise en œuvre de la bande de fréquences suit les recommandations du document de spécification des paramètres régionaux. Ces paramètres régionaux sont regroupés dans un ensemble de paramètres radio, avec une attribution 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.

Ajoutez 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 explique comment ajouter un rôle IAM qui permettra au serveur de configuration et de mise à jour (CUPS) de gérer les informations d'identification de la passerelle. Assurez-vous d'exécuter

cette procédure avant qu'une passerelle LoRa WAN n'essaie de se connecter à AWS IoT Core for LoRaWAN. Toutefois, vous ne devez effectuer cette opération qu'une seule fois.

Ajoutez 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. Ouvrez le [hub Rôles de la console IAM](#) et choisissez Créer un rôle.
2. Si vous pensez avoir déjà ajouté le WirelessGatewayCertManagerRole rôle IoT, entrez dans la barre de recherche **IoTWirelessGatewayCertManagerRole**.

Si vous voyez un WirelessGatewayCertManagerRole rôle IoT dans les résultats de recherche, vous disposez du rôle IAM nécessaire. Vous pouvez quitter la procédure maintenant.

Si les résultats de recherche sont vides, vous ne disposez pas du rôle IAM nécessaire. Continuez la procédure pour l'ajouter.

3. Dans Sélectionner le type d'entité sécurisée, choisissez AutreCompte AWS.
4. Dans Identifiant du compte, saisissez votreCompte AWS identifiant, puis choisissez Suivant : Autorisations.
5. Dans la zone de recherche, saisissez **AWSIoTWirelessGatewayCertManager**.
6. Dans la liste des résultats de recherche, sélectionnez la politique nommée AWSIoTWirelessGatewayCertManager.
7. Sélectionnez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
8. Dans Nom du rôle, entrez **IoTWirelessGatewayCertManagerRole**, puis choisissez Créer un rôle.
9. Pour modifier le nouveau rôle, dans le message de confirmation, choisissez IoTWirelessGatewayCertManagerRole.
10. Dans Résumé, choisissez l'onglet Relations de confiance, puis choisissez Modifier la relation de confiance.
11. Dans le document de politique, modifiez la **Principal** propriété pour qu'elle ressemble à cet exemple.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Une fois la **Principal** propriété modifiée, le document de politique complet doit ressembler à 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, choisissez Mettre à jour la politique de confiance.

Vous avez maintenant créé l'IoTWirelessGatewayCertManagerRole. Tu n'auras pas besoin de le refaire.

Si vous avez effectué cette procédure alors que vous ajoutez une passerelle, vous pouvez fermer cette fenêtre et la console IAM, puis revenir à la AWS IoT console pour terminer l'ajout de la passerelle.

Ajouter une passerelle à AWS IoT Core for LoRaWAN

Vous pouvez ajouter votre passerelle à AWS IoT Core for LoRaWAN l'aide de la console ou d'CLI.

Avant d'ajouter votre passerelle, nous vous recommandons de prendre en compte les facteurs mentionnés dans la section Avant d'intégrer votre passerelle de [Intégrez vos passerelles vers AWS IoT Core for LoRaWAN \(p. 1279\)](#).

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 déjà avoir créé le rôle IAM nécessaire pour que la passerelle puisse se connecter à AWS IoT Core for LoRaWAN. Pour plus d'informations sur la création du rôle, consultez [Ajoutez un rôle IAM pour permettre au serveur de configuration et de mise à jour \(CUPS\) de gérer les informations d'identification de la passerelle \(p. 1280\)](#).

Ajouter une passerelle à l'aide de la console

Accédez à la page d'[AWS IoT Core for LoRaWAN](#) introduction de la AWS IoT console et choisissez Commencer, puis Ajouter une passerelle. Si vous avez déjà ajouté une passerelle, choisissez Afficher la passerelle pour afficher la passerelle que vous avez ajoutée. Si vous souhaitez ajouter d'autres passerelles, choisissez Ajouter une passerelle.

1. Fournissez des détails sur la passerelle et des informations sur les bandes de fréquences

Utilisez la section Détails de la passerelle pour fournir des informations sur les données de configuration du périphérique, telles que l'EUI de la passerelle et la configuration de la bande de fréquences.

- Interface utilisateur de Gateway

L'EUI (Extended Unique Identifier) du périphérique passerelle individuel. L'EUI est un code alphanumérique à 16 chiffres: `00ee40fffff29df10`, tel que celui qui identifie de manière unique une passerelle sur votre réseau LoRa WAN. 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'EUI suit une norme EUI-64 qui identifie de manière unique votre passerelle et ne peut donc pas être réutilisée dans d'autres Compte AWS pays et régions.

- Bande de fréquence (RFRegion)

La bande de fréquence de la passerelle. Vous pouvez choisir entre US915,, ou EU868AU915AS923-1, en fonction de ce que prend en charge votre passerelle et du pays ou de la région à partir duquel la passerelle se connecte physiquement. Pour de plus amples informations sur les bandes, veuillez consulter [Envisagez de sélectionner des bandes de LoRa fréquences pour vos passerelles et la connexion de vos appareils \(p. 1280\)](#).

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 attribuer un nom à la passerelle, utiliser le champ Description pour fournir des informations sur la passerelle et utiliser des balises pour ajouter des paires clé-valeur de métadonnées concernant la passerelle. Pour plus d'informations sur la dénomination et la description de vos ressources, consultez [Décrivez vos ressources AWS IoT Core pour le LoRa WAN \(p. 1278\)](#).

- LoRaConfiguration WAN à l'aide de sous-bandes et de filtres

En option, vous pouvez également spécifier des données de configuration LoRa WAN telles que les sous-bandes que vous souhaitez utiliser et des filtres capables de contrôler le flux de trafic. Pour ce didacticiel, vous pouvez ignorer ces champs. Pour plus d'informations, veuillez consulter [Configurer la position des ressources sans fil avec AWS IoT Core for LoRa WAN \(p. 1300\)](#).

- Informations sur la position et destination

En option, vous pouvez également spécifier les informations de position et une destination qui décrit la AWS IoT règle qui traite les données de position de l'appareil à des fins d'utilisation par AWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Configurer la position des ressources sans fil avec AWS IoT Core for LoRa WAN \(p. 1300\)](#).

Note

La fonction d'information sur la position est disponible en version préliminaire pour AWS IoT Core for LoRaWAN et susceptible d'être modifiée.

3. Associez n'AWS IoT importe quel objet à la passerelle

Spécifiez s'il faut créer un AWS IoT objet et l'associer à la passerelle. Les éléments AWS IoT inclus peuvent faciliter la recherche et la gestion de vos appareils. L'association d'un objet à votre passerelle permet à cette dernière d'accéder à d'autres AWS IoT fonctionnalités.

4. Crédit et téléchargement du certificat de passerelle

Pour authentifier votre passerelle afin qu'elle puisse communiquer en toute sécurité AWS IoT, votre passerelle LoRa WAN doit présenter une clé privée et un certificat à AWS IoT Core for LoRaWAN. Créez un certificat de passerelle AWS IoT afin de vérifier l'identité de votre passerelle à l'aide de la norme X.509.

Cliquez sur le bouton Créez un certificat et 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 LoRa WAN doit se connecter à un point de terminaison CUPS ou LNS lors de l'établissement d'une connexion à AWS IoT Core for LoRaWAN. Nous vous recommandons d'utiliser le point de terminaison CUPS car il permet également la gestion de la configuration. Pour vérifier l'authenticité des AWS IoT Core for LoRaWAN points de terminaison, votre passerelle utilisera un certificat de confiance pour chacun des points de terminaison CUPS et LNS.

Cliquez sur le bouton Copier pour copier les points de terminaison CUPS et LNS. Vous aurez besoin de cette information ultérieurement pour configurer votre passerelle. Cliquez ensuite sur le bouton Télécharger les certificats de confiance du serveur pour télécharger les certificats de confiance pour les points de terminaison CUPS et LNS.

6. Créez 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 la passerelle. Vous devez effectuer cette opération avant qu'une passerelle LoRa WAN n'essaie de se connecter à AWS IoT Core for LoRaWAN. Toutefois, vous ne devez le faire qu'une seule fois.

Pour créer le rôle IoTWirelessGatewayCertManager IAM pour votre compte, cliquez sur le bouton Créez un rôle. Si le rôle existe déjà, sélectionnez-le dans la liste déroulante.

Cliquez sur Soumettre pour terminer la création de la passerelle.

Ajouter une passerelle à l'aide de l'API

Note

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 le rôle `WirelessGatewayCertManager` IAM IoT afin que la passerelle puisse se connecter à AWS IoT Core for LoRaWAN. Pour plus d'informations sur la création du rôle, consultez la section suivante [Ajoutez un rôle IAM pour permettre au serveur de configuration et de mise à jour \(CUPS\) de gérer les informations d'identification de la passerelle \(p. 1280\)](#).

Les sections suivantes montrent comment ajouter une passerelle à l'aide des opérations AWS IoT Wireless d'API ou du AWS CLI. Vous ajoutez d'abord votre passerelle, puis vous associez un certificat, un certificat à la passerelle. Vous pouvez également utiliser les opérations d'API supplémentaires, par exemple pour mettre à jour une passerelle existante.

Rubriques

- [Comment ajouter votre passerelle \(p. 1284\)](#)
- [Associez un certificat à votre passerelle \(p. 1284\)](#)
- [Opérations d'API supplémentaires \(p. 1286\)](#)

Comment ajouter votre passerelle

Vous pouvez utiliser le AWS CLI pour créer une passerelle sans fil en utilisant l'opération `CreateWirelessGateway` API ou la commande `create-wireless-gateway` CLI pour ajouter votre passerelle sans fil.

Note

Si votre passerelle communique avec des périphériques LoRa WAN de classe B, vous pouvez également spécifier certains paramètres de balisage lors de l'ajout de la passerelle à l'aide de l'`CreateWirelessGateway` API ou de la commande `create-wireless-gateway` CLI. Pour plus d'informations, veuillez consulter [Configuration de vos passerelles pour envoyer des balises à des appareils de classe B \(p. 1321\)](#).

L'exemple suivant crée une passerelle de périphérique LoRa WAN Wireless. Vous pouvez également fournir un `input.json` fichier contenant des informations supplémentaires telles que le certificat de passerelle et les informations d'identification de configuration.

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 file://input.json
```

Associez un certificat à votre passerelle

Une fois que vous avez ajouté votre passerelle à AWS IoT Wireless, elle doit être associée à un certificat pour se connecter au point de terminaison CUPS. Pour se connecter au point de terminaison, votre passerelle exécutant LoRa Basics Station a besoin des fichiers suivants :

- `cups.crt` - Le certificat CUPS de la passerelle qu'elle utilise pour se connecter au point de terminaison CUPS.

- `cups.key`- Clé privée correspondant au certificat.
- `cups.trust`- Le certificat de confiance du point de terminaison CUPS.
- `cups.uri`- L'URI du point de terminaison CUPS.

Les étapes suivantes vous montrent comment générer un certificat et l'associer à votre passerelle.

Rubriques

- [Étape 1 : génération d'un certificat de passerelle \(p. 1285\)](#)
- [Étape 2 : obtention du certificat de confiance du serveur et du point de terminaison CUPS \(p. 1285\)](#)
- [Étape 3 : Associer le certificat à votre passerelle \(p. 1286\)](#)

Étape 1 : génération d'un certificat de passerelle

Pour générer un certificat pour votre passerelle, utilisez l'action AWS IoT API Reference ou la AWS CLI commande [create-keys-and-certificate](#) CLI. [CreateKeysAndCertificate](#)

La commande suivante montre un exemple de génération du certificat et de la clé privée `cups.key.cups.crt`

```
aws iot create-keys-and-certificate \
--set-as-active --certificate-pem-outfile "cups.crt" \
--private-key-outfile "cups.key"
```

L'exécution de cette commande génère le certificat et la clé privée, ainsi qu'un ID de certificat. L'exemple suivant montre le résultat de l'exécution de cette commande.

```
{
  "certificateArn": "arn:aws:iot:us-
east-1:123456789012:cert/abc1234d55ef32101a34434bb123cba2a011b2cdefa6bb5cee1a221b4567ab12",
  "certificateId": "abc1234d55ef32101a34434bb123cba2a011b2cdefa6bb5cee1a221b4567ab12",
  "certificatePem": "-----BEGIN CERTIFICATE-----\n..\n-----END CERTIFICATE-----\n",
  "KeyPair": {
    "PublicKey": "-----BEGIN PUBLIC KEY -----\\n..\\n-----END PUBLIC KEY----\\n",
    "PrivateKey": "----BEGIN RSA PRIVATE KEY----\\n..\\nEND RSA PRIVATE KEY----\\n"
  }
}
```

Enregistrez temporairement l'identifiant du certificat, car il sera utilisé à l'étape suivante pour associer votre certificat à la passerelle.

Note

Vous devez stocker la clé privée en toute sécurité `cups.key`. Si vous égarez la clé privée, réexécutez la `create-keys-and-certificate` commande pour générer un autre certificat.

Étape 2 : obtention du certificat de confiance du serveur et du point de terminaison CUPS

Maintenant que vous avez généré le certificat et la clé privée, utilisez l'action d'[GetServiceEndpoint](#) API ou la commande [get-service-endpoint](#) CLI pour obtenir le certificat de confiance du serveur `cups.trust` et l'URI du point de terminaison `cups.uri`.

La commande suivante montre un exemple d'obtention du certificat de confiance du serveur et de l'URI du point de terminaison. Lors de l'exécution de la commande, définissez le `service-type` paramètre sur CUPS.

```
aws iotwireless get-service-endpoint --service-type CUPS
```

Voici une sortie d'exécution de la commande.

```
{  
    "ServiceType": "CUPS",  
    "ServiceEndpoint": "https://ABCDEFGHIJKLMN.cups.lorawan.us-east-1.amazonaws.com:443",  
    "ServerTrust": "-----BEGIN CERTIFICATE-----\n..\n-----END CERTIFICATE-----\n"  
}
```

Le `ServiceEndpoint` résultat obtenu à partir de la réponse correspond au paramètre `CUPS,cups.uri`.

Note

Stockez le `ServerTrust` certificat dans un `.pem` fichier en le \n remplaçant par de nouvelles lignes.

Étape 3 : Associer le certificat à votre passerelle

Vous devez associer le certificat de passerelle que vous avez généré à la passerelle que vous avez ajoutée. AWS IoT Core for LoRaWAN utilisera ces informations pour identifier le certificat que la passerelle utilisera pour se connecter au point de terminaison CUPS.

Pour associer le certificat à votre passerelle, utilisez l'action [AssociateWirelessGatewaywithCertificate API](#) ou la commande [associate-wireless-gateway-with-certificate CLI](#).

La commande suivante montre un exemple d'association d'un certificat à votre passerelle.

```
aws iotwireless associate-wireless-gateway-with-certificate \  
  --id <WirelessGatewayId> \  
  --iot-certificate-id <CertificateId>
```

L'exécution de cette commande renvoie le `IotCertificateId`, qui est l'ID du certificat que vous avez associé à la passerelle. Ce qui suit montre le résultat de l'exécution de la commande, où `IotCertificateId` est l'ID du certificat, tel que **abc1234d55ef32101a34434bb123cba2a011b2cdefa6bb5cee1a221b4567ab12**.

```
{  
    "IotCertificateId": "<CertificateId>"  
}
```

Opérations d'API supplémentaires

Vous pouvez utiliser les actions d'API suivantes pour effectuer les tâches associées à l'ajout, à la mise à jour ou à la suppression d'une passerelle LoRa WAN.

AWS IoT WirelessActions d'API pour les AWS IoT Core for LoRaWAN passerelles

- [GetWirelessGateway](#)
- [ListWirelessGateways](#)
- [UpdateWirelessGateway](#)
- [DeleteWirelessGateway](#)

Pour obtenir la liste complète des actions et des types de données disponibles pour créer et gérer AWS IoT Core for LoRaWAN des ressources, consultez la [référence de l'AWS IoT Wireless API](#).

Pour plus d'informations sur les CLI que vous pouvez utiliser, consultez la [AWS CLI référence](#).

Connectez votre passerelle LoRa WAN et vérifiez son état de connexion

Avant de pouvoir vérifier l'état de la connexion à la passerelle, vous devez déjà avoir ajouté votre passerelle et y être connectée AWS IoT Core for LoRaWAN. Pour plus d'informations sur la procédure à suivre pour ajouter votre passerelle, consultez [Ajouter une passerelle à AWS IoT Core for LoRaWAN \(p. 1282\)](#).

Connectez votre passerelle à AWS IoT Core for LoRaWAN

Après avoir ajouté votre passerelle, connectez-vous à l'interface de configuration de votre passerelle pour saisir les informations de configuration et les certificats de confiance.

Après avoir ajouté les informations de la passerelle à AWS IoT Core for LoRaWAN, ajoutez AWS IoT Core for LoRaWAN des informations au périphérique de passerelle. La documentation fournie par le fournisseur de la passerelle doit décrire le processus de téléchargement des fichiers de certificat vers la passerelle et de configuration du périphérique de passerelle avec lequel communiquer AWS IoT Core for LoRaWAN.

Passerelles qualifiées pour être utilisées avec AWS IoT Core for LoRaWAN

Pour obtenir des instructions sur la configuration de votre passerelle LoRa WAN, reportez-vous à la section [Configuration du périphérique de passerelle](#) de l'AWS IoT Core for LoRaWAN Atelier. Vous trouverez ici des informations sur les instructions de connexion de passerelles qualifiées pour être utilisées avec AWS IoT Core for LoRaWAN.

Passerelles compatibles avec le protocole CUPS

Les instructions suivantes montrent 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.
 - Certificats du périphérique passerelle et fichiers de clé privée.
 - Fichier de certificat de confiance 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 procédure à suivre pour obtenir cette information, consultez [Ajouter une passerelle à AWS IoT Core for LoRaWAN \(p. 1282\)](#).

Passerelles compatibles avec le protocole LNS

Les instructions suivantes montrent 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.
 - Certificats du périphérique passerelle et fichiers de clé privée.
 - Fichier de certificat de confiance 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 procédure à suivre pour obtenir cette information, consultez [Ajouter une passerelle à AWS IoT Core for LoRaWAN \(p. 1282\)](#).

Une fois que vous avez connecté votre passerelle AWS IoT Core for LoRaWAN, vous pouvez vérifier l'état de votre connexion et obtenir des informations sur la date de réception de la dernière liaison montante à l'aide de la console ou de l'API.

Vérifier l'état de la connexion à la passerelle à l'aide de la console

Pour vérifier l'état de la connexion à l'aide de la console, accédez à la page [Passerelles](#) de la AWS IoT console et choisissez la passerelle que vous avez ajoutée. Dans la section Détails spécifiques au LoRa WAN de la page de détails de la passerelle, vous pouvez voir l'état de la connexion ainsi que la date et l'heure de réception de la dernière liaison montante.

Vérifiez l'état de la connexion à la passerelle à l'aide de l'API

Pour vérifier l'état de la connexion à l'aide de l'API, utilisez l'`GetWirelessGatewayStatistics` API. Cette API ne possède pas de corps de requête et contient uniquement un corps de réponse qui indique si la passerelle est connectée et quand le dernier lien ascendant a été reçu.

Note

L'heure à laquelle le dernier lien ascendant a été reçu, ou la `LastUplinkReceivedAt` valeur, n'est valide que pendant 3 mois.

```
HTTP/1.1 200
Content-type: application/json

{
  "ConnectionStatus": "Connected",
  "LastUplinkReceivedAt": "2021-03-24T23:13:08.476015749Z",
  "WirelessGatewayId": "30cbdcf3-86de-4291-bfab-5bfa2b12bad5"
}
```

Intégrer vos appareils à AWS IoT Core for LoRaWAN

Une fois que vous avez intégré votre passerelle AWS IoT Core for LoRaWAN et vérifié son état de connexion, vous pouvez intégrer vos appareils sans fil. Pour plus d'informations sur l'intégration de vos passerelles, consultez [Intégrer vos passerelles vers AWS IoT Core for LoRaWAN \(p. 1279\)](#).

Les appareils LoRaWAN utilisent un protocole LoRa WAN pour échanger des données avec des applications hébergées dans le cloud. AWS IoT Core for LoRaWAN prend en charge les appareils conformes aux spécifications LoRa WAN 1.0.x ou 1.1 normalisées par LoRa Alliance.

Un périphérique LoRa WAN contient généralement un ou plusieurs capteurs et acteurs. Les appareils envoient des données de télémétrie par liaison montante via des passerelles LoRa WAN à AWS IoT Core for LoRaWAN. Les applications hébergées dans le cloud peuvent contrôler les capteurs en envoyant des commandes de liaison descendante aux appareils LoRa WAN via des passerelles WAN.

Avant d'intégrer votre appareil sans fil

Avant d'intégrer votre appareil sans fil au système AWS IoT Core for LoRaWAN, vous devez d'abord disposer des informations suivantes :

Note

Facultativement, si les informations de position de votre appareil sont disponibles, vous pouvez les spécifier lors de l'intégration de votre appareil. Pour plus d'informations, veuillez consulter [Configurer la position des ressources sans fil avec AWS IoT Core for LoRa WAN \(p. 1300\)](#).

- LoRa Specification WAN et configuration des appareils sans fil

Le fait que les paramètres de configuration propres à chaque appareil soient prêts à être saisis à l'avance facilite la saisie des données dans la console. Les paramètres spécifiques que vous devez saisir dépendent de la spécification LoRa WAN utilisée par le périphérique. Pour obtenir la liste complète de ses spécifications et de ses paramètres de configuration, consultez la documentation de chaque appareil.

- 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 en savoir plus sur l'attribution de noms et la description de vos ressources, veuillez consulter [Décrivez vos ressourcesAWS IoT Core pour le LoRa WAN \(p. 1278\)](#).

- Profils d'appareils et de services

Préparez certains paramètres de configuration des appareils sans fil qui sont partagés par de nombreux appareils et peuvent être stockés AWS IoT Core for LoRaWAN sous forme de profils d'appareils et de services. Les paramètres de configuration se trouvent dans la documentation de l'appareil ou sur l'appareil lui-même. Vous devez identifier un profil d'appareil correspondant aux paramètres de configuration de l'appareil, ou en créer un si nécessaire, avant d'ajouter l'appareil. Pour plus d'informations, veuillez consulter [Ajoutez des profils àAWS IoT Core for LoRaWAN \(p. 1291\)](#).

- AWS IoT Core for LoRaWAN destination

Chaque appareil doit être affecté à une destination qui traitera ses messages à envoyer AWS IoT et à d'autres services. Les AWS IoT règles qui traitent et envoient les messages de l'appareil sont spécifiques au format du message de l'appareil. Pour traiter les messages provenant de l'appareil et les envoyer au service approprié, identifiez la destination que vous allez créer pour les messages de l'appareil et attribuez-la à l'appareil.

Pour intégrer votre appareil sans fil àAWS IoT Core for LoRaWAN

- [Ajoutez votre appareil sans fil àAWS IoT Core for LoRaWAN \(p. 1289\)](#)
- [Ajoutez des profils àAWS IoT Core for LoRaWAN \(p. 1291\)](#)
- [Ajouter des destinations àAWS IoT Core pour le LoRa WAN \(p. 1293\)](#)
- [Création de règles pour traiter les messages des appareils LoRa WAN \(p. 1296\)](#)
- [Connectez votre périphérique LoRa WAN et vérifiez son état de connexion \(p. 1299\)](#)

Ajoutez votre appareil sans fil àAWS IoT Core for LoRaWAN

Si vous ajoutez votre appareil Wireless pour la première fois, nous vous recommandons d'utiliser la console. Accédez à la page d'[AWS IoT Core for LoRaWAN](#) introduction de la AWS IoT console, choisissez Commencer, puis Ajouter un appareil. Si vous avez déjà ajouté un appareil, choisissez Afficher l'appareil pour afficher la passerelle que vous avez ajoutée. Si vous souhaitez ajouter d'autres appareils, choisissez Ajouter un appareil.

Vous pouvez également ajouter des appareils sans fil depuis la page [Appareils](#) de la AWS IoT console.

Note

Si vous ajoutez vos appareils depuis la page Appareils, vous pouvez également spécifier des informations de position facultatives pour votre appareil. Ces informations comprennent les coordonnées de position statiques, les solveurs de géolocalisation que vous souhaitez utiliser pour calculer la position de l'appareil et une destination qui décrit la AWS IoT règle qui traite les données de position de l'appareil à des fins d'utilisation par AWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Configurer la position des ressources sans fil avec AWS IoT Core for LoRa WAN \(p. 1300\)](#).

Ajoutez les spécifications de votre appareil sans fil àAWS IoT Core for LoRaWAN l'utilisation de la console

Choisissez une spécification de périphérique sans fil en fonction de votre méthode d'activation et de la version LoRa WAN. Une fois sélectionnées, vos données sont cryptées à l'aide d'une clé que vous AWS détenez et gérez pour vous.

Modes d'activation OTAA et ABP

Avant que votre appareil LoRa WAN puisse envoyer des données de liaison montante, vous devez effectuer un processus appelé procédure d'activation ou de connexion. Pour activer votre terminal, vous pouvez utiliser l'OTAA (activation sans fil) ou l'ABP (activation par personnalisation).

ABP ne nécessite pas de procédure de jointure et utilise des clés statiques. Lorsque vous utilisez OTAA, votre périphérique LoRa WAN envoie une demande de connexion et le serveur réseau peut accepter cette demande. Nous vous recommandons d'utiliser OTAA pour activer votre appareil, car de nouvelles clés de session sont générées à chaque activation, ce qui le rend plus sécurisé.

LoRaVersion WAN

Lorsque vous utilisez OTAA, votre appareil LoRa WAN et les applications hébergées dans le cloud partagent les clés racines. Ces clés racines varient selon que vous utilisez la version v1.0.x ou v1.1. La version v1.0.x ne possède qu'une seule clé racine AppKey (clé d'application) tandis que la version 1.1 possède deux clés racines, AppKey (clé d'application) et NwkKey (clé réseau). Les clés de session sont dérivées en fonction des clés racines de chaque activation. Les NwkKey et AppKeys sont des valeurs hexadécimales à 32 chiffres fournies par votre fournisseur de services sans fil.

eUI pour appareils Wireless

Après avoir sélectionné la spécification 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 propre à votre appareil et figurant sur l'étiquette de l'appareil ou sa documentation.
- AppEUI : valeur hexadémique à 16 chiffres propre au serveur de jointure et disponible dans la documentation de l'appareil. Dans la version LoRa WAN v1.1, l'AppEUI s'appelle JoinEUI.

Pour plus d'informations sur les identificateurs uniques, les clés de session et les clés racines, consultez la documentation de [LoRa l'Alliance](#).

Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'aide de l'API

Si vous ajoutez un appareil sans fil à l'aide de l'API, vous devez d'abord créer le profil de votre appareil et votre profil de service avant de créer l'appareil sans fil. Vous utiliserez le profil de l'appareil et l'ID du profil de service lors de la création du périphérique sans fil. Pour plus d'informations sur la création de ces profils à l'aide de l'API, consultez [Ajouter un profil d'appareil à l'aide de l'API \(p. 1292\)](#).

Les listes suivantes décrivent les actions d'API qui exécutent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoT WirelessActions d'API pour les profils de service

- [CreateWirelessDevice](#)
- [GetWirelessDevice](#)
- [ListWirelessDevices](#)
- [UpdateWirelessDevice](#)
- [DeleteWirelessDevice](#)

Pour obtenir la liste complète des actions et des types de données disponibles pour créer et gérer AWS IoT Core for LoRaWAN des ressources, consultez la [référence de l'AWS IoT Wireless API](#).

Comment utiliser le AWS CLI pour créer un appareil sans fil

Vous pouvez utiliser le AWS CLI pour créer un périphérique sans fil à l'aide de la [create-wireless-device](#) commande. L'exemple suivant crée un périphérique sans fil à l'aide d'un fichier input.json pour saisir 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 du fichier 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": "3f4aca100e2fc675ea123f4eb12c4a012",  
            "JoinEui": "b4c231a359bc2e3d",  
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"  
        },  
        "DevEui": "ac12efc654d23fc2"  
    },  
    "Name": "SampleIoTWirelessThing",  
    "Type": LoRaWAN  
}
```

Vous pouvez fournir ce fichier en entrée de la `create-wireless-device` commande.

```
aws iotwireless create-wireless-device \  
  --cli-input-json file://input.json
```

Pour plus d'informations sur les CLI que vous pouvez utiliser, consultez la [AWS CLI référence](#)

Ajoutez des profils à AWS IoT Core for LoRaWAN

Des profils de périphériques et de services peuvent être définis pour décrire les configurations de périphériques courantes. Ces profils décrivent les paramètres de configuration qui sont partagés par les appareils afin de faciliter l'ajout de ces appareils. AWS IoT Core for LoRaWAN prend en charge les profils d'appareils et les profils de service.

Les paramètres de configuration et les valeurs à saisir dans ces profils sont fournis par le fabricant de l'appareil.

Ajouter des profils d'appareil

Les profils de périphérique définissent les fonctionnalités du périphérique et les paramètres de démarrage que le serveur réseau utilise pour définir le service d'accès radio LoRa WAN. Il inclut la sélection de paramètres tels que LoRa la bande de fréquence, la version des paramètres LoRa régionaux et la version MAC de l'appareil. Pour en savoir plus sur les différentes bandes de fréquences, reportez-vous à la section [Envisagez de sélectionner des bandes de LoRa fréquences pour vos passerelles et la connexion de vos appareils \(p. 1280\)](#).

Ajouter un profil d'appareil à l'aide de la console

Si vous ajoutez un appareil sans fil à l'aide de la console comme décrit dans [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#), après avoir ajouté les spécifications du périphérique sans fil, vous pouvez ajouter le profil de votre appareil. Vous pouvez également ajouter des périphériques sans fil à partir de la page [Profils](#) de la AWS IoT console dans l'onglet LoRaWAN.

Vous pouvez choisir parmi les profils d'appareil par défaut ou créer un nouveau profil d'appareil. Nous vous recommandons d'utiliser les profils d'appareil par défaut. Si votre application vous oblige à créer un profil d'appareil, fournissez un nom de profil d'appareil, sélectionnez la bande de fréquences (RfRegion) que vous utilisez pour l'appareil et la passerelle, et conservez les valeurs par défaut des autres paramètres, sauf indication contraire dans la documentation de l'appareil.

Ajouter un profil d'appareil à l'aide de l'API

Si vous ajoutez un appareil sans fil à l'aide de l'API, vous devez créer le profil de votre appareil avant de créer l'appareil sans fil.

Les listes suivantes décrivent les actions d'API qui exécutent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoT WirelessActions d'API pour les profils de service

- [CreateDeviceProfile](#)
- [GetDeviceProfile](#)
- [ListDeviceProfiles](#)
- [UpdateDeviceProfile](#)
- [DeleteDeviceProfile](#)

Pour obtenir la liste complète des actions et des types de données disponibles pour créer et gérer AWS IoT Core for LoRaWAN des ressources, consultez la [référence de l'AWS IoT Wireless API](#).

Comment utiliser le AWS CLI pour créer un profil d'appareil

Vous pouvez utiliser le AWS CLI pour créer un profil d'appareil à l'aide de la [create-device-profile](#) commande. L'exemple suivant crée un profil d'appareil.

```
aws iotwireless create-device-profile
```

L'exécution de cette commande crée automatiquement un profil d'appareil avec un identifiant que vous pouvez utiliser lors de la création du périphérique sans fil. Vous pouvez désormais 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 plus d'informations sur les CLI que vous pouvez utiliser, consultez la [AWS CLI référence](#)

Ajouter des profils de service

Les profils de service décrivent les paramètres de communication dont l'appareil a besoin pour communiquer avec le serveur d'applications.

Ajouter un profil de service à l'aide de la console

Si vous ajoutez un appareil sans fil à l'aide de la console comme décrit dans [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#), 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 de la page [Profils](#) de la AWS IoT console dans l'onglet LoRaWAN.

Nous vous recommandons de laisser le paramètre AddGWMetaData activé afin de recevoir 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 appareil sans fil au moyen de l'API, vous devez d'abord créer votre profil de service au moyen de l'API.

Les listes suivantes décrivent les actions d'API qui exécutent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'un profil de service.

AWS IoT WirelessActions d'API pour les profils de service

- [CreateServiceProfile](#)
- [GetServiceProfile](#)
- [ListServiceProfiles](#)
- [DeleteServiceProfile](#)

Pour obtenir la liste complète des actions et des types de données disponibles pour créer et gérer AWS IoT Core for LoRaWAN des ressources, consultez la [référence de l'AWS IoT Wireless API](#).

Comment utiliser le AWS CLI pour créer un profil de service

Vous pouvez utiliser le AWS CLI pour créer un service à l'aide de la [create-service-profile](#) commande. 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 identifiant 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 pour le LoRa WAN

AWS IoT Core pour les destinations LoRa WAN, décrivez la AWS IoT règle qui traite les données d'un appareil destinées à être utilisées par AWS les services.

Étant donné que la plupart des périphériques LoRa WAN n'envoient pas AWS IoT Core de données vers le LoRa WAN dans un format utilisable par les AWS services, une AWS IoT règle doit d'abord les traiter. La AWS IoT règle contient l'instruction SQL qui interprète les données du périphérique et les actions de la règle thématique qui envoient le résultat de l'instruction SQL aux services qui l'utiliseront.

Si vous ajoutez votre destination pour la première fois, nous vous recommandons d'utiliser la console.

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 [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#), après avoir déjà ajouté les spécifications et les profils du périphérique sans fil AWS IoT Core for LoRaWAN comme décrit précédemment, vous pouvez continuer et ajouter une destination.

Vous pouvez également ajouter une AWS IoT Core for LoRaWAN destination depuis la page [Destinations](#) de la AWS IoT console.

Pour traiter les données d'un appareil, spécifiez les champs suivants lors de la création d'une destination AWS IoT pour le LoRa réseau étendu, puis choisissez Ajouter une destination.

- Détails sur la destination

Entrez un nom de destination et une description facultative pour votre destination.

- Nom de la règle

AWS IoT règle configurée pour évaluer les messages envoyés par votre appareil et traiter les données de l'appareil. Le nom de la règle sera mappé à votre destination. La destination a besoin de la règle pour traiter les messages qu'elle reçoit. Vous pouvez choisir les messages à traiter en invoquant une AWS IoT règle ou en les publiant sur le courtier de AWS IoT messages.

- Si vous choisissez Entrer un nom de règle, entrez un nom, puis choisissez Copier pour copier le nom de règle que vous allez saisir lors de la création de la AWS IoT règle. Vous pouvez soit choisir Créer une règle pour créer la règle maintenant, soit accéder au hub de [règles](#) de la AWS IoT console et créer une règle portant ce nom.

Vous pouvez également saisir une règle et utiliser le paramètre Avancé pour spécifier un nom de rubrique. Le nom de la rubrique est fourni lors de l'invocation de la règle et est accessible à l'aide de l'topic expression contenue dans la règle. Pour plus d'informations sur les règles du AWS IoT, consultez [Règles pour AWS IoT \(p. 524\)](#).

- Si vous choisissez Publier sur AWS IoT Message Broker, entrez un nom de rubrique. Vous pouvez ensuite copier le nom du sujet MQTT et plusieurs abonnés peuvent s'abonner à ce sujet pour recevoir les messages publiés sur ce sujet. Pour plus d'informations, veuillez consulter [Rubriques MQTT \(p. 115\)](#).

Pour plus d'informations sur AWS IoT les règles relatives aux destinations, consultez [Création de règles pour traiter les messages des appareils LoRa WAN \(p. 1296\)](#).

- Nom du rôle

Rôle IAM qui autorise les données de l'appareil à accéder à la règle nommée dans Nom de la règle. Dans la console, vous pouvez créer un nouveau rôle de service ou sélectionner un rôle de service existant. Si vous créez un nouveau rôle de service, vous pouvez saisir un nom de rôle (par exemple, **IoTWirelessDestinationRole**) ou le laisser vide AWS IoT Core for LoRaWAN pour générer un nouveau nom de rôle. AWS IoT Core for LoRaWAN créera ensuite automatiquement le rôle IAM avec les autorisations appropriées en votre nom.

Pour de plus amples informations sur les rôles IAM, veuillez consulter [Utilisation des rôles IAM](#).

Ajouter une destination à l'aide de l'API

Si vous souhaitez plutôt ajouter une destination à l'aide de l'interface de ligne de commande, vous devez déjà avoir créé la règle et le rôle IAM pour votre destination. Pour en savoir plus sur les détails dont une destination a besoin pour le rôle, consultez [Créer un rôle IAM pour vos destinations \(p. 1295\)](#).

La liste suivante contient les actions d'API qui exécutent les tâches associées à l'ajout, à la mise à jour ou à la suppression d'une destination.

AWS IoT WirelessActions d'API pour les destinations

- [CreateDestination](#)
- [GetDestination](#)
- [ListDestinations](#)

- [UpdateDestination](#)
- [DeleteDestination](#)

Pour obtenir la liste complète des actions et des types de données disponibles pour créer et gérer AWS IoT Core for LoRaWAN des ressources, consultez la [référence de l'AWS IoT Wireless API](#).

Comment utiliser le AWS CLI pour ajouter une destination

Vous pouvez utiliser le AWS CLI pour ajouter une destination à l'aide de la commande [create-destination](#). L'exemple suivant montre comment créer une destination en saisissant le nom d'une règle en utilisant RuleName comme valeur pour le expression-type paramètre. Si vous souhaitez spécifier un nom de rubrique pour la publication ou l'abonnement au courtier de messages, remplacez la valeur du expression-type paramètre par MqttTopic d.

```
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, consultez [Création de règles pour traiter les messages des appareils LoRaWAN \(p. 1296\)](#) et [Créer un rôle IAM pour vos destinations \(p. 1295\)](#).

Pour plus d'informations sur les CLI que vous pouvez utiliser, consultez la [AWS CLI référence](#).

Créer un rôle IAM pour vos destinations

AWS IoT Core for LoRaWAN les destinations nécessitent des rôles IAM qui confèrent AWS IoT Core for LoRaWAN les autorisations nécessaires pour envoyer des données à la AWS IoT règle. Si un tel rôle n'est pas déjà défini, vous devez le définir afin qu'il apparaisse dans la liste des rôles.

Lorsque vous utilisez la console pour ajouter une destination, un rôle IAM est AWS IoT Core for LoRaWAN automatiquement créé pour vous, comme décrit précédemment dans cette rubrique. Lorsque vous ajoutez une destination à l'aide de l'API ou de la CLI, vous devez créer le rôle IAM pour votre destination.

Pour créer une politique IAM pour votre rôle de AWS IoT Core for LoRaWAN destination

1. Ouvrez le [hub Policies de la console IAM](#).
2. Choisissez Créez une politique, puis choisissez l'onglet JSON.
3. Dans l'éditeur, supprimez tout contenu de l'éditeur et collez ce document de politique.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:DescribeEndpoint",
        "iot:Publish"
      ],
      "Resource": "*"
    }
  ]
}
```

4. Choisissez Réviser la politique, puis dans Nom, entrez le nom de cette politique. Vous aurez besoin de ce nom pour la procédure suivante.

Vous pouvez également décrire cette politique dans la section Description, si vous le souhaitez.

5. Choisissez Create Policy (Créer une politique).

Pour créer un rôle IAM pour une AWS IoT Core for LoRaWAN destination

1. Ouvrez le [hub Rôles de la console IAM](#) et choisissez Créer un rôle.
2. Dans Sélectionner le type d'entité sécurisée, choisissez AutreCompte AWS.
3. Dans Identifiant du compte, saisissez votreCompte AWS identifiant, puis choisissez Suivant : Autorisations.
4. Dans la zone de recherche, saisissez le nom de la stratégie IAM que vous avez créée dans la procédure précédente.
5. Dans les résultats de la recherche, vérifiez la stratégie IAM que vous avez créée dans la procédure précédente.
6. Sélectionnez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
7. Dans Nom du rôle, entrez le nom de ce rôle, puis choisissez Créer un rôle.
8. Dans le message de confirmation, choisissez le nom du rôle que vous avez créé pour modifier le nouveau rôle.
9. Dans Résumé, choisissez l'onglet Relations de confiance, puis choisissez Modifier la relation de confiance.
10. Dans le document de politique, modifiez laPrincipal propriété pour qu'elle ressemble à cet exemple.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Une fois laPrincipal propriété modifiée, le document de politique complet doit ressembler à 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, choisissez Mettre à jour la politique de confiance.

Une fois ce rôle défini, vous pouvez le trouver dans la liste des rôles lorsque vous configurez vos AWS IoT Core for LoRaWAN destinations.

Création de règles pour traiter les messages des appareils LoRa WAN

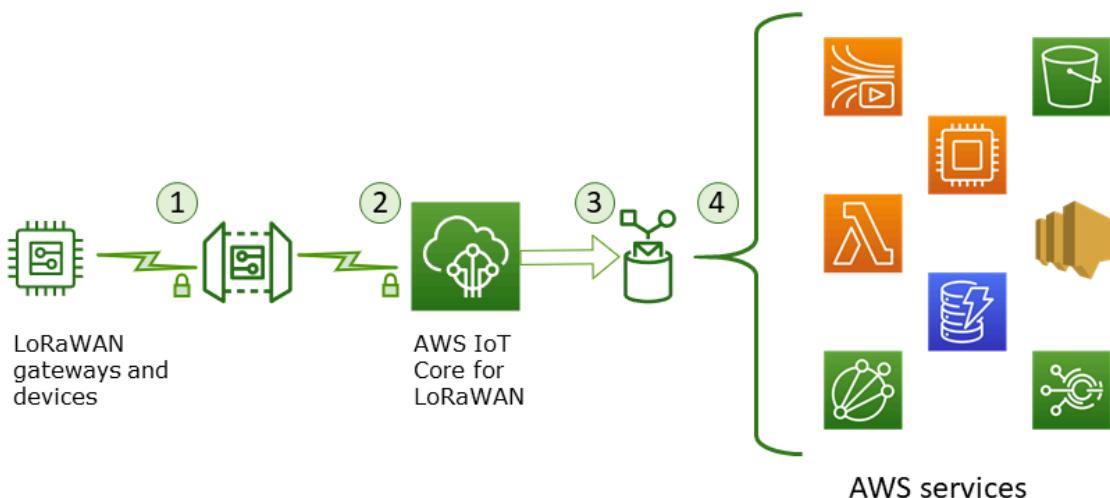
AWS IoT les règles envoient des messages de l'appareil à d'autres services. AWS IoT les règles peuvent également traiter les messages binaires reçus d'un périphérique LoRa WAN afin de les convertir dans d'autres formats afin de faciliter leur utilisation par d'autres services.

AWS IoT Core for LoRaWAN les [destinations \(p. 1293\)](#) associent un périphérique sans fil à la règle qui traite les données de message du périphérique pour les envoyer à d'autres services. La règle agit sur les données de l'appareil dès AWS IoT Core for LoRaWAN leur réception. AWS IoT Core for LoRaWAN les [destinations \(p. 1293\)](#) peuvent être partagées par tous les appareils dont les messages ont le même format de données et qui envoient leurs données au même service.

Comment AWS IoT les règles traitent les messages des appareils

La manière dont une AWS IoT règle traite les données des messages d'un appareil dépend du service qui recevra les données, du format des données de message de l'appareil et du format de données requis par le service. En général, la règle appelle une AWS Lambda fonction pour convertir les données des messages 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 lorsqu'elles sont transférées du périphérique sans fil vers un AWS service.



1. Le périphérique sans fil LoRa WAN chiffre ses messages binaires à l'aide du mode AES128 CTR avant de les transmettre.
2. AWS IoT Core for LoRaWAN déchiffre le message binaire et code la charge utile du message binaire déchiffré sous la forme d'une chaîne base64.
3. Le message codé en base64 qui en résulte 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) à la AWS IoT règle décrite dans la destination attribuée au périphérique.
4. La AWS IoT règle dirige les données du message vers le service décrit dans la configuration de la règle.

La charge utile binaire cryptée reçue du dispositif sans fil n'est ni modifiée ni interprétée par AWS IoT Core for LoRaWAN. La charge utile du message binaire déchiffré est codée uniquement sous la forme d'une chaîne base64. Pour que les services puissent accéder aux éléments de données contenus dans la charge utile du message binaire, les éléments de données doivent être extraits de la charge utile par une fonction appelée par la règle. La charge utile du message codée en base64 est une chaîne ASCII, elle peut donc être stockée telle quelle pour être analysée ultérieurement.

Création de règles pour le LoRa WAN

AWS IoT Core for LoRaWAN utilise des AWS IoT règles pour envoyer en toute sécurité des messages de l'appareil directement à d'autres AWS services sans avoir à utiliser le courtier de messages. En supprimant le courtier de messages du chemin d'ingestion, cela réduit les coûts et optimise le flux de données.

Pour qu'une AWS IoT Core for LoRaWAN règle puisse envoyer des messages de terminal à d'autres AWS services, elle nécessite une AWS IoT Core for LoRaWAN destination et une AWS IoT règle attribuée à cette destination. La AWS IoT règle doit contenir une instruction de requête SQL et au moins une action de règle.

En général, l'instruction de requête de AWS IoT règle se compose des éléments suivants :

- Clause SQL SELECT qui sélectionne et met en forme les données de la charge utile du message
- Un filtre de rubrique (l'objet FROM dans l'instruction de requête de règle) qui identifie les messages à utiliser
- Une instruction conditionnelle facultative (une 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
```

Lorsque vous créez AWS IoT des règles pour traiter des charges utiles à partir de périphériques LoRa WAN, il n'est pas nécessaire de spécifier la clause FROM dans l'objet de requête de règles. L'instruction de requête de règle doit comporter la clause SQL SELECT et peut éventuellement contenir la clause WHERE. Si l'instruction de requête utilise la clause FROM, elle est ignorée.

Voici un exemple d'instruction de requête de règle capable de traiter des charges utiles provenant de périphériques LoRa WAN :

```
SELECT WirelessDeviceId, WirelessMetadata.LoRaWAN.FPort as FPort,  
       WirelessMetadata.LoRaWAN.DevEui as DevEui,  
       PayloadData
```

Dans cet exemple, PayloadData il s'agit d'une charge utile binaire codée en base64 envoyée par votre périphérique LoRa WAN.

Voici un exemple d'instruction de requête de règle qui peut effectuer un décodage binaire de la charge utile entrante et la transformer dans 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, voir [Référence SQL AWS IoT \(p. 618\)](#)

Pour plus d'informations sur AWS IoT les règles et sur la manière de les créer et de les utiliser, consultez [Règles pour AWS IoT \(p. 524\)](#) et [Création de AWS IoT règles pour acheminer les données de l'appareil vers d'autres services \(p. 213\)](#).

Pour plus d'informations sur la création et l'utilisation de AWS IoT Core for LoRaWAN destinations, consultez [Ajouter des destinations à AWS IoT pour le LoRa WAN \(p. 1293\)](#).

Pour plus d'informations sur l'utilisation de charges utiles de messages binaires dans une règle, consultez [Utilisation des charges utiles binaires \(p. 684\)](#).

Pour plus d'informations sur la sécurité et le cryptage des données utilisés pour protéger la charge utile du message pendant son transfert, consultez [Protection des données dans AWS IoT Core \(p. 408\)](#).

Pour une architecture de référence présentant un exemple de décodage binaire et d'implémentation pour les règles de l'IoT, consultez la section [Exemples de AWS IoT Core for LoRaWAN solutions sur GitHub](#).

Connect votre périphérique LoRa WAN et vérifiez son état de connexion

Avant de pouvoir vérifier l'état de connexion de l'appareil, vous devez déjà avoir ajouté votre appareil et l'y avoir connecté AWS IoT Core for LoRaWAN. Pour plus d'informations sur la procédure à suivre pour ajouter votre appareil, consultez [Ajoutez votre appareil sans fil àAWS IoT Core for LoRaWAN \(p. 1289\)](#).

Après avoir ajouté votre appareil, consultez le manuel d'utilisation de votre appareil pour savoir comment lancer l'envoi d'un message de liaison montante depuis votre périphérique LoRa WAN.

Vérifier l'état de connexion du périphérique à l'aide de la console

Pour vérifier l'état de la connexion à l'aide de la console, accédez à la page [Appareils](#) de la AWS IoT console et choisissez l'appareil que vous avez ajouté. Dans la section Détails de la page de détails des appareils sans fil, vous pouvez voir la date et l'heure de réception de la dernière liaison montante.

Vérifiez l'état de connexion de l'appareil à l'aide de l'API

Pour vérifier l'état de la connexion à l'aide de l'API, utilisez l'`GetWirelessDeviceStatistics` API. Cette API ne possède pas de corps de requête et contient uniquement un corps de réponse qui indique la date de réception du dernier lien ascendant.

Note

L'heure à laquelle le dernier lien ascendant a été reçu, ou la `LastUplinkReceivedAt` valeur, n'est valide que pendant 3 mois.

```
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 de l'appareil en utilisant le [client de test MQTT](#) sur la page Test de la AWS IoT console. Pour plus d'informations, veuillez consulter [Afficher le format des messages de liaison montante envoyés depuis des périphériques LoRa WAN \(p. 1339\)](#).

Configurer la position des ressources sans fil avec AWS IoT Core for LoRa WAN

Avant d'utiliser cette fonctionnalité, notez que le fournisseur tiers choisi pour résoudre les informations de position des appareils LoRa WAN s'appuie sur des flux de données et des ensembles de données fournis ou gérés par le Service GNSS international (IGS), EarthData via la NASA ou d'autres tiers. Ces flux de données et ensembles de données constituent du contenu tiers (tel que défini dans le contrat client) et sont fournis tels quels. Pour plus d'informations, consultez [Conditions de service AWS](#).

Vous pouvez utiliser AWS IoT Core for LoRa WAN pour spécifier vos données de position statique, ou activer le positionnement pour identifier la position de votre appareil en temps réel à l'aide de solveurs tiers. Vous pouvez ajouter ou mettre à jour les informations de position pour les périphériques LoRa WAN ou les passerelles, ou les deux.

Vous spécifiez les informations de position soit lorsque vous ajoutez votre appareil ou votre passerelle AWS IoT Core à un LoRa réseau WAN, soit lorsque vous modifiez les détails de configuration de votre appareil ou de votre passerelle. Les informations de position sont spécifiées sous la forme d'une charge utile [GeoJSON](#). Le format GeoJSON est un format utilisé pour coder des structures de données géographiques. La charge utile contient les coordonnées de latitude et de longitude de la position de votre appareil, qui sont basées sur le système de [coordonnées du Système géodésique mondial \(WGS84\)](#).

Une fois que les solveurs ont calculé la position de votre ressource, si vous utilisez Amazon Location Service, vous pouvez activer une carte de localisation Amazon sur laquelle la position de votre ressource sera affichée. À l'aide des données de position, vous pouvez :

- Activez le positionnement pour identifier et obtenir la position de vos appareils LoRa WAN.
- Suivez et surveillez la position de vos passerelles et appareils.
- Définissez AWS IoT des règles qui traitent toutes les mises à jour des données de position et les acheminent vers d'autres services AWS. Pour obtenir la liste des actions des règles, consultez [Actions de règle AWS IoT \(p. 531\)](#).
- Créez des alertes et recevez des notifications sur les appareils en cas d'activité inhabituelle en utilisant les données de position et Amazon SNS.

Comment fonctionne le positionnement pour les appareils LoRa WAN

Vous pouvez activer le positionnement pour identifier la position de vos appareils à l'aide de solveurs WiFi et GNSS tiers. Ces informations peuvent être utilisées pour suivre et surveiller votre appareil. Les étapes suivantes expliquent comment activer le positionnement et comment afficher les informations de position des périphériques LoRa WAN.

Note

Les solveurs tiers ne peuvent être utilisés qu'avec des appareils LoRa WAN dotés de la puce [LoRa Edge](#). Il ne peut pas être utilisé avec des passerelles LoRa WAN. Pour les passerelles, vous pouvez toujours spécifier les informations de position statique et identifier la position sur une carte de localisation Amazon.

1. Ajoutez votre dispositif

Avant d'activer le positionnement, ajoutez d'abord votre appareil AWS IoT Core au LoRa réseau étendu. Le périphérique LoRa WAN doit être doté du chipset LoRa Edge, qui est une plate-forme à

très faible consommation d'énergie qui intègre un LoRa émetteur-récepteur longue portée, un scanner GNSS multiconstellation et un scanner MAC Wi-Fi passif ciblant les applications de géolocalisation.

2. Activer le positionnement

Pour obtenir la position en temps réel de vos appareils, activez le positionnement. Lorsque votre appareil LoRa WAN envoie un message de liaison montante, les données de numérisation Wi-Fi et GNSS contenues dans le message sont envoyées à AWS IoT Core for LoRa WAN via le port de trame de géolocalisation.

3. Eplace des informations sur la position

Récupérez la position estimée de l'appareil à partir des solveurs calculés sur la base des résultats de numérisation des émetteurs-récepteurs. Si les informations de position ont été calculées à l'aide des résultats d'analyse Wi-Fi et GNSS, AWS IoT Core pour LoRa WAN, la position estimée la plus précise est sélectionnée.

4. Afficher les informations de position

Une fois que le solveur a calculé les informations de position, il fournit également les informations de précision qui indiquent la différence entre la position calculée par les solveurs et les informations de position statique que vous avez saisies. Vous pouvez également consulter l'emplacement de l'appareil sur une carte de localisation d'Amazon.

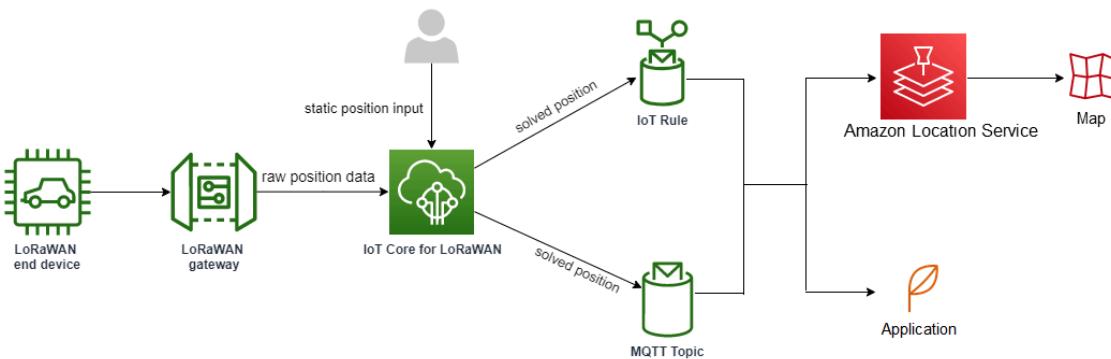
Note

Comme les solveurs ne peuvent pas être utilisés pour les passerelles LoRa WAN, les informations de précision seront signalées sous la forme 0.0 .

Pour plus d'informations sur le format des messages de liaison montante et les ports de fréquence utilisés pour le solveur de positionnement, consultez [Message de liaison montante entre AWS IoT Core et LoRa WAN et le moteur de règles \(p. 1305\)](#).

Présentation des flux de travail de positionnement

Le schéma suivant montre comment AWS IoT Core LoRa WAN stocke et met à jour les informations de position de vos appareils et passerelles.



1. Spécifiez la position statique de votre ressource

Spécifiez les informations de position statique de votre appareil ou de votre passerelle sous forme de charge utile GeoJSON, à l'aide des coordonnées de latitude et de longitude. Vous pouvez également spécifier une coordonnée d'altitude facultative. Ces coordonnées sont basées sur le système de coordonnées WGS84. Pour plus d'informations, voir [Système géodésique mondial \(WGS84\)](#).

2. Activer le positionnement des appareils

Si vous utilisez des appareils LoRa WAN dotés de la puce LoRa Edge, vous pouvez éventuellement activer le positionnement pour suivre la position de votre appareil en temps réel. Lorsque votre appareil envoie un message de liaison montante, les données de numérisation GNSS et Wi-Fi sont envoyées vers AWS IoT Core via le port de trame de géolocalisation. Les résolveurs utilisent ensuite ces informations pour déterminer la position du périphérique.

3. Ajouter une destination aux données de position de l'itinéraire

Vous pouvez ajouter une destination qui décrit la règle IoT pour le traitement des données de l'appareil et acheminer les informations de position mises AWS IoT Core à jour vers le LoRa WAN. Vous pouvez également consulter la dernière position connue de votre ressource sur une carte de localisation Amazon.

Configuration de votre position de ressource

Vous pouvez configurer la position de votre ressource à l'AWS Management Console avec l'API AWS IoT Wireless ou du AWS CLI.

Si vos appareils sont équipés de la puce LoRa Edge, vous pouvez activer le positionnement pour calculer les informations de position en temps réel. Pour vos passerelles, vous pouvez toujours saisir les coordonnées de position statiques et utiliser Amazon Location pour suivre la position de la passerelle sur une carte de localisation Amazon.

Configuration de la position des passerelles LoRa WAN

Lorsque vous ajoutez votre passerelle à AWS IoT Core for LoRa WAN, vous pouvez spécifier les données de position statiques. Si vous avez activé les cartes Amazon Location Service, les données de position sont affichées sur une carte de localisation Amazon.

Note

Les solveurs tiers ne peuvent pas être utilisés avec les passerelles LoRa WAN. Pour les passerelles, vous pouvez toujours spécifier les coordonnées de position statiques. Lorsque les solveurs ne sont pas utilisés pour calculer la position, comme dans le cas des passerelles, les informations de précision sont signalées sous la forme 0.0 .

Vous pouvez configurer la position de la passerelle à l'AWS Management Console avec l'API AWS IoT sans fil ou du AWS CLI.

Configuration de la position de votre passerelle à l'aide de la console

Pour configurer la position de vos ressources de passerelle à l'aide du AWS Management Console, connectez-vous d'abord à la console, puis accédez à la page [Gateways Hub](#) de la AWS IoT console.

Ajouter des informations sur position

Pour ajouter une configuration de position pour votre passerelle

1. Sur la page du hub de passerelles, choisissez Ajouter une passerelle.
2. Entrez l'EUI, la bande de fréquences (RFRegion) de la passerelle et tous les détails supplémentaires de la passerelle et les informations de configuration LoRa WAN. Pour plus d'informations, veuillez consulter [Ajouter une passerelle à l'aide de la console \(p. 1282\)](#).

3. Accédez à la section Informations de position - Facultative, et entrez les informations de position de votre passerelle en utilisant les coordonnées de latitude et de longitude, ainsi qu'une coordonnée d'altitude facultative. Les informations de position sont basées sur le système de coordonnées WGS84.

Afficher la position de la passerelle

Une fois que vous avez configuré la position de votre passerelle, AWS IoT Core for LoRa WAN crée une carte de localisation Amazon appelée `iotwireless.map`. Vous pouvez voir cette carte sur la page de détails de votre passerelle, dans l'onglet Position. En fonction des coordonnées de position que vous avez spécifiées, la position de votre passerelle sera affichée sous forme de marqueur sur la carte. Vous pouvez zoomer ou dézoomer pour voir clairement la position de votre passerelle sur la carte. Dans l'onglet Position, vous pouvez également voir les informations de précision et l'horodatage auxquels la position de votre passerelle a été déterminée.

Note

Si les cartes Amazon Location Service ne sont pas installées, vous verrez un message indiquant que vous devez utiliser Amazon Location Service pour accéder à la carte et visualiser la position de la passerelle. L'utilisation des cartes d'Amazon Location peut entraîner des frais supplémentaires pour votre Compte AWS. Pour en savoir plus, consultez [Pricing AWS IoT Core](#) (Tarification).

La carte `iotwireless.map`, agit comme une source de données cartographiques accessibles à l'aide d'opérations d'`GetAPI`, telles que [GetMapTile](#). Pour plus d'informations sur les API utilisées avec les cartes, consultez la [référence d'API Amazon Location Service](#).

Pour obtenir des informations supplémentaires sur cette carte, accédez à la console Amazon Location Service, choisissez cartes, puis [iotwireless.map](#). Pour plus d'informations, consultez [Maps](#) dans le guide du développeur Amazon Location Service.

Mettre à jour la configuration de la position de la passerelle

Pour modifier la configuration de position de la passerelle, sur la page de détails de la passerelle, choisissez Modifier, puis mettez à jour les informations de position et la destination.

Note

Les informations sur les données de position historiques ne sont pas disponibles. Lorsque vous mettez à jour les coordonnées de position de la passerelle, celle-ci remplace les données de position précédemment signalées. Après avoir mis à jour la position, dans l'onglet Position des détails de la passerelle, vous verrez les informations relatives à la nouvelle position. Le changement d'horodatage indique qu'il correspond à la dernière position connue de la passerelle.

Configurez la position de votre passerelle à l'aide de l'API

Vous pouvez spécifier les informations de position et configurer la position de la passerelle à l'aide de l'API AWS IoT sans fil ou du AWS CLI.

Important

Les actions d'API [UpdatePositionGetPosition](#), [PutPositionConfigurationGetPositionConfiguration](#), et ne [ListPositionConfiguration](#) sont plus prises en charge. Les appels visant à mettre à jour et à récupérer les informations de position doivent plutôt utiliser les opérations de [UpdateResourcePosition](#) l'API [GetResourcePosition](#) et.

Ajouter des informations sur position

Pour ajouter les informations de position statique pour une passerelle sans fil donnée, spécifiez les coordonnées à l'aide de l'opération [UpdateResourcePosition](#) API ou de la commande [update-resource-position](#) CLI. Spécifiez `WirelessGateway` sous la forme `ResourceType`, l'ID de la passerelle sans fil à

mettre à jour en tant que `ResourceIdentifier`, et les informations de position en tant que charge utile GeoJSON.

```
aws iotwireless update-resource-position \
--resource-type WirelessGateway \
--resource-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
--cli-input-json file://gatewayposition.json
```

L'exemple suivant affiche le contenu du fichier `gatewayposition.json`.

Contenu du fichier gatewayposition.json

```
{
  "type": "Point",
  "coordinates": [33.3318, -22.2155, 13.123],
  "properties": {
    "timestamp": "2018-11-30T18:35:24Z"
  }
}
```

L'exécution de cette commande ne produit aucune sortie. Pour voir les informations de position que vous avez spécifiées, utilisez l'opération `GetResourcePosition` API.

Obtenir des informations sur un poste

Pour obtenir les informations de position d'une passerelle sans fil donnée, utilisez l'opération [GetResourcePosition](#) API ou la commande `get-resource-position` CLI. Spécifiez `WirelessGateway` comme `resourceType` et fournissez l'ID de la passerelle sans fil comme `resourceIdentifier`.

```
aws iotwireless get-resource-position \
--resource-type WirelessGateway \
--resource-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

L'exécution de cette commande affiche les informations de position de votre passerelle sans fil sous forme de charge utile GeoJSON. Vous verrez des informations sur les coordonnées de position, le type d'informations de position et des propriétés supplémentaires, telles que l'horodatage qui correspond à la dernière position connue de la passerelle.

```
{
  {
    "type": "Point",
    "coordinates": [33.3318, -22.2155, 13.123],
    "properties": {
      "timestamp": "2018-11-30T18:35:24Z"
    }
  }
}
```

Configuration de la position des périphériques LoRa WAN

Lorsque vous ajoutez votre appareil à AWS IoT Core for LoRa WAN, vous pouvez spécifier les informations de position statique, éventuellement activer le positionnement et spécifier une destination. La destination décrit la règle IoT qui traite les informations de position de l'appareil et achemine la position mise à jour vers Amazon Location Service. Une fois que vous avez configuré la position de votre appareil, les données de position s'affichent sur une carte de localisation Amazon avec les informations de précision et la destination que vous avez spécifiées.

Vous pouvez configurer la position de votre appareil à l'AWS Management Console ou à l'aide de l'API AWS IoT sans fil ou du CLI AWS.

Ports de trame et format des messages de liaison montante

Si vous activez le positionnement, vous devez spécifier le port de la trame de géolocalisation pour communiquer les données de numérisation Wi-Fi et GNSS de l'appareil vers AWS IoT Core via LoRa WAN. Les informations de position sont communiquées AWS IoT Core au LoRa WAN via ce port de trame.

La spécification LoRa WAN fournit un champ de distribution de données (FRMPayload) et un champ de port (FPort) pour distinguer les différents types de messages. Pour communiquer les informations de position, vous pouvez spécifier une valeur comprise entre 1 et 223 pour le port de trame. Le port 0 est réservé aux messages MAC, le port 224 est réservé aux tests de conformité MAC et les ports 225 à 255 sont réservés aux futures extensions d'applications standardisées.

Message de liaison montante entre AWS IoT Core et LoRa WAN et le moteur de règles

Lorsque vous ajoutez une destination, une règle AWS IoT est créée pour acheminer les données vers Amazon Location Service à l'aide du moteur de règles. Les informations de position mises à jour sont ensuite affichées sur une carte de localisation Amazon. Si vous n'avez pas activé le positionnement, la destination achemine les données de position lorsque vous mettez à jour les coordonnées de position statiques de votre appareil.

Le code suivant indique le format du message de liaison montante envoyé AWS IoT Core depuis le LoRa réseau WAN avec les informations de position, la précision, la configuration du solveur et les métadonnées sans fil. Les champs soulignés ci-dessous sont facultatifs. S'il n'existe aucune information de précision verticale, la valeur est null.

```
{  
    // Position configuration parameters for given wireless device  
    "WirelessDeviceId": "5b58245e-146c-4c30-9703-0ca942e3ff35",  
  
    // Position information for a device in GeoJSON format. Altitude  
    // is optional. If no vertical accuracy information is available  
    // or positioning isn't activated, the value is set to null.  
    // The position information coordinates are listed in the order  
    // [longitude, latitude, altitude].  
    "coordinates": [33.33000183105469, -22.219999313354492, 99.0],  
    "type": "Point",  
    "properties": {  
        "horizontalAccuracy": number,  
        "verticalAccuracy": number,  
        "timestamp": "2022-08-19T03:08:35.061Z"  
    },  
  
    // Parameters controlled by IoT Core for LoRaWAN  
    "WirelessMetadata":  
    {  
        "LoRaWAN":  
        {  
            "ADR": false,  
            "Bandwidth": 125,  
            "ClassB": false,  
            "CodeRate": "4/5",  
            "DataRate": "0",  
            "DevAddr": "00b96cd4",  
            "DevEui": "58a0cb000202c99",  
            "FOptLen": 2,  
            "FCnt": 1,  
            "FPort": 1,  
            "Preamble": 8,  
            "SF": 12,  
            "TxPower": 23  
        }  
    }  
}
```

```
"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"
}
```

Configuration de la position de vos appareils à l'aide de la console

Pour configurer et gérer la position de vos appareils à l'aide du AWS Management Console, connectez-vous d'abord à la console, puis accédez à la page [Devices Hub](#) de la AWS IoT console.

Ajouter des informations sur position

Pour ajouter des informations de position pour votre appareil :

1. Sur la page Devices hub, choisissez Ajouter un appareil sans fil.
2. Entrez les spécifications du périphérique sans fil, les profils de l'appareil et du service, ainsi que la destination qui définit la règle IoT pour le routage des données vers d'autresService AWS. Pour plus d'informations, veuillez consulter [Intégrez vos appareils à AWS IoT Core for LoRaWAN \(p. 1288\)](#).
3. Entrez les informations de position, activez éventuellement la géolocalisation et spécifiez la destination des données de position que vous souhaitez utiliser pour le routage des messages.
 - Informations sur le poste

Spécifiez les données de position de votre appareil à l'aide des coordonnées de latitude et de longitude et d'une coordonnée d'altitude facultative. Les informations de position sont basées sur le système de coordonnées WGS84.

- Géolocalisation

Activez le positionnement si vous souhaitez que AWS IoT Core le LoRa WAN utilise la géolocalisation pour calculer la position de l'appareil. Il utilise des solveurs GNSS et Wi-Fi tiers pour identifier la position de votre appareil en temps réel.

Pour saisir les informations de géolocalisation, choisissez Activer le positionnement et entrez le port de la trame de géolocalisation pour communiquer les données de numérisation GNSS et Wi-Fi vers AWS IoT Core le LoRa WAN. Vous verrez les ports par défaut renseignés à titre de référence. Toutefois, vous pouvez choisir une valeur différente comprise entre 1 et 223.

- Destination des données de position

Choisissez une destination pour décrire la AWS IoT règle qui traite les données de position de l'appareil et les transmet vers le AWS IoT Core LoRa réseau étendu. Utilisez cette destination uniquement pour acheminer les données de position. Elle doit être différente de la destination que vous utilisez pour acheminer les données du périphérique vers d'autresService AWS.

Afficher la configuration de la position de l'appareil

Une fois que vous avez configuré la position de votre appareil, AWS IoT Core for LoRa WAN crée une carte de localisation Amazon appelée `iotwireless.map`. Vous pouvez voir cette carte sur la page de détails de votre appareil, dans l'onglet Position. En fonction des coordonnées de position que vous avez spécifiées ou de la position calculée par les solveurs tiers, la position de votre appareil sera affichée sous forme de marqueur sur la carte. Vous pouvez zoomer ou dézoomer pour voir clairement la position de votre appareil sur la carte. Sur la page de détails de l'appareil, dans l'onglet Position, vous pouvez également voir les informations de précision, l'horodatage auquel la position de votre appareil a été déterminée et la destination des données de position que vous avez spécifiée.

Note

Si vous n'avez pas activé les cartes Amazon Location Service, vous verrez un message indiquant que vous devez utiliser Amazon Location Service pour accéder à la carte et visualiser la position. L'utilisation des cartes d'Amazon Location peut entraîner des frais supplémentaires pour votre Compte AWS. Pour en savoir plus, consultez [Pricing AWS IoT Core](#) (Tarification).

La carte `iotwireless.map`, agit comme une source de données cartographiques accessibles à l'aide d'opérations d'`GetAPI`, telles que [GetMapTile](#). Pour plus d'informations sur les API utilisées avec les cartes, consultez la [référence d'API Amazon Location Service](#).

Pour obtenir des informations supplémentaires sur cette carte, accédez à la console Amazon Location Service, choisissez cartes, puis [iotwireless.map](#). Pour plus d'informations, consultez [Maps](#) dans le guide du développeur Amazon Location Service.

Mettre à jour la configuration de la position de l'appareil

Pour modifier la configuration de la position de l'appareil, sur la page des détails de l'appareil, choisissez Modifier, puis mettez à jour les informations de position, les paramètres de géolocalisation et la destination.

Note

Les informations sur les données de position historiques ne sont pas disponibles. Lorsque vous mettez à jour les coordonnées de position de l'appareil, celui-ci remplace les données de position précédemment signalées. Après avoir mis à jour la position, dans l'onglet Position des détails de l'appareil, vous verrez les nouvelles informations de position. Le changement d'horodatage indique qu'il correspond à la dernière position connue de l'appareil.

Configurer la position de l'appareil à l'aide de l'API

Vous pouvez spécifier les informations de position, configurer la position de l'appareil et activer la géolocalisation facultative à l'aide de l'API AWS IoT Wireless ou du AWS CLI.

Important

Les actions d'API [UpdatePositionGetPosition](#), [PutPositionConfigurationGetPositionConfiguration](#), et ne [ListPositionConfiguration](#)sont plus prises en charge. Les appels visant à mettre à jour et à récupérer les informations de position doivent plutôt utiliser les opérations de [UpdateResourcePosition](#)! API [GetResourcePosition](#)et.

Ajouter des informations de position et une configuration

Pour ajouter les informations de position pour un périphérique sans fil donné, spécifiez les coordonnées à l'aide de l'opération [UpdateResourcePosition](#) API ou de la commande [update-resource-position](#) CLI. Spécifiez `WirelessDevice` comme `ResourceType`, l'ID du périphérique sans fil à mettre à jour en tant que `ResourceIdentifier`, et les informations de position.

```
aws iotwireless update-resource-position \
```

```
--resource-type WirelessDevice \
--resource-id "1ffd32c8-8130-4194-96df-622f072a315f" \
--position [33.33, -33.33, 10.0]
```

L'exemple suivant affiche le contenu du fichier [deviceposition.json](#). Pour spécifier les valeurs FPort pour l'envoi des données de géolocalisation, utilisez l'objet [Positioning](#) avec les opérations [UpdateWirelessDevice](#)API [CreateWirelessDevice](#)et.

Contenu du fichier deviceposition.json

```
{
  "type": "Point",
  "coordinates": [33.3318, -22.2155, 13.123],
  "properties": {
    "verticalAccuracy": 707,
    "horizontalAccuracy": 389,
    "horizontalConfidenceLevel": 0.68,
    "verticalConfidenceLevel": 0.68,
    "timestamp": "2018-11-30T18:35:24Z"
  }
}
```

L'exécution de cette commande ne produit aucune sortie. Pour voir les informations de position que vous avez spécifiées, utilisez l'opération [GetResourcePosition](#)API.

Obtenir des informations de position et de configuration

Pour obtenir les informations de position d'un périphérique sans fil donné, utilisez l'[GetResourcePosition](#)API ou la commande [get-resource-position](#)CLI. Spécifiez `WirelessDevice` comme `resourceType` et fournissez l'ID du périphérique sans fil comme `resourceIdentifier`.

```
aws iotwireless get-resource-position \
--resource-type WirelessDevice \
--resource-id "1ffd32c8-8130-4194-96df-622f072a315f"
```

L'exécution de cette commande affiche les informations de position de votre appareil sans fil sous forme de charge utile GeoJSON. Vous verrez des informations sur les coordonnées de position, le type de position et les propriétés, qui peuvent inclure des informations de précision et l'horodatage correspondant à la dernière position connue de l'appareil.

```
{
  "type": "Point",
  "coordinates": [33.3318, -22.2155, 13.123],
  "properties": {
    "verticalAccuracy": 707,
    "horizontalAccuracy": 389,
    "horizontalConfidenceLevel": 0.68,
    "verticalConfidenceLevel": 0.68,
    "timestamp": "2018-11-30T18:35:24Z"
  }
}
```

ConnexionAWS IoT Core for LoRaWAN via un point de terminaison d'interface VPC

Au lieu de vous connecter [sur l'Internet public, vous pouvez vous connecter directement à des points de terminaison de VPC](#) AWS IoT Core for LoRaWAN via [Interface](#) dans votre Virtual Private Cloud (VPC).

PrivateLink Lorsque vous utilisez un point de terminaison d'un VPC d'interface, la communication entre votre VPC et AWS IoT Core for LoRaWAN est réalisée en totalité et en toute sécurité au sein du réseau AWS.

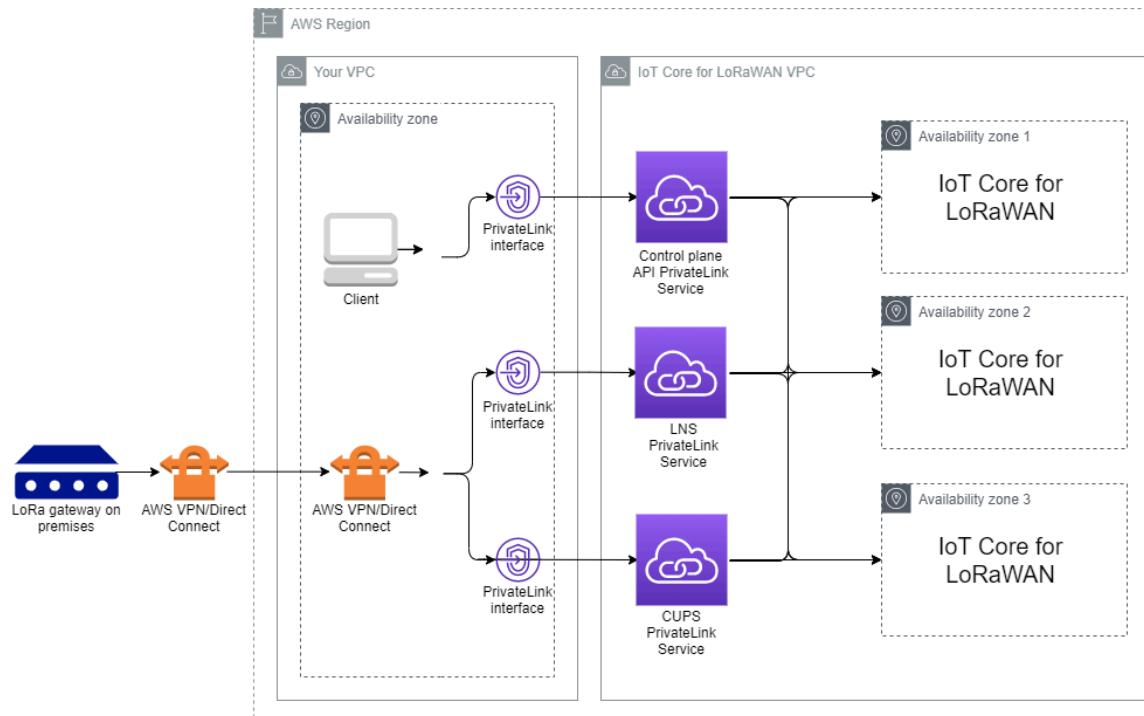
AWS IoT Core for LoRaWAN prend en charge les points de terminaison de l'interface Amazon Virtual Private Cloud alimentés par AWS PrivateLink. Chaque point de terminaison d'un VPC est représenté par une ou plusieurs interfaces réseau Elastic (ENI) avec des adresses IP privées dans vos sous-réseaux VPC.

Pour plus d'informations sur le VPC et les points de terminaison, consultez la section [Qu'est-ce qu'Amazon VPC ?](#)

Pour plus d'informations sur les points de terminaison [VPCAWS PrivateLink, consultezAWS PrivateLink la section et les points de terminaison VPC.](#)

AWS IoT Core for LoRaWAN architecture de liens privés

Le diagramme suivant illustre l'architecture Privatelink de AWS IoT Core for LoRaWAN. L'architecture utilise une Transit Gateway et un résolveur Route 53 pour partager les points de terminaison de l'AWS PrivateLink interface entre votre AWS IoT Core for LoRaWAN VPC, le VPC et un environnement sur site. Vous trouverez un schéma d'architecture plus détaillé lors de la configuration de la connexion aux points de terminaison de l'interface VPC.



Points de terminaison AWS IoT Core for LoRaWAN

AWS IoT Core for LoRaWAN possède trois points de terminaison publics. Chaque point de terminaison public possède un point de terminaison d'interface VPC correspondant. Les points de terminaison publics peuvent être classés en points de terminaison du plan de contrôle et du plan de données. Pour plus d'informations sur ces points de terminaison, consultez la section [Points de terminaison d'AWS IoT Core for LoRaWAN API.](#)

Note

AWS PrivateLink la prise en charge des points de terminaison est uniquement disponible dans les régions USA Est (Virginie du Nord) et Europe (Irlande).

- Points de terminaison de l'API du plan de contrôle

Vous pouvez utiliser les points de terminaison de l'API du plan de contrôle pour interagir avec lesAWS IoT Wireless API. Ces points de terminaison sont accessibles depuis un client hébergé dans votre Amazon VPC en utilisantAWS PrivateLink.

- Points de terminaison de l'API Data Plane

Les points de terminaison de l'API Data Plane sont des points de terminaison du serveur réseau LoRa WAN (LNS) et du serveur de configuration et de mise à jour (CUPS) que vous pouvez utiliser pour interagir avec les points de terminaisonAWS IoT Core for LoRaWAN LNS et CUPS. Ces points de terminaison sont accessibles depuis vos LoRa passerelles sur site en utilisantAWS VPN ouAWS Direct Connect. Vous obtenez ces points de terminaison lors de l'intégration de votre passerelle versAWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Ajouter une passerelle àAWS IoT Core for LoRaWAN \(p. 1282\)](#).

Les rubriques suivantes montrent comment intégrer ces points de terminaison.

Rubriques

- [Point de terminaison d'API du plan deAWS IoT Core for LoRaWAN contrôle intégré \(p. 1310\)](#)
- [Points de terminaison de l'API du plan deAWS IoT Core for LoRaWAN données \(p. 1313\)](#)

Point de terminaison d'API du plan deAWS IoT Core for LoRaWAN contrôle intégré

Vous pouvez utiliser les points de terminaison de l'API du plan deAWS IoT Core for LoRaWAN contrôle pour interagir avec lesAWS IoT Wireless API. Par exemple, vous pouvez utiliser ce point de terminaison pour exécuter l'[SendDataToWirelessDevice](#)API afin d'envoyer des donnéesAWS IoT à votre périphérique LoRa WAN. Pour plus d'informations, consultez [Points de terminaison de l'APIAWS IoT Core for LoRaWAN Control Plane](#)

Vous pouvez utiliser le client hébergé dans votre Amazon VPC pour accéder aux points de terminaison du plan de contrôle qui sont alimentés parAWS PrivateLink. Au lieu de vous connecter sur l'Internet public, vous pouvez vous connecter à l'AWS WirelessAPI via un point de terminaison d'interface dans votre Virtual Private Cloud (VPC).

Pour intégrer le point de terminaison du plan de contrôle :

- [Créez votre Amazon VPC et votre sous-réseau \(p. 1310\)](#)
- [Lancez une instance Amazon EC2 dans votre sous-réseau \(p. 1311\)](#)
- [Créer un point de terminaison de l'interface Amazon VPC \(p. 1311\)](#)
- [Tester votre connexion au point de terminaison de l'interface \(p. 1312\)](#)

Créez votre Amazon VPC et votre sous-réseau

Avant de pouvoir vous connecter au point de terminaison de l'interface, vous devez créer un VPC et un sous-réseau. Vous allez ensuite lancer une instance EC2 dans votre sous-réseau, que vous pouvez utiliser pour vous connecter au point de terminaison de l'interface.

Pour créer votre VPC, procédez comme suit :

1. Accédez à la page [VPC](#) de la console Amazon VPC et choisissez Créez un VPC.
2. Sur la page Créez un VPC :
 - Entrez un nom pour la balise VPC Name, facultatif (par exemple[VPC-A](#)).

- Entrez une plage d'adresses IPv4 pour votre VPC dans le bloc d'adresse CIDR IPv4 (par exemple,**10.100.0.0/16**).
3. Conservez les valeurs par défaut pour les autres champs et choisissez Create VPC.

Pour créer votre sous-réseau, procédez comme suit :

1. Accédez à la page [Sous-réseaux](#) de la console Amazon VPC et choisissez Créer un sous-réseau.
2. Sur la page Créer un sous-réseau :
 - Pour VPC ID, choisissez le VPC que vous avez créé précédemment (par exemple,VPC-A).
 - Attribuez un nom au sous-réseau (par exemple,**Private subnet**).
 - Choisissez la zone de disponibilité pour votre sous-réseau.
 - Entrez le bloc d'adresse IP de votre sous-réseau dans le bloc d'adresse CIDR IPv4 au format CIDR (par exemple,**10.100.0.0/24**).
3. Pour créer votre sous-réseau et l'ajouter à votre VPC, choisissez Créer un sous-réseau.

Pour plus d'informations, consultez [Utilisation des VPC et des sous-réseaux](#).

Lancez une instance Amazon EC2 dans votre sous-réseau

Pour lancer votre instance EC2 :

1. Accédez à la console [Amazon EC2](#) et choisissez Launch Instance.
2. Pour AMI, choisissez Amazon Linux 2 AMI (HVM), le type de volume SSD, puis choisissez le type de micro instance t2. Pour configurer les détails de l'instance, choisissez Next.
3. Sur la page Configurer les détails de l'instance :
 - Pour Réseau, choisissez le VPC que vous avez créé précédemment (par exemple,VPC-A).
 - Pour Sous-réseau, choisissez le sous-réseau que vous avez créé précédemment (par exemple,**Private subnet**).
 - Pour le rôle IAM, choisissez le rôle AWSIoTWirelessFullAccess pour accorder une politique d'accèsAWS IoT Core for LoRaWAN complète. Pour plus d'informations, consultez le [résuméAWSIoTWirelessFullAccess de la stratégie](#).
 - Pour Assume Private IP, utilisez une adresse IP, par exemple 10.100.0.42.
4. Choisissez Suivant : Ajouter du stockage, puis Suivant : Ajouter des balises. Vous pouvez éventuellement ajouter des balises à associer à votre instance EC2. Choisissez Suivant : Configurer le groupe de sécurité.
5. Sur la page Configurer le groupe de sécurité, configurez le groupe de sécurité pour autoriser :
 - Ouvrez tous les TCP pour Source en tant que**10.200.0.0/16**.
 - Ouvrez tous les fichiers ICMP - IPV4 pour Source as**10.200.0.0/16**.
6. Pour consulter les détails de l'instance et lancer votre instance EC2, choisissez Review and Launch (Lancer).

Pour plus d'informations, veuillez consulter [Premiers pas avec les instances Linux Amazon EC2](#).

Créer un point de terminaison de l'interface Amazon VPC

Vous pouvez créer un point de terminaison de VPC, qui sera ensuite accessible via l'API EC2. Pour créer le point de terminaison :

1. Accédez à la console [VPC Endpoints](#) et choisissez Create Endpoint.
2. Sur la page Create (Créer un point de terminaison), spécifiez les informations suivantes.

- Choisissez Service AWSs pour la catégorie de service.
- Dans le champ Nom du service, effectuez une recherche en saisissant le mot clé*iotwireless*.
Dans la liste des iotwireless services affichés, choisissez le point de terminaison de l'API du plan de contrôle pour votre région. Le point de terminaison sera au format `com.amazonaws.region.iotwireless.api`.
- Pour VPC et sous-réseaux, choisissez le VPC dans lequel vous voulez créer le point de terminaison et les zones de disponibilité (AZ) dans lesquelles vous voulez créer le réseau de points de terminaison.

Note

Le iotwireless service peut ne pas prendre en charge toutes les zones de disponibilité.

- Pour Activer le nom DNS, choisissez Activer pour ce point de terminaison.

Le choix de cette option résoudra automatiquement le DNS et créera une route d'entrée Amazon Route 53 Public Data Plane afin que les API que vous utiliserez ultérieurement pour tester la connexion passent par les points de terminaison privatelink.

- Pour Groupe de sécurité, choisissez les groupes de sécurité que vous voulez associer aux interfaces réseau du point de terminaison.
- Vous pouvez ajouter ou supprimer des balises. Les balises sont des paires nom-valeur que vous utilisez pour associer à votre point de terminaison.

3. Pour créer votre point de terminaison VPC, choisissez Créer un point de terminaison.

Tester votre connexion au point de terminaison de l'interface

Vous pouvez utiliser un SSH pour accéder à votre instance Amazon EC2, puis l'utiliser pour vous connecter AWS CLI aux points de terminaison de l'interface Privatelink.

Avant de vous connecter au point de terminaison de l'interface, téléchargez la AWS CLI version la plus récente en suivant les instructions décrites dans [Installation, mise à jour et désinstallation de AWS CLI la version 2 sous Linux](#).

Les exemples suivants montrent comment tester votre connexion au point de terminaison de l'interface à l'aide de la CLI.

```
aws iotwireless create-service-profile \
--endpoint-url https://api.iotwireless.region.amazonaws.com \
--name='test-privatelink'
```

Ce qui suit présente un exemple d'exécution de la commande.

```
Response:
{
  "Arn": "arn:aws:iotwireless:region:acct_number:ServiceProfile/1a2345ba-4c5d-67b0-ab67-e0c8342f2857",
  "Id": "1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
}
```

De même, vous pouvez exécuter les commandes suivantes pour obtenir les informations du profil de service ou répertorier tous les profils de service.

```
aws iotwireless get-service-profile \
--endpoint-url https://api.iotwireless.region.amazonaws.com
--id="1a2345ba-4c5d-67b0-ab67-e0c8342f2857"
```

Ce qui suit présente un exemple de list-device-profiles commande.

```
aws iotwireless list-device-profiles \
--endpoint-url https://api.iotwireless.region.amazonaws.com
```

Points de terminaison de l'API du plan deAWS IoT Core for LoRaWAN données

AWS IoT Core for LoRaWAN les extrémités du plan de données se composent des extrémités suivantes. Vous obtenez ces points de terminaison lorsque vous ajoutez votre passerelle à AWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Ajouter une passerelle à AWS IoT Core for LoRaWAN \(p. 1282\)](#).

- LoRaPoints de terminaison du serveur réseau WAN (LNS)

Les points de terminaison LNS sont du même format *account-specific-prefix.lns.lorawan.*region*.amazonaws.com*. Vous pouvez utiliser ce point de terminaison pour établir une connexion afin d'échanger des messages en liaison LoRa montante et en liaison descendante.

- Points de terminaison du serveur de configuration et de mise à jour (CUPS)

Les points de terminaison CUPS sont du même format *account-specific-prefix.cups.lorawan.*region*.amazonaws.com*. Vous pouvez utiliser ce point de terminaison pour la gestion des informations d'identification, la configuration à distance et la mise à jour du microprogramme des passerelles.

Pour plus d'informations, veuillez consulter [Utilisation des protocoles CUPS et LNS \(p. 1320\)](#).

Pour trouver les points de terminaison de l'API Data Plane pour votre région Compte AWS et votre région, utilisez la commande [get-service-endpoint](#) CLI présentée ici ou l'[GetServiceEndpoint](#) API REST. Pour plus d'informations, consultez [Points de terminaison de l'API AWS IoT Core for LoRaWAN Data Plane](#).

Vous pouvez connecter votre passerelle LoRa WAN sur site pour communiquer avec les AWS IoT Core for LoRaWAN points de terminaison. Pour établir cette connexion, connectez d'abord votre passerelle locale à celle de votre VPC Compte AWS à l'aide d'une connexion VPN. Vous pouvez ensuite communiquer avec les points de terminaison de l'interface du plan de données du AWS IoT Core for LoRaWAN VPC qui sont alimentés par Privatelink.

Ce qui suit montre comment intégrer ces points de terminaison.

- [Création d'un point de terminaison d'interface VPC et d'une zone hébergée privée \(p. 1313\)](#)
- [Utilisez un VPN pour connecter LoRa des passerelles à votre Compte AWS \(p. 1316\)](#)

Création d'un point de terminaison d'interface VPC et d'une zone hébergée privée

AWS IoT Core for LoRaWAN possède deux points de terminaison du plan de données : le point de terminaison du serveur de configuration et de mise à jour (CUPS) et le point de terminaison du serveur réseau LoRa WAN (LNS). Le processus de configuration pour établir une connexion par lien privé vers les deux points de terminaison est le même. Nous pouvons donc utiliser le point de terminaison LNS à des fins d'illustration.

Pour les points de terminaison de votre plan de données, les LoRa passerelles se connectent d'abord à votre Compte AWS dans votre Amazon VPC, qui se connecte ensuite au point de terminaison VPC du AWS IoT Core for LoRaWAN VPC.

Lors de la connexion aux points de terminaison, les noms DNS peuvent être résolus au sein d'un seul VPC, mais ne peuvent pas être résolus entre plusieurs VPC. Pour désactiver le DNS privé lors de la création du point de terminaison, désactivez le paramètre Activer le nom DNS. Vous pouvez utiliser une zone hébergée privée pour fournir des informations sur la façon dont vous voulez que Route 53 réponde aux requêtes DNS de vos VPC. Pour partager votre VPC avec un environnement sur site, vous pouvez utiliser un résolveur Route 53 afin de faciliter le DNS hybride.

Pour terminer cette procédure, effectuez les opérations suivantes.

- [Création d'un Amazon VPC et d'un sous-réseau \(p. 1314\)](#)
- [Création d'un point de terminaison d'interface Amazon VPC \(p. 1314\)](#)
- [Configuration d'une zone hébergée privée \(p. 1315\)](#)
- [Configurer le résolveur entrant Route 53 \(p. 1316\)](#)
- [Étapes suivantes \(p. 1316\)](#)

Création d'un Amazon VPC et d'un sous-réseau

Vous pouvez réutiliser votre Amazon VPC et votre sous-réseau que vous avez créés lors de l'intégration du point de terminaison de votre plan de contrôle. Pour plus d'informations, consultez [Créez votre Amazon VPC et votre sous-réseau \(p. 1310\)](#).

Création d'un point de terminaison d'interface Amazon VPC

Vous pouvez créer un point de terminaison VPC pour votre VPC, de la même manière que vous le feriez pour le point de terminaison de votre plan de contrôle.

1. Accédez à la console [VPC Endpoints](#) et choisissez Create Endpoint.
2. Sur la page Create (Créer un point de terminaison), spécifiez les informations suivantes.
 - Choisissez Service AWSs pour la catégorie de service.
 - Dans le champ Nom du service, effectuez une recherche en saisissant le mot clé **lns**. Dans la liste des services affichés, choisissez le point de terminaison de l'API du plan de données LNS pour votre région. Le point de terminaison sera au format `com.amazonaws.region.lorawan.lns`.

Note

Si vous suivez cette procédure pour votre point de terminaison CUPS, recherchez `cups`. Le point de terminaison sera au format `com.amazonaws.region.lorawan.cups`.

- Pour VPC et sous-réseaux, choisissez le VPC dans lequel vous voulez créer le point de terminaison et les zones de disponibilité (AZ) dans lesquelles vous voulez créer le réseau de points de terminaison.

Note

Le `iotwireless` service peut ne pas prendre en charge toutes les zones de disponibilité.

- Pour Activer le nom DNS, assurez-vous que l'option Activer pour ce point de terminaison n'est pas sélectionnée.

En ne sélectionnant pas cette option, vous pouvez désactiver le DNS privé pour le point de terminaison du VPC et utiliser une zone hébergée privée à la place.

- Pour Groupe de sécurité, choisissez les groupes de sécurité que vous voulez associer aux interfaces réseau du point de terminaison.
- Vous pouvez ajouter ou supprimer des balises. Les balises sont des paires nom-valeur que vous utilisez pour associer à votre point de terminaison.

3. Pour créer votre point de terminaison VPC, choisissez Créez un point de terminaison.

Configuration d'une zone hébergée privée

Après avoir créé le point de terminaison privatelink, dans l'onglet Détails de votre point de terminaison, vous verrez une liste de noms DNS. Vous pouvez utiliser l'un de ces noms DNS pour configurer votre zone hébergée privée. Le nom DNS sera au format `vpce-xxxx.lns.lorawan.region.vpce.amazonaws.com`.

Créer la zone hébergée privée

Pour créer la zone hébergée privée :

1. Accédez à la console [Route 53 Hosted Zones](#) et choisissez Créer une zone hébergée.
2. Sur la page Create (Créer une zone hébergée), spécifiez les informations suivantes.
 - Dans le champ Nom de domaine, entrez le nom de service complet pour votre point de terminaison `LNS.lns.lorawan.region.amazonaws.com`.

Note

Si vous suivez cette procédure pour votre point de terminaison CUPS, entrez `cups.lorawan.region.amazonaws.com`.

- Pour Type, choisissez Zone hébergée privée.
 - Vous pouvez éventuellement ajouter ou supprimer des balises à associer à votre zone hébergée.
3. Pour créer votre zone hébergée privée, choisissez Créer une zone hébergée.

Pour plus d'informations, consultez [Création d'une zone hébergée privée](#).

Après avoir créé une zone hébergée privée, vous pouvez créer un enregistrement qui indique au DNS comment vous souhaitez que le trafic soit acheminé vers ce domaine.

Créer un enregistrement

Après avoir créé une zone hébergée privée, vous pouvez créer un enregistrement qui indique au DNS comment vous souhaitez que le trafic soit acheminé vers ce domaine. Pour créer un enregistrement, procédez comme suit :

1. Dans la liste des zones hébergées qui s'affiche, choisissez la zone hébergée privée que vous avez créée précédemment et sélectionnez Create (Créer un).
2. Utilisez la méthode de l'assistant pour créer l'enregistrement. Si la console vous propose la méthode de création rapide, choisissez Passer à l'assistant.
3. Choisissez Routage simple pour la politique de routage, puis choisissez Suivant.
4. Sur la page Configurer les enregistrements, choisissez Définir un enregistrement simple.
5. Dans la page Définir un enregistrement simple :
 - Dans le champ Nom de l'enregistrement, entrez l'alias de votre Compte AWS numéro. Vous obtenez cette valeur lors de l'intégration de votre passerelle ou en utilisant l'[GetServiceEndpoint](#) API REST.
 - Pour le type d'enregistrement, conservez la valeur telle que `A - Routes traffic to an IPv4 address and some AWS resources`.
 - Pour Valeur/Acheminer le trafic vers, choisissez Alias vers le point de terminaison VPC. Choisissez ensuite votre région, puis choisissez le point de terminaison que vous avez créé précédemment, comme décrit dans [Création d'un point de terminaison d'interface Amazon VPC \(p. 1314\)](#) la liste des points de terminaison affichée.
6. Choisissez Définir un enregistrement simple pour créer votre enregistrement.

Configurer le résolveur entrant Route 53

Pour partager un point de terminaison VPC avec un environnement sur site, un résolveur Route 53 peut être utilisé pour faciliter le DNS hybride. Le résolveur entrant vous permettra d'acheminer le trafic depuis le réseau local vers les points de terminaison du plan de données sans passer par l'Internet public. Pour renvoyer les valeurs d'adresse IP privée de votre service, créez le résolveur Route 53 dans le même VPC que le point de terminaison du VPC.

Lorsque vous créez le résolveur entrant, il vous suffit de spécifier votre VPC et les sous-réseaux que vous avez créés précédemment dans vos zones de disponibilité (AZ). Le résolveur Route 53 utilise ces informations pour attribuer automatiquement une adresse IP afin d'acheminer le trafic vers chacun des sous-réseaux.

Pour créer le résolveur entrant, procédez comme suit :

1. Accédez à la console [Route 53 Inbound Endpoints](#) et choisissez Create Inbound Endpoints.

Note

Assurez-vous d'utiliser la même méthode que celleRégion AWS que vous avez utilisée lors de la création du point de terminaison et de la zone hébergée privée.

2. Sur la page Create (Créer un point de terminaison), spécifiez les informations suivantes.
 - Attribuez un nom au point de terminaison (par exemple,**VPC_A_Test**).
 - Pour le VPC de la région, choisissez le même VPC que celui que vous avez utilisé lors de la création du point de terminaison du VPC.
 - Configurez le groupe de sécurité pour ce point de terminaison afin d'autoriser le trafic entrant en provenance du réseau local.
 - Pour Adresse IP, choisissez Utiliser une adresse IP sélectionnée automatiquement.
3. Choisissez Soumettre pour créer votre résolveur entrant.

Pour cet exemple, supposons que les adresses IP 110.100.0.145 et 210.100.192.10 aient été attribuées au résolveur Route 53 entrant pour le routage du trafic.

Étapes suivantes

Vous avez créé la zone hébergée privée et un résolveur entrant pour acheminer le trafic vers vos entrées DNS. Vous pouvez désormais utiliser un point de terminaison de Site-to-Site VPN. Pour plus d'informations, veuillez consulter [Utilisez un VPN pour connecter LoRa des passerelles à votreCompte AWS \(p. 1316\)](#).

Utilisez un VPN pour connecter LoRa des passerelles à votreCompte AWS

Pour connecter vos passerelles sur siteCompte AWS, vous pouvez utiliser une connexion Site-to-Site VPN ou un point de terminaison Client VPN.

Avant de pouvoir connecter vos passerelles locales, vous devez avoir créé le point de terminaison VPC et configuré une zone hébergée privée et un résolveur entrant afin que le trafic provenant des passerelles ne passe pas par l'Internet public. Pour plus d'informations, veuillez consulter [Création d'un point de terminaison d'interface VPC et d'une zone hébergée privée \(p. 1313\)](#).

Point de terminaison d'un Site-to-Site VPN

Si vous ne disposez pas du matériel de passerelle ou si vous souhaitez tester la connexion VPN à l'aide d'une autreCompte AWS, vous pouvez utiliser une connexion Site-to-Site VPN. Vous pouvez utiliser un Site-to-Site VPN pour vous connecter aux points de terminaison VPC à partir du même pointCompte AWS ou d'un autre pointCompte AWS que vous utilisez peut-être sur un autreRégion AWS.

Note

Si vous avez le matériel de la passerelle avec vous et que vous souhaitez configurer une connexion VPN, nous vous recommandons d'utiliser le Client VPN à la place. Pour des instructions, consultez [Point de terminaison VPN Client \(p. 1317\)](#).

Pour configurer un Site-to-Site VPN, procédez comme suit :

1. Créez un autre VPC sur le site à partir duquel vous souhaitez établir la connexion. EnVPC-A effet, vous pouvez réutiliser le VPC que vous avez créé précédemment. Pour créer un autre VPC (par exemple, VPC-B), utilisez un bloc d'adresse CIDR qui ne chevauche pas le bloc d'adresse CIDR du VPC que vous avez créé précédemment.

Pour plus d'informations sur la configuration des VPC, suivez les instructions décrites dans [AWS Configuration de la connexion Site-to-Site VPN](#).

Note

La méthode Site-to-Site VPN décrite dans le document utilise OpenSwan pour la connexion VPN, qui ne prend en charge qu'un seul tunnel VPN. Si vous utilisez un autre logiciel commercial pour le VPN, vous pourrez peut-être configurer deux tunnels entre les sites.

2. Après avoir configuré la connexion VPN, mettez à jour le /etc/resolv.conf fichier en ajoutant l'adresse IP du résolveur entrant provenant de votreCompte AWS. Vous utilisez cette adresse IP pour le serveur de noms. Pour plus d'informations sur la procédure d'obtention de cette adresse IP, consultez [Configurer le résolveur entrant Route 53 \(p. 1316\)](#). Pour cet exemple, nous pouvons utiliser l'adresse IP 10.100.0.145 qui a été attribuée lorsque vous avez créé le résolveur Route 53.

```
options timeout:2 attempts:5
; generated by /usr/sbin/dhclient-script
search region.compute.internal
nameserver 10.100.0.145
```

3. Nous pouvons maintenant tester si la connexion VPN utilise le AWS PrivateLink point de terminaison au lieu de passer par l'Internet public à l'aide d'un nslookup commande. Ce qui suit présente un exemple d'exécution de la commande.

```
nslookup account-specific-prefix.lns.lorawan.region.amazonaws.com
```

Ce qui suit montre un exemple de sortie de l'exécution de la commande, qui affiche une adresse IP privée indiquant que la connexion a été établie avec le point de terminaison AWS PrivateLink LNS.

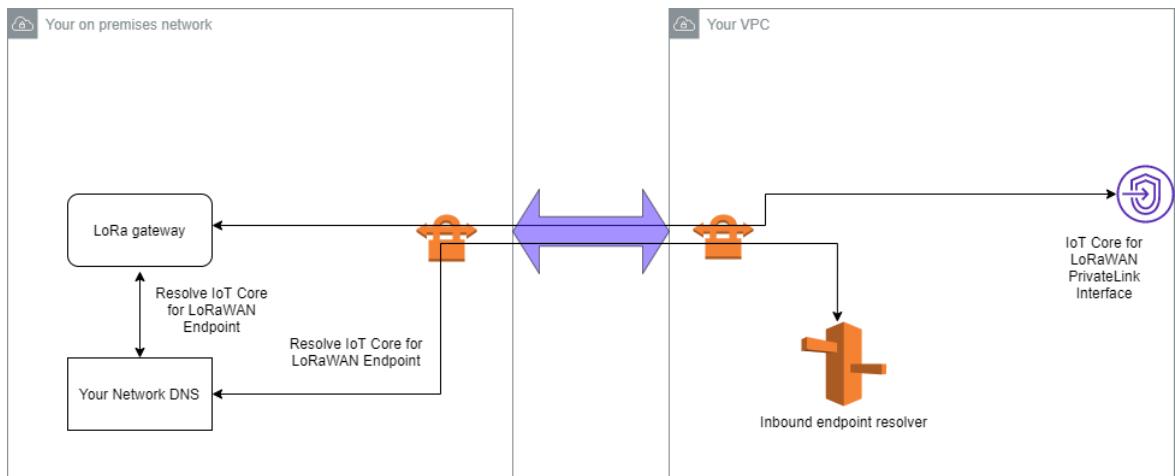
```
Server: 10.100.0.145
Address: 10.100.0.145

Non-authoritative answer:
Name: https://xxxxx.lns.lorawan.region.amazonaws.com
Address: 10.100.0.204
```

Pour plus d'informations sur l'utilisation d'une connexion Site-to-Site VPN, consultez [Fonctionnement d'un Site-to-Site VPN](#).

Point de terminaison VPN Client

AWS Client VPN est un service VPN géré basé sur le client qui vous permet d'accéder de façon sécurisée à vos ressources AWS et aux ressources de votre réseau sur site. L'architecture du service VPN client est présentée ci-dessous.



Pour établir une connexion VPN à un point de terminaison Client VPN comme suit :

1. Créez un point de terminaison Client VPN en suivant les instructions décrites dans la section [Mise en routeAWS Client VPN](#).
2. Connectez-vous à votre réseau local (par exemple, un routeur Wi-Fi) à l'aide de l'URL d'accès de ce routeur (par exemple, 192.168.1.1) et recherchez le nom racine et le mot de passe.
3. Configurez votre passerelle LoRa WAN en suivant les instructions de la documentation de la passerelle, puis ajoutez-y votre passerelleAWS IoT Core for LoRaWAN. Pour plus d'informations sur la procédure d'ajout de votre passerelle, consultez[Intégrez vos passerelles versAWS IoT Core for LoRaWAN \(p. 1279\)](#).
4. Vérifiez si le micrologiciel de votre passerelle est à jour. Si le microprogramme n'est pas à jour, vous pouvez suivre les instructions fournies sur le réseau local pour mettre à jour le microprogramme de votre passerelle. Pour plus d'informations, veuillez consulter [Mettez à jour le firmware de la passerelle à l'aide du service CUPSAWS IoT Core for LoRaWAN \(p. 1324\)](#).
5. Vérifiez si OpenVPN a été activé. S'il a été activé, passez à l'étape suivante pour configurer le client OpenVPN au sein du réseau local. S'il n'a pas été activé, suivez les instructions du [Guide d'installation d'OpenVPN pour OpenWrt](#).

Note

Dans le cadre de cet exemple, nous utilisons OpenVPN. Vous pouvez utiliser d'autres clients VPN tels queAWS VPN ouAWS Direct Connect pour configurer votre connexion Client VPN.

6. Configurez le client OpenVPN en fonction des informations provenant de la configuration du client et de la manière dont vous pouvez [utiliser le client OpenVPN LuCi](#).
7. Connectez-vous via SSH à votre réseau local et mettez à jour le /etc/resolv.conf fichier en ajoutant l'adresse IP du résolveur entrant dans votreCompte AWS (10.100.0.145).
8. Pour que le trafic de passerelle soit utiliséAWS PrivateLink pour se connecter au point de terminaison, remplacez la première entrée DNS de votre passerelle par l'adresse IP du résolveur entrant.

Pour plus d'informations sur l'utilisation d'une connexion Site-to-Site VPN, veuillez consulter [Premiers pas avec Client VPN](#).

Connect à des points de terminaison VPC LNS et CUPS

Ce qui suit montre comment tester votre connexion aux points de terminaison VPC LNS et CUPS.

Tester le terminal CUPS

Pour tester votre AWS PrivateLink connexion au point de terminaison CUPS depuis votre LoRa passerelle, exécutez la commande suivante :

```
curl -k -v -X POST https://xxxx.cups.region.iotwireless.iot:443/update-info
    --cacert cups.trust --cert cups.crt --key cups.key --header "Content-Type:
application/json"
    --data '{
        "router": "xxxxxxxxxxxxxx",
        "cupsUri": "https://xxxx.cups.lorawan.region.amazonaws.com:443",
        "cupsCredCrc":1234, "tcCredCrc":552384314
    }'
    -output cups.out
```

Tester le point de terminaison LNS

Pour tester votre point de terminaison LNS, commencez par LoRa configurer un périphérique WAN qui fonctionnera avec votre passerelle sans fil. Vous pouvez ensuite ajouter votre appareil et exécuter la procédure de connexion, après quoi vous pouvez commencer à envoyer des messages de liaison montante.

Gestion des passerelles avec AWS IoT Core for LoRa WAN

Les passerelles agissent comme un pont et transportent les données des appareils LoRa WAN vers et depuis un serveur réseau, généralement via des réseaux à large bande passante tels que Wi-Fi, Ethernet ou Cellular. Les passerelles WAN connectent des appareils sans fil à AWS IoT Core for LoRaWAN.

Voici quelques points importants à prendre en compte lors de l'utilisation de vos passerelles avec AWS IoT Core for LoRaWAN. Pour plus d'informations sur l'ajout de votre passerelle à AWS IoT Core for LoRaWAN, consultez [Intégrer vos passerelles vers AWS IoT Core for LoRaWAN \(p. 1279\)](#).

LoRa Configuration logicielle requise pour Basics Station

Pour s'y connecter AWS IoT Core for LoRaWAN, le logiciel [LoRa Basics Station](#) doit être exécuté sur votre passerelle LoRa WAN. LoRa Basics Station est un logiciel open source maintenu par Semtech Corporation et distribué par leur [GitHub](#) référentiel. AWS IoT Core for LoRaWAN prend en charge LoRa les versions 2.0.4 et ultérieures de Basics Station.

Utilisation de passerelles qualifiées figurant dans le catalogue d'appareils AWS partenaires

Le [catalogue d'appareils AWS partenaires](#) contient des passerelles et des kits de développement pouvant être utilisés avec AWS IoT Core for LoRaWAN. Nous vous recommandons d'utiliser ces passerelles qualifiées car vous n'avez pas à modifier le logiciel d'intégration auquel les passerelles sont connectées AWS IoT Core. Ces passerelles disposent déjà d'une version du BasicStation logiciel compatible avec AWS IoT Core for LoRaWAN.

Note

Si vous possédez une passerelle qui ne figure pas dans le catalogue des partenaires en tant que passerelle qualifiée AWS IoT Core for LoRaWAN, vous pourrez peut-être toujours l'utiliser

si la passerelle exécute le logiciel LoRa Basics Station avec la version 2.0.4 et les versions ultérieures. Assurez-vous d'utiliser l'authentification du serveur et du client TLS pour authentifier votre passerelle LoRa WAN.

Utilisation des protocoles CUPS et LNS

Le logiciel Basics Station contient deux sous-protocoles pour connecter des passerelles à des serveurs réseau, les protocoles LoRa WAN Network Server (LNS) et Configuration and Update Server (CUPS).

Le protocole LNS établit une connexion de données entre une passerelle compatible LoRa Basics Station et un serveur réseau. LoRa les messages de liaison montante et descendante sont échangés via cette connexion de données sécurisée WebSockets.

Le protocole CUPS permet la gestion des informations d'identification, ainsi que la configuration à distance et la mise à jour du microprogramme des passerelles. AWS IoT Core for LoRaWAN fournit à la fois des points de terminaison LNS et CUPS pour l'ingestion de données via le LoRa WAN et la gestion à distance des passerelles, respectivement.

Pour plus d'informations, consultez les [sections Protocole LNS](#) et [Protocole CUPS](#).

Configurez les fonctionnalités de balisage et de filtrage de vos passerelles LoRa WAN

Lorsque vous travaillez avec des périphériques LoRa WAN, vous pouvez configurer certains paramètres facultatifs pour vos passerelles LoRa WAN. Les paramètres incluent :

- **Balisage**

Vous pouvez configurer des paramètres de balisage pour vos passerelles LoRa WAN qui font office de pont pour vos périphériques LoRa WAN de classe B. Ces appareils reçoivent un message de liaison descendante à des créneaux horaires planifiés. Vous devez donc configurer les paramètres de balisage pour que vos passerelles transmettent ces balises synchronisées dans le temps.

- **Filtrage**

Vous pouvez configurer les `JoinEUI` paramètres `NetID` et de vos passerelles LoRa WAN afin de filtrer le trafic de données des appareils. Le filtrage du trafic permet de préserver l'utilisation de la bande passante et de réduire le flux de trafic entre les passerelles et le LNS.

- **Sous-bandes**

Vous pouvez configurer les sous-bandes de votre passerelle afin de spécifier la sous-bande particulière que vous souhaitez utiliser. Pour les appareils sans fil qui ne peuvent pas passer d'une sous-bande à l'autre, vous pouvez utiliser cette fonctionnalité pour communiquer avec les appareils en utilisant uniquement les canaux de fréquence de cette sous-bande particulière.

Les rubriques suivantes contiennent des informations supplémentaires sur ces paramètres et sur la manière de les configurer. Les paramètres de balisage ne sont pas disponibles dans le AWS Management Console et ne peuvent être spécifiés qu'à l'aide de l'AWS IoT Wireless API ou du AWS CLI.

Rubriques

- [Configuration de vos passerelles pour envoyer des balises à des appareils de classe B \(p. 1321\)](#)
- [Configuration des sous-bandes et des capacités de filtrage de votre passerelle \(p. 1322\)](#)

Configuration de vos passerelles pour envoyer des balises à des appareils de classe B

Si vous intégrez des appareils sans fil de classe B àAWS IoT Core for LoRaWAN, les appareils reçoivent des messages de liaison descendante dans des plages horaires planifiées. Les appareils ouvrent ces emplacements en fonction de balises synchronisées dans le temps transmises par la passerelle. Pour que vos passerelles transmettent ces balises synchrones, vous pouvez les utiliserAWS IoT Core for LoRaWAN pour configurer certains paramètres liés aux balises pour les passerelles.

Pour configurer ces paramètres de balisage, votre passerelle doit exécuter la version 2.0.6 du logiciel LoRa Basics Station. Consultez [Utilisation de passerelles qualifiées figurant dans le catalogue d'appareils AWS partenaires \(p. 1319\)](#).

Comment configurer les paramètres de balisage

Note

Vous ne devez configurer les paramètres de balisage de votre passerelle que si elle communique avec un périphérique sans fil de classe B.

Vous configurez les paramètres de balisage lorsque vous ajoutez votre passerelle àAWS IoT Core for LoRaWAN l'utilisation de l'opération [CreateWirelessGateway](#)d'API. Lorsque vousappelez l'opération d'API, spécifiez les paramètres suivants à l'aide de l'Beaconingobjet de vos passerelles. Une fois les paramètres configurés, les passerelles enverront les balises à vos appareils à un intervalle de 128 secondes.

- **DataRate:** débit de données pour les passerelles qui transmettent les balises.
- **Frequencies:** la liste des fréquences utilisées par les passerelles pour transmettre les balises.

L'exemple suivant montre comment configurer ces paramètres pour la passerelle. Le `input.json` fichier contiendra des informations supplémentaires, telles que le certificat de passerelle et les informations d'identification de configuration. Pour plus d'informations sur l'ajout de votre passerelle àAWS IoT Core for LoRaWAN l'utilisation de l'opération d'CreateWirelessGatewayAPI, consultez[Ajouter une passerelle à l'aide de l'API \(p. 1284\)](#).

Note

Les paramètres de balisage ne sont pas disponibles lorsque vous ajoutez votre passerelle àAWS IoT Core for LoRaWAN l'utilisation de laAWS IoT console.

```
aws iotwireless create-wireless-gateway \
--name "myLoRaWANGateway" \
--cli-input-json file://input.json
```

L'exemple suivant affiche le contenu du fichier `input.json`.

Contenu du fichier `input.json`

```
{
  "Description": "My LoRaWAN gateway",
  "LoRaWAN": {
    "Beaconing": {
      "DataRate": 8,
      "Frequencies": ["923300000", "923900000"]
    },
    "GatewayEui": "a1b2c3d4567890ab",
    "RfRegion": "US915",
    "JoinEuiFilters": [
      "0000000000000001", "00000000000000ff"
    ],
    "UpLinkFilters": [
      "0000000000000001", "00000000000000ff"
    ]
  }
}
```

```
[ "000000000000ff00", "000000000000ffff" ]  
],  
"NetIdFilters": [ "000000", "000001" ],  
"RfRegion": "US915",  
"SubBands": [ 2 ]  
}  
}
```

Le code suivant montre un exemple de sortie de l'exécution de cette commande.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:400232685877aa:WirelessGateway/a01b2c34-d44e-567f-abcd-0123e445663a",  
    "Id": "a01b2c34-d44e-567f-abcd-0123e445663a"  
}
```

Obtention d'informations sur les paramètres de balisage

Vous pouvez obtenir des informations sur les paramètres de balisage de votre passerelle à l'aide de l'opération [GetWirelessGateway](#)d'API.

Note

Si une passerelle a déjà été intégrée, vous ne pouvez pas utiliser l'opération d'UpdateWirelessGatewayAPI pour configurer les paramètres de balisage. Pour configurer les paramètres, vous devez supprimer la passerelle, puis spécifier les paramètres lors de l'ajout de votre passerelle à l'aide de l'opérationCreateWirelessGateway d'API.

```
aws iotwireless get-wireless-gateway \  
--identifier "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
--identifier-type WirelessGatewayId
```

L'exécution de cette commande renvoie des informations sur votre passerelle et les paramètres de balisage.

Configuration des sous-bandes et des capacités de filtrage de votre passerelle

LoRaLes passerelles WAN exécutent un logiciel [LoRa Basics Station](#) qui permet aux passerelles de se connecterAWS IoT Core for LoRaWAN. Pour s'y connecterAWS IoT Core for LoRaWAN, votre LoRa passerelle interroge d'abord le serveur CUPS sur le point de terminaison LNS, puis établit une connexion de WebSockets données avec ce point de terminaison. Une fois la connexion établie, les trames montantes et descendantes peuvent être échangées via cette connexion.

Filtrage des trames de LoRa données reçues par la passerelle

Une fois que votre passerelle LoRa WAN a établi une connexion avec le point de terminaison, elleAWS IoT Core for LoRaWAN répond par unrouter_config message qui spécifie un ensemble de paramètres pour la configuration de la LoRa passerelle, y compris les paramètres de filtrageNetID etJoinEui. Pour plus d'informations surrouter_config et la manière dont une connexion est établie avec le serveur réseau LoRa WAN (LNS), consultez le [protocole 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
"npsc"     : BOOL
"nodwell"  : BOOL
}
```

Les passerelles transportent les données des appareils LoRa WAN vers et depuis le LNS, généralement via des réseaux à large bande passante tels que Wi-Fi, Ethernet ou Cellular. Les passerelles captent généralement tous les messages et transmettent le trafic qui leur parvient AWS IoT for LoRaWAN. Toutefois, vous pouvez configurer les passerelles pour filtrer une partie du trafic de données de l'appareil, ce qui permet d'économiser de la bande passante et de réduire le flux de trafic entre la passerelle et le LNS.

Pour configurer votre LoRa passerelle afin de filtrer les blocs de données, vous pouvez utiliser les paramètres `NetID` et `JoinEui` le `router_config` message. `NetID` est une liste de valeurs `NetID` acceptées. Toute trame de LoRa données contenant une trame de données autre que celles répertoriées sera supprimée. `JoinEui` est une liste de paires de valeurs entières codant des plages de valeurs `JoinEui`. Les trames de demande de jointure seront supprimées par la passerelle `JoinEui` à moins que le champ du message ne se trouve dans la plage `[BegEui,EndEui]`.

Canaux et sous-bandes de fréquences

Pour les régions RF US915 et AU915, les appareils sans fil ont le choix entre 64 canaux de liaison montante à 125 kHz et 8 à 500 kHz pour accéder aux réseaux LoRa WAN via les LoRa passerelles. Les canaux de fréquence de liaison montante sont divisés en 8 sous-bandes, chacune avec 8 canaux de 125 kHz et un canal de 500 kHz. Pour chaque passerelle standard de la région AU915, une ou plusieurs sous-bandes seront prises en charge.

Certains appareils sans fil ne peuvent pas passer d'une sous-bande à l'autre et n'utilisent les canaux de fréquence que dans une seule sous-bande lorsqu'ils y sont connectés AWS IoT for LoRaWAN. Pour que les paquets de liaison montante provenant de ces appareils soient transmis, configurez les LoRa passerelles pour utiliser cette sous-bande particulière. Pour les passerelles situées dans d'autres régions RF, telles que l'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 une sous-bande particulière et également activer la fonctionnalité de filtrage des trames de LoRa données. Pour spécifier ces paramètres à l'aide de la console, procédez comme suit :

1. Accédez à la page [AWS IoT Core for LoRaWAN](#) Passerelles de la AWS IoT console et choisissez Ajouter une passerelle.
2. Spécifiez les détails de la passerelle, tels que l'interface utilisateur, la bande de fréquences (RFRegion), un nom et une description facultatifs, et choisissez d'associer ou non un AWS IoT élément à votre passerelle. Pour plus d'informations sur l'ajout d'une passerelle, consultez [Ajouter une passerelle à l'aide de la console \(p. 1282\)](#).
3. Dans la section de configuration LoRa WAN, vous pouvez spécifier les sous-bandes et les informations de filtrage.
 - `SubBands`: pour ajouter un sous-canal, choisissez Ajouter SubBand et spécifiez une liste de valeurs entières indiquant les sous-bandes prises en charge par la passerelle. Le `SubBands` paramètre ne peut être configuré que dans les RFRegion modèles US915 et AU915 et doit avoir des valeurs comprises dans l'une de ces régions prises en charge. [1, 8]
 - `NetIdFilters`: pour filtrer les trames de liaison montante, choisissez Ajouter NetId et spécifiez une liste de valeurs de chaîne utilisées par la passerelle. Le NetID de la trame de liaison montante entrante en provenance du périphérique sans fil doit correspondre à au moins l'une des valeurs répertoriées, sinon la trame est supprimée.

- JoinEuiFilters: choisissez Ajouter une JoinEui plage et spécifiez une liste de paires de valeurs de chaîne qu'une passerelle utilise pour filtrer les LoRa trames. La valeur JoinEui spécifiée dans le cadre de la demande de jointure provenant du périphérique sans fil doit se situer dans la plage d'au moins l'une des JoinEuiRange valeurs, chacune répertoriée sous la forme d'une paire de [BegEui, EndEui], sinon l'image est supprimée.
4. Vous pouvez ensuite continuer à configurer votre passerelle en suivant les instructions décrites dans [Ajouter une passerelle à l'aide de la console \(p. 1282\)](#).

Après avoir ajouté une passerelle, sur la page [AWS IoT Core for LoRaWAN](#) Passerelles de la AWS IoT console, si vous sélectionnez la passerelle que vous avez ajoutée, vous pouvez voir les filtersSubBandsNetIdFilters et JoinEuiFilters dans la section Détails spécifiques au LoRa WAN de la page de détails de la passerelle.

Configurez votre passerelle pour utiliser le filtrage et les sous-bandes à l'aide de l'API

Vous pouvez utiliser l'[CreateWirelessGateway](#) API que vous utilisez pour créer une passerelle afin de configurer les sous-bandes que vous souhaitez utiliser et d'activer la fonctionnalité de filtrage. À l'aide de l'[CreateWirelessGateway](#) API, vous pouvez spécifier les sous-bandes et les filtres dans le cadre des informations de configuration de la passerelle que vous fournissez à l'aide du LoRaWAN champ. Ce qui suit montre 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 l'[UpdateWirelessGateway](#) API pour mettre à jour les filtres, mais pas les sous-bandes. Si les NetIdFilters et les JoinEuiFilters 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 l'[GetWirelessGateway](#) API.

Mettez à jour le firmware de la passerelle à l'aide du service CUPSAWS IoT Core for LoRaWAN

Le logiciel [LoRa Basics Station](#) qui s'exécute sur votre passerelle fournit une interface de gestion des informations d'identification et 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 micrologiciel de votre passerelle. Vous pouvez utiliser le service CUPSAWS IoT Core for LoRaWAN pour fournir des mises à jour du microprogramme à la passerelle où les mises à jour peuvent également être signées. Pour mettre à jour le microprogramme 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 prend environ 45 minutes. Cela peut prendre plus de temps si vous configurez votre passerelle pour vous y connecter pour la première fois AWS IoT Core for LoRaWAN. Les fabricants de passerelles fournissent généralement leurs propres fichiers de mise à jour du microprogramme et leurs propres signatures. Vous pouvez donc les utiliser à la place et passer à l'étape suivante [Téléchargement du fichier du microprogramme dans un compartiment S3 et ajout d'un rôle IAM \(p. 1328\)](#).

Si vous ne disposez pas des fichiers de mise à jour du microprogramme, consultez un [Générez le fichier de mise à jour du microprogramme et la signature \(p. 1325\)](#) exemple que vous pouvez utiliser pour vous adapter à votre application.

Pour effectuer la mise à jour du microprogramme de votre passerelle, procédez comme suit :

- [Générez le fichier de mise à jour du microprogramme et la signature \(p. 1325\)](#)
- [Téléchargement du fichier du microprogramme dans un compartiment S3 et ajout d'un rôle IAM \(p. 1328\)](#)
- [Planifiez et exécutez la mise à jour du microprogramme à l'aide d'une définition de tâche \(p. 1331\)](#)

Générez le fichier de mise à jour du microprogramme et la signature

Les étapes de cette procédure sont facultatives et dépendent de la passerelle que vous utilisez. Les fabricants de passerelles fournissent leur propre mise à jour du microprogramme 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 microprogramme 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 à [Téléchargement du fichier du microprogramme dans un compartiment S3 et ajout d'un rôle IAM \(p. 1328\)](#).

Si vous ne disposez pas de ce script, vous trouverez ci-dessous les commandes à exécuter pour générer le fichier de mise à jour du microprogramme. Les mises à jour peuvent également être signées pour garantir que le code n'a pas été modifié ou corrompu et que les appareils exécutent du code publié uniquement par des auteurs fiables.

Au cours de cette procédure, vous allez :

- [Générez le fichier de mise à jour du microprogramme \(p. 1325\)](#)
- [Générer une signature pour la mise à jour du microprogramme \(p. 1327\)](#)
- [Passez en revue les Étapes suivantes \(p. 1328\)](#)

Générez le fichier de mise à jour du microprogramme

Le logiciel LoRa Basics Station exécuté sur la passerelle est capable de recevoir des mises à jour du microprogramme dans la réponse CUPS. Si aucun script n'a été fourni par le fabricant, reportez-vous au script de mise à jour du microprogramme suivant, écrit pour la passerelle RAKWireless Gateway basée sur Raspberry Pi. Nous avons un script de base et le fichier de version du nouveau binaire de la station, et nous ystation.conf sommes joints.

Note

Le script est spécifique à la passerelle RAK Wireless, 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 la passerelle sans fil RAK basée sur Raspberry Pi. Vous pouvez enregistrer les commandes suivantes dans un fichier, `base.sh`, puis exécuter le script dans le terminal du navigateur Web du 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_version
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 fichier version.txt qui identifie la version vers laquelle effectuer la mise 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 microprogramme, fwstation.

```
$ ./addpayload.sh station version.txt station.conf
```

Générer une signature pour la mise à jour du microprogramme

Le logiciel LoRa Basics Station fournit des mises à jour du microprogramme signées avec des signatures ECDSA. Pour prendre en charge les mises à jour signées, il vous faut :

- Signature qui doit être générée par une clé privée ECDSA de moins de 128 octets.
- La clé privée qui est 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 du fichiersig-0.key.
- Un CRC 32 bits sur la clé privée.

La signature et le CRC seront transmis aux AWS IoT Core for LoRaWAN API. Pour générer les fichiers précédents, vous pouvez utiliser le script suivant gen.sh qui s'inspire de l'exemple [basicstation](#) du GitHub référentiel.

```
*#!/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/p2ssgXIPNC0gZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBR0en+XlIdMScv
AsfVfU/ZScJCa1kVNZh4esyS8mNIgA==

$ 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 microprogramme et la signature, passez à la rubrique suivante pour télécharger le fichier du microprogramme fwstation,, dans un compartiment Amazon S3. Le bucket est un conteneur qui stockera le fichier de mise à jour du microprogramme en tant qu'objet. Vous pouvez ajouter un rôle IAM qui autorisera le serveur CUPS à lire le fichier de mise à jour du microprogramme dans le compartiment S3.

Téléchargement du fichier du microprogramme dans un compartiment S3 et ajout d'un rôle IAM

Vous pouvez utiliser Amazon S3 pour créer un compartiment, qui est un conteneur qui peut stocker votre fichier de mise à jour du microprogramme. 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 depuis le

compartiment. Pour de plus informations sur Amazon S3, veuillez consulter la section [Mise en route avec Amazon S3](#).

Le fichier de mise à jour du microprogramme que vous souhaitez télécharger dépend de la passerelle que vous utilisez. Si vous avez suivi une procédure similaire à celle décrite dans[Générez le fichier de mise à jour du microprogramme et la signature \(p. 1325\)](#), vous téléchargerez le fwstation fichier généré en exécutant les scripts.

Cette procédure prend environ 20 minutes.

Pour télécharger le fichier de votre microprogramme, procédez comme suit :

- [Création d'un compartiment Amazon S3 et chargement du fichier de mise à jour \(p. 1329\)](#)
- [Création d'un rôle IAM autorisé à lire le compartiment S3 \(p. 1330\)](#)
- [Passez en revue les Étapes suivantes \(p. 1331\)](#)

Création d'un compartiment Amazon S3 et chargement du fichier de mise à jour

Vous allez créer un compartiment Amazon S3 à l'aide du AWS Management Console puis charger votre fichier de mise à jour du microprogramme dans le compartiment.

Création d'un compartiment S3

Pour créer un compartiment S3, ouvrez la [console Amazon S3](#). Connectez-vous si vous ne l'avez pas déjà fait, puis effectuez les étapes suivantes :

1. Choisissez Create bucket (Créer un compartiment).
2. Entrez un nom unique et significatif pour le nom du compartiment (par exemple, iotwirelessfwupdate). Pour connaître la convention de dénomination recommandée pour votre compartiment, consultez<https://docs.aws.amazon.com/AmazonS3/latest/userguide/bucketnamingrules.html>.
3. Assurez-vous que vous avez Région AWS sélectionné celui que vous avez utilisé pour créer votre passerelle et votre appareil LoRa WAN, et que le paramètre Bloquer tout accès public est sélectionné afin que votre compartiment utilise les autorisations par défaut.
4. Choisissez Activer pour le contrôle des versions du compartiment, ce qui vous aidera à conserver plusieurs versions du fichier de mise à jour du microprogramme dans le même compartiment.
5. Vérifiez que le chiffrement côté serveur est défini sur Désactiver et choisissez Créer un compartiment.

Téléchargez votre fichier de mise à jour du micrologiciel

Vous pouvez désormais voir votre compartiment dans la liste des compartiments affichée dans le AWS Management Console. Choisissez votre compartiment et procédez comme suit pour charger votre fichier.

1. Choisissez votre compartiment, puis choisissez Charger.
2. Choisissez Ajouter un fichier, puis chargez le fichier de mise à jour du microprogramme. Si vous avez suivi la procédure décrite dans[Générez le fichier de mise à jour du microprogramme et la signature \(p. 1325\)](#), vous téléchargerez le fwstation fichier, sinon vous téléchargerez le fichier fourni par le fabricant de votre passerelle.
3. Assurez-vous que tous les paramètres sont réglés sur leurs valeurs par défaut. Assurez-vous que les ACL prédefinies sont définies sur privées et choisissez Charger pour télécharger votre fichier.
4. Copiez l'URI S3 du fichier que vous avez chargé. Choisissez votre compartiment et le fichier que vous avez chargé s'affichera dans la liste des objets. Choisissez votre fichier, puis choisissez Copier l'URI S3. L'URI sera quelque chose comme :s3://iotwirelessfwupdate/fwstation si vous avez nommé votre compartiment de la même manière que dans l'exemple décrit précédemment (fwstation). Vous utiliserez l'URI Amazon S3 lors de la création du rôle IAM.

Création d'un rôle IAM autorisé à lire le compartiment S3

Vous allez maintenant créer un rôle et une politique IAM qui accorderont à CUPS l'autorisation de lire votre fichier de mise à jour du microprogramme depuis le compartiment S3.

Créez une politique IAM pour votre rôle

Pour créer une politique IAM pour votre rôle de AWS IoT Core for LoRaWAN destination, ouvrez le [hub Policies de la console IAM](#), puis procédez comme suit :

1. Choisissez Créer une politique, puis choisissez l'onglet JSON.
2. Supprimez tout contenu de l'éditeur et collez ce document de politique. La politique fournit des autorisations pour accéder au `iotwireless` compartiment et au fichier de mise à jour du microprogramme `fwstation`, stockés dans un objet.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "s3>ListBucketVersions",  
                "s3>ListBucket",  
                "s3GetObject"  
            ],  
            "Resource": [  
                "arn:aws:s3:::iotwirelessfwupdate/fwstation",  
                "arn:aws:s3:::iotwirelessfwupdate"  
            ]  
        }  
    ]  
}
```

3. Choisissez Réviser la politique et, dans Nom, entrez le nom de cette politique (par exemple, `IoTWirelessFwUpdatePolicy`). Vous aurez besoin de ce nom pour la procédure suivante sera le suivant.
4. Choisissez Create Policy (Créer une politique).

Création d'un rôle IAM avec la politique jointe

Vous allez maintenant créer un rôle IAM et attacher la stratégie créée précédemment pour l'accès au compartiment S3. Ouvrez le [hub Roles de la console IAM](#) et procédez comme suit :

1. Sélectionnez Create role (Créer un rôle).
2. Dans Sélectionner le type d'entité sécurisée, choisissez AutreCompte AWS.
3. Dans Identifiant du compte, saisissez votreCompte AWS identifiant, puis choisissez Suivant : Autorisations.
4. Dans la zone de recherche, tapez le nom de la stratégie IAM que vous avez créée au cours de la procédure précédente, tapez le nom de la stratégie IAM que vous avez créée au cours de la procédure précédente Vérifiez la politique IAM (par exemple `IoTWirelessFwUpdatePolicy`) que vous avez créée précédemment dans les résultats de recherche et choisissez-la.
5. Sélectionnez Next: Tags (Suivant : Balises), puis Next: Review (Suivant : Vérification).
6. Dans Nom du rôle, entrez le nom de ce rôle (par exemple, `IoTWirelessFwUpdateRole`), puis choisissez Créez un rôle.

Modifier la relation de confiance du rôle IAM

Dans le message de confirmation qui s'affiche après avoir exécuté l'étape précédente, choisissez le nom du rôle que vous avez créé pour le modifier. Vous allez modifier le rôle pour ajouter la relation de confiance suivante.

1. Dans la section Résumé du rôle que vous avez créé, choisissez l'onglet Relations de confiance, puis choisissez Modifier la relation de confiance.
2. Dans le document de politique, modifiez laPrincipal propriété pour qu'elle ressemble à cet exemple.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Une fois laPrincipal propriété modifiée, le document de politique complet doit ressembler à 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, choisissez Mettre à jour la politique de confiance.
4. Obtenez l'ARN correspondant à votre rôle. Choisissez votre rôle IAM et dans la section Résumé, vous verrez un ARN de rôle, tel quearn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole. Copiez ce rôle ARN.

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. Conservez l'URI S3 et l'ARN de rôle que vous avez copiés précédemment afin de pouvoir les saisir pour créer une définition de tâche qui sera exécutée pour effectuer la mise à jour du microprogramme.

Planifiez et exécutez 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 for LoRaWAN fournit une mise à jour du microprogramme basée sur les informations provenant des trois champs suivants associés à la passerelle.

- Gare

Version et heure de construction du logiciel Basics Station. Pour identifier ces informations, vous pouvez également les générer à l'aide du logiciel Basics Station qui est exécuté par votre passerelle (par exemple, 2.0.5(rpi/std) 2021-03-09 03:45:09).

- PackageVersion

Version du microprogramme, 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 utiliser pour définir la version de votre microprogramme (par exemple, `1.0.0`).

- **Modèle**

Plateforme ou modèle utilisé par la passerelle (par exemple, Linux).

Cette procédure prend 20 minutes.

Pour réaliser cette procédure :

- [Exécutez la version actuelle sur votre passerelle \(p. 1332\)](#)
- [Création d'une définition de tâche de passerelle sans fil \(p. 1333\)](#)
- [Exécutez la tâche de mise à jour du micrologiciel et suivez la progression \(p. 1334\)](#)

Exécutez la version actuelle sur votre passerelle

Pour déterminer l'éligibilité de votre passerelle à une mise à jour du microprogramme, le serveur CUPS vérifie la correspondance des trois champs `StationPackageVersionModel`, et, lorsque la passerelle les présente lors d'une demande CUPS. Lorsque vous utilisez une définition de tâche, ces champs sont stockés dans le `CurrentVersion` champ.

Vous pouvez utiliser l'AWS IoT Core for LoRaWAN API ou l'AWS CLI `getWirelessGatewayCurrentVersion` pour votre passerelle. Les commandes suivantes montrent comment obtenir ces informations à l'aide de l'interface de ligne de commande.

1. Si vous avez déjà configuré une passerelle, vous pouvez obtenir des informations sur la passerelle à l'aide de la [get-wireless-gateway](#) commande.

```
aws iotwireless get-wireless-gateway \
--identifier 5a11b0a85a11b0a8 \
--identifier-type GatewayEui
```

L'exemple suivant affiche un exemple de sortie pour la commande est le suivant.

```
{  
    "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 indiqué par la `get-wireless-gateway` commande, vous pouvez utiliser la commande [get-wireless-gateway-firmware-information](#) pour obtenir le `CurrentVersion`.

```
aws iotwireless get-wireless-gateway-firmware-information \
--id "3039b406-5cc9-4307-925b-9948c63da25b"
```

Vous trouverez ci-dessous un exemple de sortie pour la commande, avec des informations provenant des trois champs affichés par `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 `AutoCreateTasks` paramètre. `AutoCreateTaskss` s'applique à toute passerelle qui correspond aux trois paramètres mentionnés précédemment. Si ce paramètre est désactivé, les paramètres doivent être assignés manuellement à la passerelle.

Vous pouvez créer la définition de tâche de passerelle sans fil à l'aide de l'AWS IoT Core for LoRaWAN API ou AWS CLI. Les commandes suivantes montrent comment créer la définition de tâche à l'aide de l'interface de ligne de commande.

1. Créez un fichier `input.json` contenant `input.json` les informations à transmettre à l'`CreateWirelessGatewayTaskDefinitionAPI`. Dans le `input.json` fichier, fournissez 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 le rôle ARN du rôle IAM que vous avez créé, qui fournit les autorisations nécessaires 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 dans [Générer le fichier de mise à jour du microprogramme et la signature \(p. 1325\)](#), vous trouverez ces informations lors de la génération de la signature.

- `CurrentVersion`

Fournissez la `CurrentVersion` sortie que vous avez obtenue précédemment en exécutant `aget-wireless-gateway-firmware-information` commande.

```
cat input.json
```

L'exemple suivant affiche le contenu du `input.json` fichier est le suivant.

```
{  
    "AutoCreateTasks": true,  
    "Name": "FirmwareUpdate",  
    "Update": {  
        "UpdateDataSource" : "s3://iotwirelessfwupdate/fwstation",  
        "UpdateDataRole" : "arn:aws:iam::123456789012:role/IoTWirelessFwUpdateRole",  
        "LoRaWAN" : {  
            "CurrentVersion": {  
                "PackageVersion": "1.0.0",  
                "Model": "rpi",  
                "Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"  
            }  
        }  
    }  
}
```

```
"SigKeyCrc": 3434210794,  
"UpdateSignature": "MEQCIDPY/p2ssgXIPNC0gZr+NzeTLpX+WfBo5tYWbh5pQWN3AiBR0en  
+XlIdMScvAsfVfU/ZScJCalhVNZh4esyS8mNIgA==",  
"CurrentVersion" :  
{  
"PackageVersion": "1.0.0",  
"Model": "rpi",  
"Station": "2.0.5(rpi/std) 2021-03-09 03:45:09"  
}  
}  
}
```

- Transmettez le `input.json` fichier à la commande [create-wireless-gateway-task-definition](#) pour créer la définition de la tâche.

```
aws iotwireless create-wireless-gateway-task-definition \  
--cli-input-json file://input.json
```

Le résultat de la commande est présenté ci-dessous.

```
{  
"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 microprogramme et, une fois allumée, 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 de traitement. 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 la progression de la tâche à l'aide de l'`GetWirelessGatewayTaskAPI`. Lorsque vous exécutez la [get-wireless-gateway-task](#) commande pour la première fois, elle affiche l'état de la tâche sous la forme `IN_PROGRESS`.

```
aws iotwireless get-wireless-gateway-task \  
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

Le résultat de la commande est présenté ci-dessous.

```
{  
"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 exécuterez la commande la prochaine fois, si la mise à jour du microprogramme prend effet, elle affichera les champs `mis à jour` `PackageVersion`, `Model` et le statut de la tâche passe à `COMPLETED`.

```
aws iotwireless get-wireless-gateway-task \
```

```
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

Le résultat de la commande est présenté ci-dessous.

```
{  
    "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 micrologiciel à l'aide de la passerelle RAK Wireless basée sur le Raspberry Pi. Le script de mise à jour du microprogramme arrête l'exécution BasicStation pour stocker les mises à jourPackageVersion, etModel les champs BasicStation devront donc être redémarrés.

```
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, le serveur CUPS affiche l'état et la passerelle envoie la même demande.FIRST_RETRY Si le serveur CUPS ne parvient pas à se connecter à la passerelle après unSECOND_RETRY, il affichera un état deFAILED.

Une fois la tâche précédente terminéeCOMPLETED ouFAILED, supprimez l'ancienne tâche à l'aide de la [delete-wireless-gateway-task](#)commande avant d'en démarrer une nouvelle.

```
aws iotwireless delete-wireless-gateway-task \  
--id 1352172b-0602-4b40-896f-54da9ed16b57
```

Choix des passerelles pour recevoir le trafic de données de liaison descendante du LoRa WAN

Lorsque vous envoyez un message de liaison descendante depuisAWS IoT Core for LoRaWAN votre appareil, vous pouvez choisir les passerelles que vous souhaitez utiliser pour le trafic de données en liaison descendante. Vous pouvez spécifier une passerelle individuelle ou choisir parmi une liste de passerelles pour recevoir le trafic de liaison descendante.

Comment spécifier la liste des passerelles

Vous pouvez spécifier une passerelle individuelle ou la liste des passerelles à utiliser lors de l'envoi d'un message de liaison descendante depuis votre appareilAWS IoT Core for LoRaWAN à l'aide de l'opération [SendDataToWirelessDevice](#)API. Lorsque vous appelez l'opération d'API, spécifiez les paramètres suivants à l'aide de l'ParticipatingGatewaysobjet de vos passerelles.

Note

La liste des passerelles que vous souhaitez utiliser n'est pas disponible dans laAWS IoT console. Vous pouvez spécifier cette liste de passerelles à utiliser uniquement lors de l'utilisation de l'opération d'[SendDataToWirelessDevice](#)API ou de l'interface de ligne de commande.

- **DownlinkMode:** indique s'il faut envoyer le message de liaison descendante en mode séquentiel ou en mode simultané. Pour les appareils de classe A, spécifiezUsingUplinkGateway de n'utiliser que les passerelles choisies lors de la transmission de messages par liaison montante précédente.
- **GatewayList:** liste des passerelles que vous souhaitez utiliser pour envoyer le trafic de données en liaison descendante. La charge utile en liaison descendante sera envoyée aux passerelles spécifiées à la fréquence spécifiée. Ceci est indiqué à l'aide d'une liste d'GatewayListItemobjets, composée deGatewayId et deDownlinkFrequency paires.
- **TransmissionInterval:** durée pendant laquelle ilAWS IoT Core for LoRaWAN faudra attendre avant de transmettre la charge utile à la passerelle suivante.

Note

Vous pouvez spécifier cette liste de passerelles à utiliser uniquement lors de l'envoi du message de liaison descendante à un périphérique sans fil de classe B ou de classe C. Si vous utilisez un appareil de classe A, la passerelle que vous avez choisie lors de l'envoi du message de liaison montante sera utilisée lorsqu'un message de liaison descendante est envoyé à l'appareil.

L'exemple suivant montre comment spécifier ces paramètres pour la passerelle. Leinput.json fichier contiendra des informations supplémentaires. Pour plus d'informations sur l'envoi d'un message de liaison descendante à l'aide de l'opération dSendDataToWirelessDeviceAPI, consultez[Effectuez des opérations de file d'attente de liaison descendante à l'aide de l'API \(p. 1343\)](#).

Note

Les paramètres permettant de spécifier la liste des passerelles participantes ne sont pas disponibles lorsque vous envoyez un message de liaison descendante àAWS IoT Core for LoRaWAN l'aide de laAWS IoT console.

```
aws iotwireless send-data-to-wireless-device \
--id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
--transmit-mode "1" \
--payload-data "SGVsbG8gVG8gRGV2c2lt" \
--cli-input-json file://input.json
```

L'exemple suivant affiche le contenu du fichier input.json.

Contenu du fichier input.json

```
{
    "WirelessMetadata": {
        "LoRaWAN": {
            "FPort": "1",
            "ParticipatingGateways": {
                "DownlinkMode": "SEQUENTIAL",
                "TransmissionInterval": 1200,
                "GatewayList": [
                    {
                        "DownlinkFrequency": 100000000,
                        "GatewayID": a01b2c34-d44e-567f-abcd-0123e445663a
                    },
                    {
                        "DownlinkFrequency": 100000101,
                        "GatewayID": 12345678-a1b2-3c45-67d8-e90fa1b2c34d
                    }
                ]
            }
        }
    }
}
```

Le résultat de l'exécution de cette commande génère un messageMessageId pour le lien descendant. Dans certains cas, même si vous recevez leMessageId, des paquets peuvent être perdus. Pour plus d'informations sur la résolution de l'erreur, consultez[Résoudre les erreurs liées à la file de messages de liaison descendante \(p. 1344\)](#).

```
{  
    messageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Obtenir des informations sur la liste des passerelles participantes

Vous pouvez obtenir des informations sur la liste des passerelles qui participent à la réception du message de liaison descendante en répertoriant les messages dans la file d'attente de liaison descendante. Pour répertorier les messages, utilisez l'[ListQueuedMessages API](#).

```
aws iotwireless list-queued-messages \  
    --wireless-device-type "LoRaWAN"
```

L'exécution de cette commande renvoie des informations sur les messages de la file d'attente et leurs paramètres.

Gestion des appareils avec AWS IoT Core for LoRaWAN

Les appareils WAN communiquent avec euxAWS IoT Core for LoRaWAN via des passerelles LoRa WAN. L'ajout d'appareilsAWS IoT Core for LoRaWAN permet deAWS IoT traiter les messages reçus des appareils à des fins d'utilisation parAWS IoT d'autres services.

Voici quelques points importants à prendre en compte lors de l'utilisation de vos appareils avecAWS IoT Core for LoRaWAN. Pour en savoir plus sur l'ajout de votre appareil àAWS IoT Core for LoRaWAN, veuillez consulter[Intégrez vos appareils àAWS IoT Core for LoRaWAN \(p. 1288\)](#).

Considérations sur les appareils

Lorsque vous sélectionnez un appareil avec lequel vous souhaitez communiquerAWS IoT Core for LoRaWAN, tenez compte des points suivants.

- Capteurs disponibles
- Capacité de la batterie
- Consommation d'énergie
- Coût
- Type d'antenne et portée de transmission

Utilisation d'appareils dotés de passerelles qualifiées pourAWS IoT Core for LoRaWAN

Les appareils que vous utilisez peuvent être couplés à des passerelles sans fil qualifiées pour une utilisation avecAWS IoT Core for LoRaWAN. Vous trouverez ces passerelles et kits de développement dans le [catalogue des appareilsAWS partenaires](#). Nous vous recommandons également de prendre

en compte la proximité de ces appareils par rapport à vos passerelles. Pour plus d'informations, veuillez consulter [Utilisation de passerelles qualifiées figurant dans le catalogue d'appareils AWS partenaires \(p. 1319\)](#).

LoRaVersion WAN

AWS IoT Core for LoRaWAN prend en charge tous les appareils conformes aux spécifications LoRa WAN 1.0.x ou 1.1 normalisées par LoRa Alliance.

Modes d'activation

Avant que votre appareil LoRa WAN puisse envoyer des données de liaison montante, vous devez effectuer un processus appelé procédure d'activation ou de connexion. Pour activer votre terminal, vous pouvez utiliser l'OTAA (activation sans fil) ou l'ABP (activation par personnalisation). Nous vous recommandons d'utiliser OTAA pour activer votre appareil, car de nouvelles clés de session sont générées à chaque activation, ce qui le rend plus sécurisé.

Les spécifications de votre appareil sans fil sont basées sur la version LoRa WAN et le mode d'activation, qui déterminent les clés racine et les clés de session générées pour chaque activation. Pour plus d'informations, veuillez consulter [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#).

Classes d'appareil

Les appareils WAN peuvent envoyer des messages de liaison montante à tout moment. L'écoute de messages en liaison descendante consomme la capacité de la batterie et réduit sa durée de vie. Le protocole LoRa WAN spécifie trois classes de périphériques LoRa WAN.

- 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 dont la durée de vie peut atteindre 10 ans.
- Les appareils de classe B peuvent recevoir des messages dans des emplacements de liaison descendante programmés. Ces appareils sont pour la plupart des actionneurs alimentés par batterie.
- Les appareils de classe C ne dorment jamais et écoutent en permanence les messages entrants, ce qui permet de les recevoir rapidement. Ces appareils sont pour la plupart des actionneurs alimentés sur secteur.

Pour plus d'informations sur ces considérations relatives aux périphériques sans fil, consultez les ressources mentionnées dans [En savoir plus sur le LoRa réseau WAN \(p. 1275\)](#).

Gérez la communication entre vos appareils LoRa WAN et AWS IoT

Une fois que vous avez connecté votre appareil LoRa WAN à AWS IoT Core for LoRaWAN, vos appareils peuvent commencer à envoyer des messages vers le cloud. Les messages Uplink sont des messages envoyés depuis votre appareil et reçus par AWS IoT Core for LoRaWAN. Vos appareils LoRa WAN peuvent envoyer des messages de liaison montante à tout moment, qui sont ensuite transférés vers d'autres applications AWS hébergées dans le cloud. Les messages envoyés depuis AWS IoT Core for LoRaWAN et d'autres AWS services vers vos appareils sont appelés messages de liaison descendante.

Ce qui suit montre comment afficher et gérer les messages de liaison montante et descendante qui sont envoyés entre vos appareils et le Cloud. Vous pouvez gérer une file de messages en liaison descendante et envoyer ces messages à vos appareils dans l'ordre dans lequel ils ont été ajoutés à la file d'attente.

Rubriques

- [Afficher le format des messages de liaison montante envoyés depuis des périphériques LoRa WAN \(p. 1339\)](#)
- [Mettre en file d'attente les messages de liaison descendante à envoyer aux périphériques LoRa WAN \(p. 1341\)](#)

Afficher le format des messages de liaison montante envoyés depuis des périphériques LoRa WAN

Après avoir connecté votre périphérique LoRa WAN à AWS IoT Core for LoRaWAN, vous pouvez observer le format du message de liaison montante que vous recevez de votre appareil sans fil.

Avant de pouvoir observer les messages de liaison montante

Vous devez avoir intégré votre appareil sans fil et l'avoir connecté pour AWS IoT qu'il puisse transmettre et recevoir des données. Pour plus d'informations sur l'intégration de votre appareil à AWS IoT Core for LoRaWAN, consultez [Intégrez vos appareils à AWS IoT Core for LoRaWAN \(p. 1288\)](#).

Que contiennent les messages de liaison montante ?

Les périphériques WAN se connectent à AWS IoT Core for LoRaWAN à l'aide de passerelles LoRa WAN. Le message de liaison montante que vous recevez de l'appareil contiendra les informations suivantes.

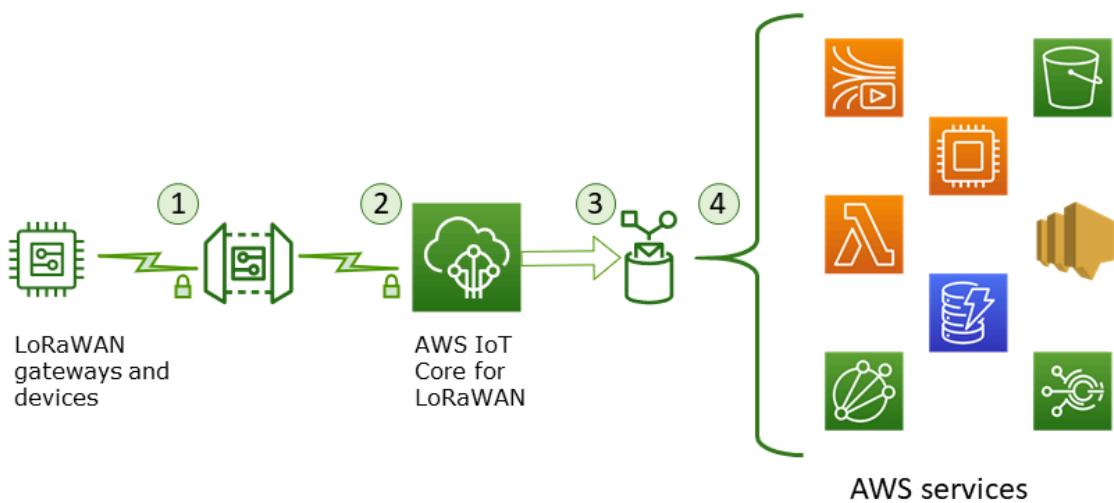
- Données de charge utile correspondant au message de charge utile crypté envoyé par le dispositif sans fil.
- Des métadonnées sans fil qui incluent :
 - Informations sur l'appareil DevEui, telles que le débit de données et le canal de fréquence dans lequel l'appareil fonctionne.
 - Paramètres supplémentaires facultatifs et informations de passerelle pour les passerelles connectées à l'appareil. Les paramètres de passerelle incluent l'EUI, le SNR et le RSSI de la passerelle.

En utilisant les métadonnées sans fil, vous pouvez obtenir des informations utiles sur le périphérique sans fil et les données transmises entre votre appareil et AWS IoT. Par exemple, vous pouvez utiliser leAckedMessageId paramètre pour vérifier si le dernier message de liaison descendante confirmé a été reçu par l'appareil. En option, si vous choisissez d'inclure les informations de passerelle, vous pouvez déterminer si vous souhaitez passer à un canal de passerelle plus puissant et plus proche de votre appareil.

Comment observer les messages de liaison montante ?

Après avoir intégré votre appareil, vous pouvez utiliser le [client de test MQTT](#) sur la page Test de la AWS IoT console pour vous abonner à la rubrique que vous avez spécifiée lors de la création de votre destination. Vous commencerez à recevoir des messages une fois que votre appareil sera connecté et aura commencé à envoyer des données utiles.

Ce diagramme identifie les éléments clés d'un système LoRa WAN auquel il est connecté AWS IoT Core for LoRaWAN. Il montre le plan de données principal et la manière dont les données circulent dans le système.



Lorsque le périphérique sans fil commence à envoyer des données de liaison AWS IoT Core for LoRaWAN montante, il intègre les informations de métadonnées sans fil à la charge utile, puis les envoie à vos AWS applications.

Exemple de message Uplink

L'exemple suivant illustre le format du message de liaison montante reçu à partir de votre appareil.

Note

Si vos appareils envoient un message de liaison montante sans valeur pour Fport, AWS IoT Core for LoRaWAN ajoutera la valeur 225Fport au message de liaison montante reçu.

```
{
    "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,
            "Rssi": -128
        }
    }
}
```

```

        "SpreadingFactor": 12,
        "Timestamp": "2021-04-29T04:19:43Z"
    }
}
}
```

Le tableau suivant présente une description des champs utilisés dans les métadonnées de liaison montante :

LoRaChamps de message WAN Uplink

Paramètre	Description	Type	Obligatoire
WirelessDeviceID	ID du périphérique sans fil qui envoie les données.	Chaîne	Oui
PayloadData	Le message binaire reçu de l'appareil, codé en base64.	Chaîne	Oui
WirelessMetadata	Métadonnées relatives au périphérique LoRa WAN et à la demande de message. Cela inclut des informations telles que les identifiants de l'appareil, le débit de données et de code, l'horodatage du message, si l'ADR (débit de données adaptatif) est activé et les métadonnées de la passerelle.	Énumération	Non

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 le AddGwMetadata paramètre lorsque vous créez le profil de service. Pour de plus amples informations sur ce paramètre, veuillez consulter [Ajouter des profils de service \(p. 1292\)](#).

Dans ce cas, vous ne verrez pas la Gateways section dans les métadonnées de liaison montante, comme illustré dans l'exemple suivant.

```
{
    "WirelessDeviceId": "0d9a439b-e77a-4573-a791-49d5c0f4db95",
    "PayloadData": "AAAAAAA//8=",
    "WirelessMetadata": {
        "LoRaWAN": {
            "ClassB": false,
            "CodeRate": "4/5",
            "DataRate": "1",
            "DevAddr": "01920f27",
            "DevEui": "fffffff10000163b0",
            "FCnt": 1,
            "FPort": 5,
            "Timestamp": "2021-04-29T04:19:43Z"
        }
    }
}
```

Mettre en file d'attente les messages de liaison descendante à envoyer aux périphériques LoRa WAN

Les applications hébergées dans le cloud et autres services AWS peuvent envoyer des messages en liaison descendante à vos appareils sans fil. Les messages de liaison descendante sont des messages AWS IoT

Core for LoRaWAN envoyés depuis votre appareil sans fil. Vous pouvez planifier et envoyer des messages en liaison descendante pour chaque appareil auquel vous êtes connecté AWS IoT Core for LoRaWAN.

Si vous avez plusieurs appareils pour lesquels vous souhaitez envoyer un message en liaison descendante, vous pouvez utiliser un groupe de multidiffusion. Les appareils d'un groupe de multidiffusion partagent la même adresse de multidiffusion, qui est ensuite distribuée à un groupe complet d'appareils destinataires. Pour plus d'informations, veuillez consulter [Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils \(p. 1352\)](#).

Fonctionnement d'une file de messages à liaison descendante

La classe d'appareil de votre périphérique LoRa WAN détermine la manière dont les messages de votre file d'attente sont envoyés à l'appareil. Les appareils de classe A envoient un message de liaison montante AWS IoT Core for LoRaWAN à pour indiquer que le périphérique est disponible pour recevoir des messages de liaison descendante. Les appareils de classe B peuvent recevoir des messages via des emplacements de liaison descendante classiques. Les appareils de classe C peuvent recevoir des messages de liaison descendante à tout moment. Pour de plus amples informations sur les catégories d'appareil, veuillez consulter [Classes d'appareil \(p. 1338\)](#).

Ce qui suit montre comment les messages sont mis en file d'attente et envoyés à vos appareils de classe A.

1. AWS IoT Core for LoRaWAN met en mémoire tampon le message de liaison descendante que vous avez ajouté à la file d'attente avec le port de trame, les données de charge utile et les paramètres du mode de confirmation que vous avez spécifiés à l'aide de la AWS IoT console ou de l'AWS IoT Wireless API.
2. Votre appareil LoRa WAN envoie un message de liaison montante pour indiquer qu'il est en ligne et qu'il peut commencer à recevoir des messages de liaison descendante.
3. Si vous avez ajouté plusieurs messages de liaison descendante à la file d'attente, AWS IoT Core for LoRaWAN envoie le premier message de liaison descendante de la file d'attente à votre appareil avec l'indicateur de confirmation (ACK) activé.
4. Votre appareil envoie un message de liaison montante à AWS IoT Core for LoRaWAN immédiatement, ou il se met en veille jusqu'au prochain message de liaison montante et inclut l'indicateur ACK dans le message.
5. Lorsque le message de liaison montante avec l'indicateur ACK est reçu, le message de liaison descendante est effacé de la file d'attente, indiquant que votre appareil a bien reçu le message de liaison descendante. Si l'indicateur ACK est absent du message de liaison montante après trois vérifications, le message est supprimé.

Exécuter des opérations de file d'attente de liaison descendante à l'aide de la console

Vous pouvez utiliser la AWS Management Console pour mettre en file d'attente les messages en liaison descendante et effacer des messages individuels, ou la totalité de la file d'attente, selon vos besoins. Pour les appareils de classe A, après réception d'une liaison montante en provenance de l'appareil indiquant qu'il est en ligne, les messages en file d'attente sont envoyés à l'appareil. Une fois le message envoyé, il est automatiquement supprimé de la file d'attente.

Mettre les messages de liaison descendante en file

Pour créer une file d'attente de messages à liaison descendante

1. Accédez au [hub Appareils de la AWS IoT console et choisissez l'appareil](#) pour lequel vous souhaitez mettre en file d'attente les messages en liaison descendante.
2. Dans la section Messages en liaison descendante de la page de détails de l'appareil, choisissez Mettre en file d'attente les messages en liaison descendante.
3. Spécifiez les paramètres suivants pour configurer votre message de liaison descendante :

- FPort : choisissez le port de trame avec lequel le périphérique doit communiquer AWS IoT Core for LoRaWAN.
 - Charge utile : Spécifiez le message de charge utile que vous souhaitez envoyer à votre appareil. La taille de la charge utile maximale est 242 octets. Si le débit de données adaptatif (ADR) est activé, AWS IoT Core for LoRaWAN utilisez-le pour choisir le débit de données optimal en fonction de la taille de votre charge utile. Vous pouvez optimiser davantage le débit de données selon vos besoins.
 - Mode de confirmation : confirmez si votre appareil a reçu le message de liaison descendante. Si un message nécessite ce mode, vous verrez un message de liaison montante avec l'indicateur ACK dans votre flux de données, et le message sera effacé de la file d'attente.
4. Pour ajouter votre message de lien descendant à la file d'attente, choisissez Soumettre.

Votre message de lien descendant a maintenant été ajouté à la file d'attente. Si votre message ne s'affiche pas ou si vous recevez un message d'erreur, vous pouvez résoudre l'erreur comme décrit dans [Résoudre les erreurs liées à la file de messages de liaison descendante \(p. 1344\)](#).

Note

Une fois que votre message de liaison descendante a été ajouté à la file d'attente, vous ne pouvez plus modifier les paramètres FPort, Payload et Acknowledge mode. Pour envoyer un message de liaison descendante avec des valeurs différentes pour ces paramètres, vous pouvez supprimer ce message et mettre en file d'attente un nouveau message de liaison descendante avec les valeurs de paramètres mises à jour.

La file d'attente répertorie les messages de liaison descendante que vous avez ajoutés. Pour voir la charge utile des messages de liaison montante et descendante échangés entre vos appareils AWS IoT Core for LoRaWAN, vous pouvez utiliser l'analyseur de réseau. Pour plus d'informations, veuillez consulter [Surveillance de votre parc de ressources sans fil en temps réel à l'aide d'un analyseur de réseau \(p. 1374\)](#).

Répertorier la file de messages en liaison descendante

Le message de liaison descendante que vous avez créé est ajouté à la file d'attente. Chaque message de liaison descendante suivant est ajouté à la file d'attente après ce message. Vous pouvez consulter la liste des messages de liaison descendante dans la section Messages de liaison descendante de la page de détails de l'appareil. Après réception d'une liaison montante, les messages sont envoyés à l'appareil. Une fois qu'un message de lien descendant a été reçu par votre appareil, il sera supprimé de la file d'attente. Le message suivant remonte ensuite dans la file d'attente pour être envoyé à votre appareil.

Supprimer des messages de liaison descendante individuels ou effacer toute la file d'attente

Chaque message de liaison descendante est automatiquement effacé de la file d'attente après avoir été envoyé à votre appareil. Vous pouvez également supprimer des messages individuels ou effacer la totalité de la file d'attente des liaisons descendantes. Ces actions ne peuvent pas être annulées.

- Si vous trouvez dans la file d'attente des messages que vous ne souhaitez pas envoyer, sélectionnez-les, puis choisissez Supprimer.
- Si vous ne souhaitez pas envoyer de messages depuis la file d'attente vers votre appareil, vous pouvez effacer toute la file d'attente en choisissant Effacer la file d'attente des liaisons descendantes.

Effectuez des opérations de file d'attente de liaison descendante à l'aide de l'API

Vous pouvez utiliser l'AWS IoT Wireless API pour mettre les messages en lien descendant en file d'attente et effacer des messages individuels, ou la totalité de la file d'attente, selon vos besoins.

Mettre les messages de liaison descendante en file

Pour créer une file de messages de liaison descendante, utilisez l'opération [SendDataToWirelessDevice](#) API ou la commande [send-data-to-wireless-device](#) CLI.

Note

Lorsque vous envoyez un message de liaison descendante à l'aide de l'`SendDataToWirelessDevice` API, vous pouvez choisir les passerelles que vous souhaitez utiliser pour le trafic de données en liaison descendante. Pour plus d'informations, veuillez consulter [Choix des passerelles pour recevoir le trafic de données de liaison descendante du LoRa WAN \(p. 1335\)](#).

```
aws iotwireless send-data-to-wireless-device \
--id "11aa5eae-2f56-4b8e-a023-b28d98494e49" \
--transmit-mode "1" \
--payload-data "SGVsbG8gVG8gRGV2c2lt" \
--wireless-metadata LoRaWAN={FFPort=1}
```

Le résultat de l'exécution de cette commande génère un `messageId` pour le lien descendant. Dans certains cas, même si vous recevez le `messageId`, des paquets peuvent être perdus. Pour en savoir plus sur la résolution de l'erreur, veuillez consulter [Résoudre les erreurs liées à la file de messages de liaison descendante \(p. 1344\)](#).

```
{  
    messageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Répertorier les messages de liaison descendante dans la file d'attente

Pour répertorier tous les messages de liaison descendante de la file d'attente, utilisez l'opération `ListQueuedMessages` API ou la commande [list-queued-messages](#) CLI.

```
aws iotwireless list-queued-messages
```

Par défaut, un maximum de 10 messages de liaison descendante sont affichés lors de l'exécution de cette commande.

Supprimer des messages de liaison descendante individuels ou effacer toute la file d'attente

Pour supprimer des messages individuels de la file d'attente ou pour effacer la totalité de la file d'attente, utilisez l'opération `DeleteQueuedMessages` API ou la commande [delete-queued-messages](#) CLI.

- Pour supprimer des messages individuels, indiquez `messageId` les messages que vous souhaitez supprimer de votre appareil sans fil, spécifiés par `fontWeightDeviceId`.
- Pour effacer l'intégralité de la file d'attente de liaison descendante, spécifiez `messageId` comme* pour votre périphérique sans fil, spécifié par `fontWeightDeviceId`.

Résoudre les erreurs liées à la file de messages de liaison descendante

Voici quelques points à vérifier si vous n'obtenez pas les résultats escomptés :

- Les messages de liaison descendante n'apparaissent pas dans la AWS IoT console

Si vous ne voyez pas votre message de lien descendant dans la file d'attente après l'avoir ajouté comme décrit dans la section [Exécuter des opérations de file d'attente de liaison descendante à l'aide de la console \(p. 1342\)](#), c'est peut-être parce que votre terminal n'a pas terminé un processus appelé procédure d'activation ou de connexion. Cette procédure est terminée lorsque votre appareil est intégré à AWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#).

Après avoir intégré votre appareil à AWS IoT Core for LoRaWAN, vous pouvez le surveiller pour vérifier si la connexion et la reconnexion ont réussi à l'aide de l'analyseur de réseau ou d'Amazon CloudWatch.

Pour plus d'informations, veuillez consulter [Surveillance et journalisation pour AWS IoT Wireless l'utilisation d'Amazon CloudWatch \(p. 1452\)](#).

- Paquets de messages de liaison descendante manquants lors de l'utilisation de l'API

Lorsque vous utilisez l'opération `SendDataToWirelessDevice` d'API, l'API renvoie une valeur `uniqueMessageId`. Cependant, il ne peut pas confirmer si votre périphérique LoRa WAN a reçu le message de liaison descendante. Les paquets de liaison descendante peuvent être supprimés dans des cas tels que lorsque votre appareil n'a pas terminé la procédure de connexion. Pour en savoir plus sur la résolution de cette erreur, veuillez consulter la section précédente.

- Erreur ARN manquante lors de l'envoi d'un message de liaison descendante

Lorsque vous envoyez un message de lien descendant à votre appareil depuis la file d'attente, vous pouvez recevoir une erreur Amazon Resource Name (ARN) manquante. Cette erreur peut se produire parce que la destination n'a pas été spécifiée correctement pour l'appareil qui reçoit le message de liaison descendante. Pour résoudre cette erreur, consultez les informations de destination pour votre appareil.

Gérez le trafic d'itinérance provenant d'appareils LoRa WAN situés hors du réseau domestique

La fonction d'itinérance est disponible en version préliminaire pour AWS IoT Core for LoRaWAN et susceptible d'être modifiée. Il est disponible pour une utilisation uniquement dans l'est (N. Virginia) (N. Virginia) et la région USA Ouest (Oregon) Régions AWS. Avant d'utiliser cette fonctionnalité, assurez-vous d'avoir lu les conditions générales de cette fonctionnalité. Pour plus d'informations, consultez [Conditions de service AWS](#).

AWS IoT Core for LoRaWAN prend en charge la fonctionnalité d'itinérance conformément à la spécification LoRa Alliance pour l'itinérance, telle que décrite dans la [spécification LoRa WAN Backend Interfaces 1.0](#). La fonctionnalité d'itinérance peut être utilisée pour connecter vos terminaux situés en dehors du réseau domestique. Pour soutenir cette capacité, AWS IoT Core for LoRaWAN les partenaires doivent EveryNet offrir une couverture radio étendue.

Avantages de l'utilisation d'un réseau LoRa WAN public

En utilisant un réseau LoRa WAN public, vous bénéficierez d'avantages tels que l'extension de la couverture, l'exécution du cœur sans réseau radio et la densification de la couverture. Cette fonctionnalité peut être utilisée pour :

- Fournissez une couverture aux appareils lorsqu'ils quittent leur réseau domestique, comme l'appareil A illustré dans la figure présentée dans la [LoRa Architecture d'itinérance WAN \(p. 1347\)](#) section.
- Étendez la couverture aux appareils qui ne disposent pas d'une LoRa passerelle à laquelle se connecter, tels que l'appareil B illustré dans la [LoRa Architecture d'itinérance WAN \(p. 1347\)](#) section. L'appareil peut ensuite utiliser la passerelle fournie par le partenaire pour se connecter au réseau domestique.

Vos appareils LoRa WAN peuvent utiliser un réseau public pour se connecter au cloud à l'aide de la fonction d'itinérance, ce qui réduit le temps de déploiement et réduit le temps et les coûts nécessaires à la maintenance d'un réseau LoRa WAN privé.

Les sections suivantes décrivent l'architecture d'itinérance, le fonctionnement de l'itinérance et l'utilisation de cette fonctionnalité.

Rubriques

- [Comment fonctionne l'itinérance sur un LoRa réseau WAN \(p. 1346\)](#)
- [Comment utiliser la fonction d'itinérance \(p. 1348\)](#)

Comment fonctionne l'itinérance sur un LoRa réseau WAN

AWS IoT Core for LoRaWAN prend en charge la fonction d'itinérance passive, conformément à la spécification de LoRa l'Alliance. Avec l'itinérance passive, le processus d'itinérance est totalement transparent pour l'appareil final. Les terminaux qui se déplacent en dehors du réseau domestique peuvent se connecter aux passerelles de ce réseau et échanger des données en liaison montante et descendante via le serveur d'applications. Les appareils restent connectés au réseau domestique pendant tout le processus d'itinérance.

Note

AWS IoT Core for LoRaWAN ne prend pas en charge le transfert en itinérance. Lors du transfert en itinérance, votre appareil passe à un autre opérateur lorsqu'il se déplace en dehors du réseau domestique.

AWS IoT Core for LoRaWAN ne prend en charge que la fonctionnalité apatride de l'itinérance passive. Grâce à l'itinérance passive sans état, AWS IoT Core for LoRaWAN il ne stocke aucune session de l'appareil et transmet le trafic de données sans effectuer de vérification de l'intégrité des messages (MIC).

Rubriques

- [LoRaConcepts d'itinérance sur un réseau WAN \(p. 1346\)](#)
- [LoRaArchitecture d'itinérance WAN \(p. 1347\)](#)

LoRaConcepts d'itinérance sur un réseau WAN

Les concepts suivants sont utilisés par la fonctionnalité d'itinérance prise en charge par AWS IoT Core for LoRaWAN.

LoRaServeur réseau WAN (LNS)

Un LNS est un serveur privé autonome qui peut fonctionner dans vos locaux ou qui peut être un service basé sur le cloud. AWS IoT Core for LoRaWAN est un LNS qui propose des services sur le cloud.

Serveur de réseau domestique (HNs)

Le réseau domestique est le réseau auquel appartient le périphérique. Le serveur de réseau domestique (HnS) est un LNS qui stocke les données de provisionnement du périphérique, telles que les DevEUI clés de session et AppEUI.

Serveur réseau visité (VNs)

Le réseau visité est le réseau à partir duquel l'appareil est couvert lorsqu'il quitte le réseau domestique. Le serveur réseau visité (vNs) est un LNS qui a conclu un accord commercial et technique avec le hNs pour pouvoir desservir le terminal. Dans cette version bêta, le partenaire EveryNet d'itinérance agit en tant que réseau visité pour fournir une couverture.

Serveur réseau de distribution (SnS)

Le serveur de réseau serveur (SnS) est un LNS qui gère les commandes MAC du périphérique. Il ne peut y avoir qu'un SN pour une LoRa session.

Serveur réseau de transfert (FNs)

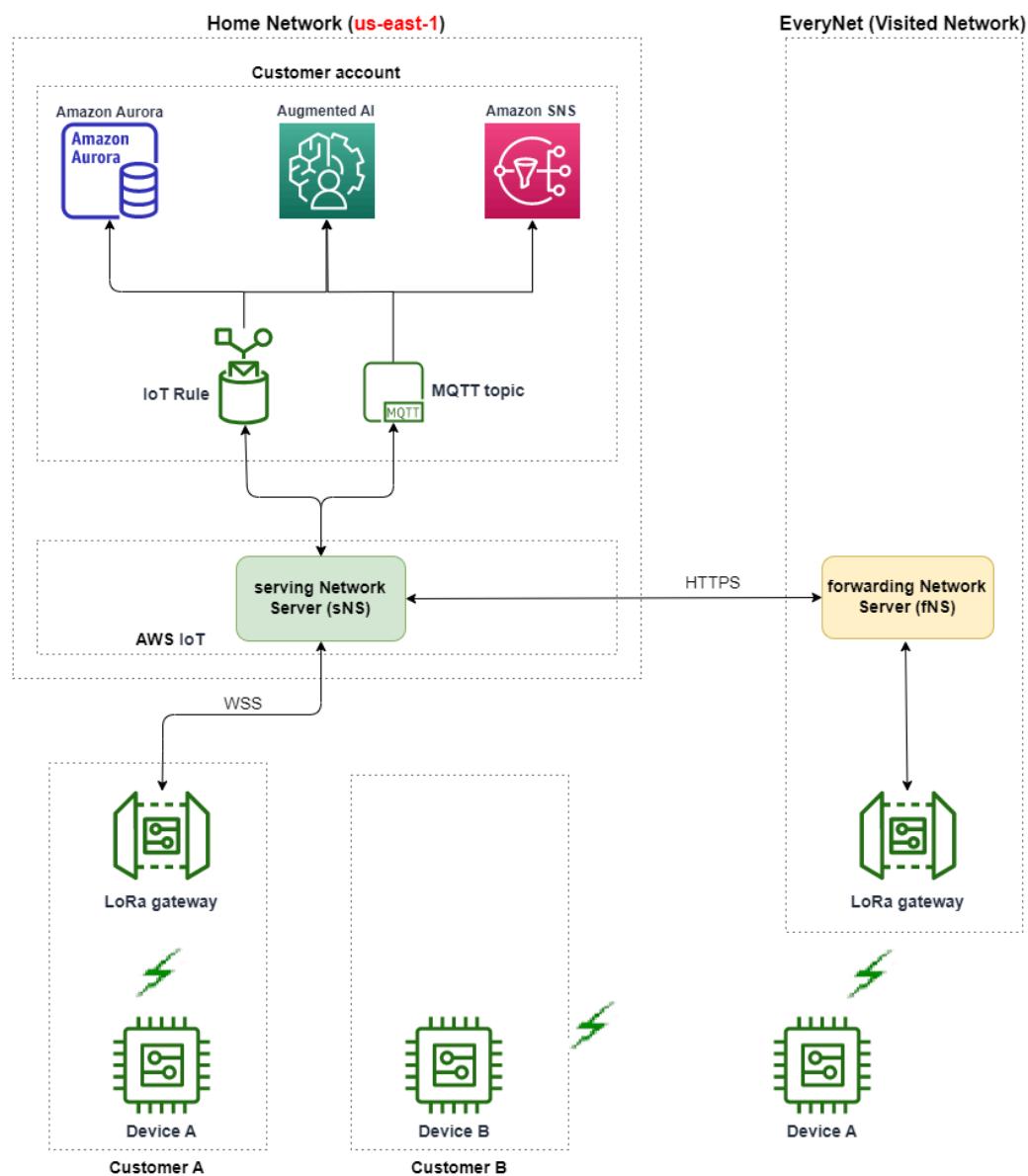
Le serveur réseau de transfert (FNs) est un LNS qui gère les passerelles radio. Il ne peut y avoir aucun ou plusieurs FN impliqués dans une LoRa session. Ce serveur réseau gère le transfert des paquets de données reçus de l'appareil vers le réseau domestique.

LoRaArchitecture d'itinérance WAN

Le schéma d'architecture suivant montre comment AWS IoT Core for LoRaWAN les partenaires peuvent EveryNet prendre en charge la fonctionnalité d'itinérance. Dans ce cas, le périphérique A est connecté au hNs (serveur de réseau domestique) fourni par le AWS IoT Core for LoRaWAN biais d'une LoRa passerelle. Lorsque le périphérique A quitte le réseau domestique, il entre dans un réseau visité et est couvert par le serveur réseau visité (VPN) fourni par EveryNet. Le VPN étend également la couverture au périphérique B qui ne dispose pas de LoRa passerelle à laquelle se connecter.

Note

Pour vérifier la EveryNet couverture, rendez-vous sur le [site Web d'Everynet](#). Vous pouvez également contacter l'équipe d'assistance de la version bêta pour obtenir des informations sur la demande d'une nouvelle couverture.



AWS IoT Core for LoRaWAN utilise une fonctionnalité de hub d'itinérance, conformément à la [recommandation technique du hub d'itinérance LoRa Alliance LoRa WAN](#). Le hub d'itinérance fournit un point de terminaison EveryNet pour acheminer le trafic reçu du terminal. Dans ce cas, EveryNet agit en tant que serveur réseau de transfert (FNs) pour transférer le trafic reçu du périphérique. Il utilise une API HTTP RESTful, telle que définie par la spécification LoRa Alliance.

Note

Si votre appareil quitte son réseau domestique et entre dans une zone où votre réseau domestique et votre réseau EveryNet peuvent offrir une couverture, il utilise une first-come-first-serve politique pour déterminer s'il doit se connecter à votre LoRa passerelle ou à EveryNet sa passerelle.

Dans un scénario d'itinérance, les HN et le serveur de réseau de distribution (SnS) sont séparés. Des paquets de liaison montante et descendante sont ensuite échangés entre les SN et les HN.

Comment utiliser la fonction d'itinérance

Pour activer la fonctionnalité d'itinérance, vous devez activer certains paramètres d'itinérance lors de la création du profil de service. Dans cette version bêta, ces paramètres sont disponibles lorsque vous utilisez l'AWS IoT WirelessAPI, ou le AWS CLI. Les sections suivantes indiquent les paramètres que vous devez activer et comment effectuer l'itinérance à l'aide du AWS CLI.

Note

Vous pouvez activer l'itinérance uniquement lors de la création d'un nouveau profil de service. Vous ne pouvez pas mettre à jour un profil existant pour activer l'itinérance à l'aide de ces paramètres.

Rubriques

- [Paramètres d'itinérance \(p. 1348\)](#)
- [Activer l'itinérance pour les appareils \(p. 1348\)](#)

Paramètres d'itinérance

Spécifiez les paramètres suivants lors de la création d'un profil de service pour votre appareil. Spécifiez ces paramètres lors de l'ajout d'un profil de service à partir du hub [Profiles](#) de la AWS IoT console, ou à l'aide de l'opération AWS IoT Wireless API ou de la AWS CLI commande, [create-service-profile](#), [CreateServiceProfile](#)

Note

AWS IoT Core for LoRaWAN ne prend pas en charge le transfert en itinérance dans cette version préliminaire. Lors de la création du profil de service, vous ne pouvez pas activer le `HrAllowed` paramètre qui indique si vous souhaitez utiliser l'itinérance avec transfert.

- Activation en itinérance autorisée (`RaAllowed`) : ce paramètre indique s'il faut activer l'activation en itinérance. L'activation de l'itinérance permet à un terminal de s'activer sous la couverture d'un réseau virtuel. Lorsque vous utilisez la fonction d'itinérance, vous `RaAllowed` devez le régler sur `true`.
- L'itinérance passive est autorisée (`PrAllowed`) : ce paramètre indique s'il faut activer l'itinérance passive. Lorsque vous utilisez la fonction d'itinérance, vous `PrAllowed` devez le régler sur `true`.

Activer l'itinérance pour les appareils

Pour activer l'itinérance sur vos appareils, exécutez la procédure suivante.

Note

La procédure suivante s'applique aux appareils OTAA. Si vous utilisez des appareils ABP, contactez le [AWSsupport](#) pour obtenir des instructions.

1. Création d'un profil de service avec des paramètres d'itinérance

Créez un profil de service en activant les paramètres d'itinérance.

- Utilisation de la console AWS IoT

Accédez au hub [Profils](#) de laAWS IoT console et choisissez Ajouter un profil de service. Lors de la création du profil, choisissez l'activation de l'itinérance autorisée et l'itinérance passive autorisée.

- Utilisation de l'API AWS IoT Wireless

Pour activer l'itinérance lors de la création d'un profil de service, utilisez l'opération [d>CreateServiceProfileAPI](#) ou la commande [create-service-profile](#) CLI, comme indiqué dans l'exemple ci-dessous.

```
aws iotwireless create-service-profile \
--region us-east-1 \
--name roamingprofile1 \
--lorawan '{"AddGwMetadata":true,"PrAllowed":true,"RaAllowed":true}'
```

L'exécution de cette commande renvoie l'ARN et l'ID du profil de service en sortie.

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
}
```

2. Vérifiez les paramètres d'itinérance dans le profil de service

Pour vérifier les paramètres d'itinérance que vous avez spécifiés, vous pouvez consulter le profil de service dans la console ou à l'aide de la commande [get-service-profile](#) CLI, comme indiqué dans l'exemple ci-dessous.

- Utilisation de la console AWS IoT

Accédez au hub [Profils](#) de laAWS IoT console et choisissez le profil que vous avez créé. Dans l'onglet Configuration du profil de la page de détails, vous pouvez voir RaAllowed et PrAllowed définis sur true.

- Utilisation de l'API AWS IoT Wireless

Pour afficher les paramètres d'itinérance que vous avez activés, utilisez l'opération [d.GetServiceProfileAPI](#) ou la commande [get-service-profile](#) CLI, comme indiqué dans l'exemple ci-dessous.

```
aws iotwireless get-service-profile \
--region us-east-1 \
--id 12345678-a1b2-3c45-67d8-e90fa1b2c34d
```

L'exécution de cette commande renvoie les détails du profil de service en sortie, y compris les valeurs des paramètres d'itinérance, RaAllowed et PrAllowed.

```
{
```

```
"Arn": "arn:aws:iotwireless:us-east-1:123456789012:ServiceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
    "Name": "roamingprofile1"
    "LoRaWAN": {
        "UlRate": 60,
        "UlBucketSize": 4096,
        "DlRate": 60,
        "DlBucketSize": 4096,
        "AddGwMetadata": true,
        "DevStatusReqFreq": 24,
        "ReportDevStatusBattery": false,
        "ReportDevStatusMargin": false,
        "DrMin": 0,
        "DrMax": 15,
        "PrAllowed": true,
        "RaAllowed": true,
        "NwkGeoLoc": false,
        "TargetPer": 5,
        "MinGwDiversity": 1
    }
}
```

3. Joindre un profil de service aux appareils

Attachez le profil de service que vous avez créé avec les paramètres d'itinérance à vos terminaux. Vous pouvez également créer un profil d'appareil et ajouter une destination pour vos appareils sans fil. Vous utiliserez cette destination pour acheminer les messages de liaison montante envoyés depuis votre appareil. Pour plus d'informations sur la création de profils d'appareils et d'une destination, consultez [Ajouter des profils d'appareil \(p. 1291\)](#) et [Ajouter des destinations à AWS IoT Core pour le LoRa WAN \(p. 1293\)](#).

- Intégration de nouveaux appareils

Si vous n'avez pas encore intégré vos appareils, vous devez spécifier ce profil de service à utiliser lors de l'ajout de votre appareil AWS IoT Core for LoRaWAN. La commande suivante montre comment utiliser la commande `create-wireless-device` CLI pour ajouter un appareil à l'aide de l'ID du profil de service que vous avez créé. Pour en savoir plus sur l'ajout du profil de service à l'aide de la console, veuillez consulter [Ajoutez les spécifications de votre appareil sans fil à AWS IoT Core for LoRaWAN l'utilisation de la console \(p. 1289\)](#).

```
aws iotwireless create-wireless-device --cli-input-json file://createdevice.json
```

L'exemple suivant affiche le contenu du fichier `createdevice.json`.

Contenu de `createdevice.json`

```
{
    "Name": "DeviceA",
    "Type": LoRaWAN,
    "DestinationName": "RoamingDestination1",
    "LoRaWAN": {
        "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
        "ServiceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
        "OtaaV1_1": {
            "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
            "JoinEui": "b4c231a359bc2e3d",
            "NwkKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
        },
        "DevEui": "ac12efc654d23fc2"
    }
},
```

}

Le résultat de l'exécution de cette commande produit l'ARN et l'ID du périphérique sans fil en sortie.

```
{  
    "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f",  
    "Id": "1ffd32c8-8130-4194-96df-622f072a315f"  
}
```

- Mise à jour des appareils existants

Si vous avez déjà intégré vos appareils, vous pouvez mettre à jour vos appareils sans fil existants pour utiliser ce profil de service. La commande suivante montre comment utiliser la commande `update-wireless-device` CLI pour mettre à jour un appareil à l'aide de l'ID du profil de service que vous avez créé.

```
aws iotwireless update-wireless-device \  
    --id "1ffd32c8-8130-4194-96df-622f072a315f" \  
    --service-profile-id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \  
    --description "Using roaming service profile A"
```

Cette commande ne produit aucune sortie. Vous pouvez utiliser l'`GetWirelessDevice` API ou la commande `get-wireless-device` CLI pour obtenir les informations mises à jour.

4. Connect l'appareil au cloud à l'aide de EveryNet

L'itinérance ayant été activée, votre appareil doit désormais effectuer une connexion pour en obtenir une nouvelle `DevAddr`. Si vous utilisez OTAA, votre périphérique LoRa WAN envoie une demande d'adhésion et le serveur réseau peut accepter cette demande. Il peut ensuite se connecter à l'AWS Cloud à la couverture réseau fournie par EveryNet. Pour obtenir des instructions sur la procédure d'activation ou de connexion pour votre appareil, consultez la documentation de l'appareil.

5. Échange de messages en liaison montante et en liaison descendante

Une fois votre appareil connecté à AWS IoT Core for LoRaWAN, vous pouvez commencer à échanger des messages entre votre appareil et le Cloud.

- Afficher les messages de liaison montante

Lorsque vous envoyez des messages par liaison montante depuis vos appareils, AWS IoT Core for LoRaWAN ces messages vous parviennent au Compte AWS en utilisant la destination que vous avez configurée précédemment. Ces messages seront envoyés depuis votre appareil vers le réseau Cloud. EveryNet

Vous pouvez soit afficher les messages à l'aide du nom de la règle, soit utiliser le client MQTT pour vous abonner à la rubrique MQTT spécifiée lors de la création de la destination. Pour plus d'informations sur le nom de la règle et les autres détails de destination que vous spécifiez, consultez [Ajouter une destination à l'aide de la console \(p. 1293\)](#).

Pour en savoir plus sur l'affichage d'un message à liaison montante et son format, veuillez consulter [Afficher le format des messages de liaison montante envoyés depuis des périphériques LoRa WAN \(p. 1339\)](#).

- Envoi de messages en lien descendant

Vous pouvez mettre en file d'attente et envoyer des messages en liaison descendante à vos appareils depuis la console, ou à l'aide de la commande `AWS IoT Wireless API` ou de la `AWS CLI` `send-data-to-wireless-device`. `SendDataToWirelessDevice` Pour plus d'informations sur la mise en file d'attente et l'envoi de messages de liaison descendante,

consultezMettre en file d'attente les messages de liaison descendante à envoyer aux périphériques LoRa WAN (p. 1341).

Le code suivant montre un exemple de la manière dont vous pouvez envoyer un message de liaison descendante à l'aide de la commande `send-data-to-wireless-device` CLI. Vous spécifiez l'ID du périphérique sans fil qui recevra les données, la charge utile, l'utilisation du mode accusé de réception et les métadonnées sans fil.

```
aws iotwireless send-data-to-wireless-device \
--id "1ffd32c8-8130-4194-96df-622f072a315f" \
--transmit-mode "1" \
--payload-data "SGVsbG8gVG8gRGV2c21t" \
--wireless-metadata LoRaWAN={FPort=1}
```

Le résultat de l'exécution de cette commande génère un message `MessageId` pour le lien descendant.

Note

Dans certains cas, même si vous recevez le `MessageId`, des paquets peuvent être perdus. Pour plus d'informations sur le dépannage de tels scénarios et leur résolution, consultezRésoudre les erreurs liées à la file de messages de liaison descendante (p. 1344).

```
{  
    MessageId: "6011dd36-0043d6eb-0072-0008"  
}
```

Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils

Pour envoyer une charge utile en liaison descendante à plusieurs appareils, créez un groupe de multidiffusion. Grâce à la multidiffusion, une source peut envoyer des données à une adresse de multidiffusion unique, qui est ensuite distribuée à un groupe complet de périphériques destinataires.

Les appareils d'un groupe de multidiffusion partagent la même adresse de multidiffusion, les mêmes clés de session et le même compteur d'images. En utilisant les mêmes clés de session, les appareils d'un groupe de multidiffusion peuvent déchiffrer le message lorsqu'une transmission en liaison descendante est initiée. Un groupe de multidiffusion ne prend en charge que les liaisons descendantes. Il ne confirme pas si la charge utile de liaison descendante a été reçue par les appareils.

Avec AWS IoT Core for LoRaWAN les groupes de multidiffusion, vous pouvez :

- Filtrez votre liste d'appareils en utilisant le profil de l'appareil, la région RF ou la classe d'appareils, puis ajoutez ces appareils à un groupe de multidiffusion.
- Planifiez et envoyez un ou plusieurs messages de charge utile en liaison descendante aux appareils d'un groupe de multidiffusion, dans un délai de distribution de 48 heures.
- Demandez aux appareils de passer temporairement en mode Classe B ou Classe C au début de votre session de multidiffusion pour recevoir le message de liaison descendante.
- Surveillez la configuration de votre groupe de multidiffusion et l'état de ses appareils, et résolvez les problèmes éventuels.
- Utilisez Firmware Updates-Over-The-Air (FUOTA) pour déployer en toute sécurité les mises à jour du microprogramme sur les appareils d'un groupe de multidiffusion.

AWS IoT Core for LoRaWANLa prise en charge du FUOTA et des groupes de multidiffusion repose sur les spécifications suivantes [deLoRa l'Alliance](#) :

- LoRaSpécification de configuration de la multidiffusion à distance WAN, TS005-2.0.0
- LoRaSpécification de transport par blocs de données fragmentés sur un réseau WAN, TS004-2.0.0
- LoRaSpécification de synchronisation de l'horloge de la couche d'application WAN, TS003-2.0.0

Note

AWS IoT Core for LoRaWANeffectue automatiquement la synchronisation de l'horloge de l'appareil conformément aux spécifications de l' LoRa Alliance. À l'aide de la fonctionAppTimeReq, il répond à l'heure côté serveur aux appareils qui le demandent en utilisant la ClockSync signalisation.

Ce qui suit montre comment créer votre groupe de multidiffusion et planifier un message de liaison descendante.

Rubriques

- [Préparer les appareils pour la multidiffusion et la configuration FUOTA \(p. 1353\)](#)
- [Créez des groupes de multidiffusion et ajoutez des appareils au groupe \(p. 1356\)](#)
- [Surveillez et résolvez les problèmes liés à l'état de votre groupe de multidiffusion et des appareils du groupe \(p. 1359\)](#)
- [Programmez un message de liaison descendante à envoyer aux appareils de votre groupe de multidiffusion \(p. 1361\)](#)

Préparer les appareils pour la multidiffusion et la configuration FUOTA

Lorsque vous ajoutez votre périphérique sans fil àAWS IoT Core for LoRaWAN, vous pouvez le préparer pour la configuration de multidiffusion et la configuration FUOTA à l'aide de la console ou de l'interface de ligne de commande. Si vous effectuez cette configuration pour la première fois, nous vous recommandons d'utiliser la console. Pour gérer votre groupe de multidiffusion et ajouter ou supprimer plusieurs appareils de votre groupe, nous vous recommandons d'utiliser la CLI pour gérer un grand nombre de ressources.

GenAppKey et FPorts

Lorsque vous ajoutez votre appareil sans fil, avant de pouvoir ajouter vos appareils à des groupes de multidiffusion ou d'effectuer des mises à jour FUOTA, configurez les paramètres suivants. Avant de configurer ces paramètres, assurez-vous que vos appareils prennent en charge le FUOTA et la multidiffusion et que la spécification de votre appareil sans fil est l'uneOTAA v1.1 ou l'autreOTAAv1.0.x.

- GenAppKey: pour les appareils qui prennent en charge la version LoRa WAN 1.0.x et qui utilisent des groupes de multidiffusion,GenAppKey il s'agit de la clé racine spécifique à l'appareil à partir de laquelle les clés de session de votre groupe de multidiffusion sont dérivées.

Note

Pour les périphériques LoRa WAN qui utilisent la spécification sans filOTAA v1.1, leAppKey est utilisé dans le même but que leGenAppKey.

Pour configurer les paramètres nécessaires au lancement du transfert de données,AWS IoT Core for LoRaWAN distribue les clés de session aux appareils finaux. Pour plus d'informations sur les versions LoRa WAN, consultez[LoRaVersion WAN \(p. 1338\)](#).

Note

AWS IoT Core for LoRaWAN enregistre les informations GenAppKey que vous fournissez dans un format crypté.

- **FPorts:** conformément aux spécifications LoRa WAN pour les groupes FUOTA et de multidiffusion, AWS IoT Core for LoRaWAN attribue les valeurs par défaut aux champs suivants du FPorts paramètre. Si vous avez déjà attribué l'une des FPort valeurs suivantes, vous pouvez choisir une autre valeur disponible, comprise entre 1 et 223.
- **Multicast:** 200

Cette FPort valeur est utilisée pour les groupes de multidiffusion.

- **FUOTA:** 201

Cette FPort valeur est utilisée pour FUOTA.

- **ClockSync:** 202

Cette FPort valeur est utilisée pour la synchronisation de l'horloge.

Profils d'appareils pour la multidiffusion et le FUOTA

Au début d'une session de multidiffusion, une fenêtre de distribution de classe B ou de classe C est utilisée pour envoyer le message de liaison descendante aux appareils de votre groupe. Les appareils que vous ajoutez pour la multidiffusion et le FUOTA doivent prendre en charge les modes de fonctionnement de classe B ou de classe C. En fonction de la classe d'appareil prise en charge par votre appareil, choisissez un profil d'appareil pour lequel l'un des modes de classe B ou de classe C est activé, ou les deux.

Pour plus d'informations sur les profils des appareils, reportez-vous à la section [Ajoutez des profils à AWS IoT Core for LoRaWAN \(p. 1291\)](#).

Préparez les appareils pour la multidiffusion et le FUOTA à l'aide de la console

Pour spécifier les FPorts et les GenAppKey paramètres pour la configuration de la multidiffusion et le FUOTA à l'aide de la console :

1. Accédez au [hub Appareils de la AWS IoT console](#) et choisissez Ajouter un appareil sans fil.
2. Choisissez la spécification du périphérique sans fil. Votre appareil doit utiliser OTAA pour l'activer. Lorsque vous choisissez OTAA v1.0.x ou OTAA v1.1, une section facultative de configuration FUOTA apparaît.
3. Entrez les paramètres EUI (Extended Unique Identifier) de votre appareil sans fil.
4. Développez la section Configuration FUOTA facultative, puis choisissez Cet appareil prend en charge les mises à jour du micrologiciel par liaison radio (FUOTA). Vous pouvez désormais saisir les valeurs FPort pour la multidiffusion, le FUOTA et la synchronisation de l'horloge. Si vous avez choisi OTAA v1.0.x la spécification du périphérique sans fil, entrez le GenAppKey.
5. Ajoutez votre appareil AWS IoT Core for LoRaWAN en choisissant vos profils et une destination pour le routage des messages. Pour le profil d'appareil associé à l'appareil, assurez-vous de sélectionner l'un des modes Supports Class B ou Supports Class C.

Note

Pour spécifier les paramètres de configuration FUOTA, vous devez utiliser le [hub Devices de la AWS IoT console](#). Ces paramètres n'apparaissent pas si vous intégrez vos appareils via la page d'introduction de la AWS IoT console.

Pour plus d'informations sur les spécifications de l'appareil sans fil et l'intégration de votre appareil, consultez [Ajoutez votre appareil sans fil à AWS IoT Core for LoRaWAN \(p. 1289\)](#).

Note

Vous ne pouvez spécifier ces paramètres que lorsque vous créez le périphérique sans fil. Vous ne pouvez ni modifier ni spécifier de paramètres lorsque vous mettez à jour un appareil existant.

Préparez les appareils pour la multidiffusion et le FUOTA à l'aide de l'opération API

Pour utiliser des groupes de multidiffusion ou pour effectuer des mises à jour FUOTA, configurez ces paramètres à l'aide de l'opération [CreateWirelessDevice](#) API ou de la commande [create-wireless-device](#) CLI. Outre la spécification de la clé d'application et des paramètres FPorts, assurez-vous que le profil de périphérique lié au périphérique prend en charge un ou les deux modes de classe B ou de classe C.

Vous pouvez fournir un `input.json` fichier en entrée à la `create-wireless-device` commande.

```
aws iotwireless create-wireless-device \
--cli-input-json file://input.json
```

où :

Contenu du fichier `input.json`

```
{
  "Description": "My LoRaWAN wireless device",
  "DestinationName": "IoTWirelessDestination",
  "LoRaWAN": {
    "DeviceProfileId": "ab0c23d3-b001-45ef-6a01-2bc3de4f5333",
    "ServiceProfileId": "fe98dc76-cd12-001e-2d34-5550432da100",
    "FPorts": {
      "ClockSync": 202,
      "Fuota": 201,
      "Multicast": 200
    },
    "OtaaV1_0_x": {
      "AppKey": "3f4ca100e2fc675ea123f4eb12c4a012",
      "AppEui": "b4c231a359bc2e3d",
      "GenAppKey": "01c3f004a2d6efffe32c4eda14bcd2b4"
    },
    "DevEui": "ac12efc654d23fc2"
  },
  "Name": "SampleIoTWirelessThing",
  "Type": LoRaWAN
}
```

Pour plus d'informations sur les commandes CLI que vous pouvez utiliser, consultez la [AWS CLI référence](#).

Note

Une fois que vous avez spécifié les valeurs de ces paramètres, vous ne pouvez pas les mettre à jour à l'aide de l'opération `UpdateWirelessDevice` d'API. Au lieu de cela, vous pouvez créer un nouveau périphérique avec les valeurs des paramètres `GenAppKey` et `FPorts`.

Pour obtenir des informations sur les valeurs spécifiées pour ces paramètres, vous pouvez utiliser l'opération [GetWirelessDevice](#) API ou la commande [get-wireless-device](#) CLI.

Étapes suivantes

Après avoir configuré les paramètres, vous pouvez créer des groupes de multidiffusion et des tâches FUOTA pour envoyer une charge utile en liaison descendante ou mettre à jour le micrologiciel de vos périphériques LoRa WAN.

- Pour en savoir plus sur la création de groupes multidiffusion, veuillez consulter [Créez des groupes de multidiffusion et ajoutez des appareils au groupe \(p. 1356\)](#).
- Pour de plus amples informations sur la création de tâches FUOTA, veuillez consulter [Création d'une tâche FUOTA et fourniture d'une image du microprogramme \(p. 1366\)](#).

Créez des groupes de multidiffusion et ajoutez des appareils au groupe

Vous pouvez créer des groupes multidiffusion à l'aide de la console ou de l'interface de ligne de commande. Si vous créez votre groupe de multidiffusion pour la première fois, nous vous recommandons d'utiliser la console pour ajouter votre groupe de multidiffusion. Lorsque vous souhaitez gérer votre groupe de multidiffusion et ajouter ou supprimer des appareils de votre groupe, vous pouvez utiliser l'interface de ligne de commande.

Après avoir échangé la signalisation avec les terminaux que vous avez ajoutés, AWS IoT Core for LoRaWAN établit les clés partagées avec les terminaux et définit les paramètres pour le transfert de données.

Prérequis

Avant de créer des groupes de multidiffusion et d'ajouter des appareils au groupe, procédez comme suit :

- Préparez vos appareils pour la multidiffusion et la configuration FUOTA en spécifiant les paramètres de configuration FUOTAGenAppKey et FPorts. Pour plus d'informations, veuillez consulter [Préparer les appareils pour la multidiffusion et la configuration FUOTA \(p. 1353\)](#).
- Vérifiez si les appareils prennent en charge les modes de fonctionnement de classe B ou de classe C. En fonction de la classe d'appareil prise en charge par votre appareil, choisissez un profil d'appareil sur lequel l'un ou les deux modes Supporte la Classe B ou Supporte la Classe C sont activés. Pour plus d'informations sur les profils des appareils, reportez-vous à la section [Ajoutez des profils à AWS IoT Core for LoRaWAN \(p. 1291\)](#).

Au début de la session de multidiffusion, une fenêtre de distribution de classe B ou de classe C est utilisée pour envoyer des messages de liaison descendante aux appareils de votre groupe.

Création de groupes de multidiffusion à l'aide de la console

Pour créer des groupes de multidiffusion à l'aide de la console, accédez à la page [Groupes de multidiffusion](#) de la AWS IoT console et choisissez Créez un groupe de multidiffusion.

1. Création d'un groupe de multidiffusion

Pour créer votre groupe de multidiffusion, spécifiez les propriétés et les balises de multidiffusion de votre groupe.

1. Spécifier les propriétés de multidiffusion

Pour spécifier les propriétés de multidiffusion, entrez les informations suivantes pour votre groupe de multidiffusion.

- Nom : saisissez un nom unique pour votre groupe multidiffusion. Le nom ne doit contenir que des lettres, des chiffres, des traits d'union et des traits de soulignement. Il ne doit pas contenir d'espace.
- Description : Vous pouvez fournir une description facultative pour votre groupe de multidiffusion. La longueur de la description peut aller jusqu'à 2 048 caractères

2. Tags pour groupe de multidiffusion

Vous pouvez éventuellement fournir des paires clé-valeur en tant que balises pour votre groupe de multidiffusion. Pour continuer à créer votre groupe de multidiffusion, choisissez Next.

2. Ajouter des appareils à un groupe de multidiffusion

Vous pouvez ajouter des appareils individuels ou un groupe d'appareils à votre groupe de multidiffusion. Pour ajouter des appareils, procédez comme suit :

1. Spécifiez RFRegion

Spécifiez la région RF ou la bande de fréquences de votre groupe de multidiffusion. La région RF de votre groupe de multidiffusion doit correspondre à la région RF des appareils que vous ajoutez au groupe de multidiffusion. Pour de plus amples informations sur la région RF, veuillez consulter[Envisagez de sélectionner des bandes de LoRa fréquences pour vos passerelles et la connexion de vos appareils \(p. 1280\)](#).

2. Sélectionnez une classe d'appareils de multidiffusion

Choisissez si vous souhaitez que les appareils du groupe de multidiffusion passent en mode de classe B ou de classe C au début de la session de multidiffusion. Une session de classe B peut recevoir des messages de liaison descendante dans des emplacements de liaison descendante normaux et une session de classe C peut recevoir des messages de liaison descendante à tout moment.

3. Choisissez les appareils que vous souhaitez ajouter au groupe

Choisissez si vous souhaitez ajouter des appareils individuellement ou en masse au groupe de multidiffusion.

- Pour ajouter des appareils individuellement, entrez l'identifiant de l'appareil sans fil de chaque appareil que vous souhaitez ajouter à votre groupe.
- Pour ajouter des appareils en masse, vous pouvez filtrer les appareils que vous souhaitez ajouter par profil d'appareil ou par balises. Pour le profil de l'appareil, vous pouvez ajouter des appareils dont le profil prend en charge la classe B, la classe C ou les deux classes d'appareils.

4. Pour créer votre groupe de multidiffusion, choisissez Créer.

Les détails du groupe de multidiffusion et les appareils que vous avez ajoutés apparaissent dans le groupe. Pour plus d'informations sur l'état du groupe de multidiffusion et de vos appareils et pour résoudre les problèmes éventuels, consultez[Surveillez et résolvez les problèmes liés à l'état de votre groupe de multidiffusion et des appareils du groupe \(p. 1359\)](#).

Après avoir créé un groupe de multidiffusion, vous pouvez choisir Action pour modifier, supprimer ou ajouter des appareils au groupe de multidiffusion. Après avoir ajouté les appareils, vous pouvez planifier une session pour que la charge utile en liaison descendante soit envoyée aux appareils de votre groupe.

Création de groupes de multidiffusion à l'aide de l'API

Pour créer des groupes de multidiffusion et y ajouter des appareils à l'aide de l'API, procédez comme suit :

1. Crédit d'un groupe de multidiffusion

Pour créer votre groupe de multidiffusion, utilisez l'opération [CreateMulticastGroup](#) API ou la commande [create-multicast-group](#) CLI. Vous pouvez fournir un `input.json` fichier en entrée à la `create-multicast-group` commande.

```
aws iotwireless create-multicast-group \
--cli-input-json file://input.json
```

où :

Contenu du fichier input.json

```
{  
    "Description": "Multicast group to send downlink payload and perform FUOTA updates.",  
    "LoRaWAN": {  
        "DlClass": "ClassB",  
        "RfRegion": "US915"  
    },  
    "Name": "MC_group_FUOTA"  
}
```

Après avoir créé votre groupe de multidiffusion, vous pouvez utiliser les opérations d'API ou les commandes CLI suivantes pour mettre à jour, supprimer ou obtenir des informations sur vos groupes de multidiffusion.

- [UpdateMulticastGroup](#) ou [update-multicast-group](#)
- [GetMulticastGroup](#) ou [get-multicast-group](#)
- [ListMulticastGroups](#) ou [list-multicast-groups](#)
- [DeleteMulticastGroup](#) ou [delete-multicast-group](#)

2. Ajouter des appareils à un groupe de multidiffusion

Vous pouvez ajouter des appareils à votre groupe de multidiffusion individuellement ou en masse.

- Pour ajouter des appareils en masse à votre groupe de multidiffusion, utilisez l'opération [StartBulkAssociateWirelessDeviceWithMulticastGroup](#) API ou la commande [start-bulk-associate-wireless-device-with-multicast-group](#) CLI. Pour filtrer les appareils que vous souhaitez associer en bloc à votre groupe de multidiffusion, fournissez une chaîne de requête. Ce qui suit montre comment ajouter un groupe d'appareils doté d'un profil d'appareil associé à l'ID spécifié.

```
aws iotwireless start-bulk-associate-wireless-device-with-multicast-group \  
--id "12abd34e-5f67-89c2-9293-593b1bd862e0" \  
--cli-input-json file://input.json
```

où :

Contenu du fichier input.json

```
{  
    "QueryString": "DeviceProfileName: MyWirelessDevice AND DeviceProfileId:  
d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf",  
    "Tags": [  
        {  
            "Key": "Multicast",  
            "Value": "ClassB"  
        }  
    ]  
}
```

multicast-groups/d6d8ef8e-7045-496d-b3f4-ebcaa1d564bf/bulkVoici l'URL utilisée pour associer les appareils au groupe.

- Pour ajouter des appareils individuellement à votre groupe de multidiffusion, utilisez l'opération [AssociateWirelessDeviceWithMulticastGroup](#) API ou la [associate-wireless-device-with-multicast-group](#) CLI. Fournissez l'identifiant de l'appareil sans fil pour chaque appareil que vous souhaitez ajouter à votre groupe.

```
aws iotwireless associate-wireless-device-with-multicast-group \
--id "12abd34e-5f67-89c2-9293-593b1bd862e0" \
--wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

Après avoir créé votre groupe de multidiffusion, vous pouvez utiliser les opérations d'API ou les commandes CLI suivantes pour obtenir des informations sur votre groupe de multidiffusion ou pour dissocier des appareils.

- [DisassociateWirelessDeviceFromMulticastGroup](#) ou [disassociate-wireless-device-from-multicast-group](#)
- [StartBulkDisassociateWirelessDeviceFromMulticastGroup](#) ou [start-bulk-disassociate-wireless-device-from-multicast-group](#)
- [ListWirelessDevices](#) ou [list-wireless-devices](#)

Note

L'opération `ListWirelessDevices` API peut être utilisée pour répertorier les appareils sans fil en général et les appareils sans fil associés à un groupe de multidiffusion ou à une tâche FUOTA.

- Pour répertorier les appareils sans fil associés à un groupe de multidiffusion, utilisez l'opération d'`ListWirelessDevices` API `withMulticastGroupID` comme filtre.
- Pour répertorier les appareils sans fil associés à une tâche FUOTA, utilisez l'opération d'`ListWirelessDevices` API `withFuotaTaskID` comme filtre.

Étapes suivantes

Après avoir créé un groupe de multidiffusion et ajouté des appareils, vous pouvez continuer à ajouter des appareils et surveiller l'état du groupe de multidiffusion et de vos appareils. Si vos appareils ont été ajoutés avec succès au groupe, vous pouvez configurer et planifier l'envoi d'un message de liaison descendante aux appareils. Avant de pouvoir envoyer un message de liaison descendante, l'état de vos appareils doit être prêt à être configuré pour la multidiffusion. Une fois que vous avez programmé un message de liaison descendante, le statut passe à Tentative de session. Pour plus d'informations, veuillez consulter [Programmez un message de liaison descendante à envoyer aux appareils de votre groupe de multidiffusion \(p. 1361\)](#).

Si vous souhaitez mettre à jour le micrologiciel des appareils du groupe de multidiffusion, vous pouvez effectuer des mises à jour du microprogramme en direct (FUOTA) avec AWS IoT Core for LoRaWAN. Pour plus d'informations, veuillez consulter [Mises à jour du micrologiciel en direct \(FUOTA\) pour les AWS IoT Core for LoRaWAN appareils \(p. 1363\)](#).

Si vos appareils n'ont pas été ajoutés ou si vous constatez une erreur dans le groupe de multidiffusion ou l'état des appareils, vous pouvez passer la souris sur l'erreur pour obtenir plus d'informations et la résoudre. Si le message d'erreur persiste, pour plus d'informations sur la façon de le dépanner et de le résoudre, consultez [Surveillez et résolvez les problèmes liés à l'état de votre groupe de multidiffusion et des appareils du groupe \(p. 1359\)](#).

Surveillez et résolvez les problèmes liés à l'état de votre groupe de multidiffusion et des appareils du groupe

Après avoir ajouté des appareils et créé votre groupe de multidiffusion, ouvrez le AWS Management Console. Accédez à la page [Groupes de multidiffusion](#) de la AWS IoT console et choisissez le groupe de multidiffusion que vous avez créé pour afficher ses détails. Vous verrez des informations sur le groupe de multidiffusion, le nombre d'appareils ajoutés et les détails de l'état des appareils. Vous pouvez utiliser les

informations d'état pour suivre la progression de votre session de multidiffusion et résoudre les erreurs éventuelles.

État du groupe de multidiffusion

L'un des messages d'état suivants peut s'afficher dans le groupe de multidiffusion AWS Management Console.

- En suspens

Ce statut indique que vous avez créé un groupe de multidiffusion mais qu'il n'existe pas encore de session de multidiffusion. Ce message d'état s'affichera lorsque votre groupe aura été créé. Pendant ce temps, vous pouvez mettre à jour votre groupe de multidiffusion et associer ou dissocier des appareils à votre groupe. Une fois que le statut est passé de En attente, aucun appareil supplémentaire ne peut être ajouté au groupe.

- Tentative de session

Une fois que vos appareils ont été ajoutés avec succès au groupe de multidiffusion, lorsque votre groupe a une session de multidiffusion planifiée, ce message d'état s'affiche. Pendant ce temps, vous ne pouvez ni mettre à jour ni ajouter d'appareils à votre groupe de multidiffusion. Si vous annulez votre session de multidiffusion, le statut du groupe passe à En attente.

- En session

Ce message d'état s'affiche lorsque c'est la première fois que votre session de multidiffusion est ouverte. Un groupe de multidiffusion continue également d'être dans cet état lorsqu'il est associé à une tâche FUOTA faisant l'objet d'une session de mise à jour du microprogramme en cours.

Si aucune tâche FUOTA n'est associée à la session et si la session de multidiffusion est annulée parce que la durée de la session a dépassé le délai d'expiration ou si vous avez annulé votre session de multidiffusion, le statut du groupe passe à En attente.

- Supprimer l'attente

Si vous supprimez votre groupe de multidiffusion, son statut passe à Supprimer en attente. Les suppressions sont permanentes et ne peut pas être annulée. Cette action peut prendre du temps et le statut du groupe sera Delete_Waiting jusqu'à ce que le groupe de multidiffusion soit supprimé. Une fois que votre groupe de multidiffusion est entré dans cet état, il ne peut pas passer à l'un des autres états.

État des appareils dans le groupe de multidiffusion

L'un des messages d'état suivants peut être affiché sur les appareils de votre groupe de multidiffusion dans le AWS Management Console. Vous pouvez passer la souris sur chaque message d'état pour obtenir plus d'informations sur ce qu'il indique.

- Tentative de Package

Une fois que vos appareils ont été associés au groupe de multidiffusion, le statut de l'appareil est En cours de création de package. Cet état indique qu'il n'AWS IoT Core for LoRaWAN pas encore été confirmé si l'appareil prend en charge la configuration et le fonctionnement de la multidiffusion.

- Package non pris en charge

Une fois que vos appareils ont été associés au groupe de multidiffusion, AWS IoT Core for LoRaWAN vérifie si le microprogramme de votre appareil est capable de configurer et de faire fonctionner la multidiffusion. Si le package de multidiffusion n'est pas pris en charge sur votre appareil, son statut est Package non pris en charge. Pour résoudre l'erreur, vérifiez si le microprogramme de votre appareil est capable de configurer et de faire fonctionner la multidiffusion.

- Tentative de configuration de multidiffusion

Si les appareils associés à votre groupe de multidiffusion sont capables de configurer et de faire fonctionner la multidiffusion, l'état indique Tentative de configuration de multidiffusion. Cet état indique que l'appareil n'a pas encore terminé la configuration de multidiffusion.

- Prêt pour la configuration de multidiffusion

Votre appareil a terminé la configuration de multidiffusion et a été ajouté au groupe de multidiffusion. Cet état indique que les appareils sont prêts pour une session de multidiffusion et qu'un message de liaison descendante peut être envoyé à ces appareils. L'état indique également quand vous pouvez utiliser FUOTA pour mettre à jour le micrologiciel des appareils du groupe.

- Tentative de session

Une session de multidiffusion a été planifiée pour les appareils de votre groupe de multidiffusion. Au début d'une session de groupe de multidiffusion, l'état du périphérique est Session en cours, et des demandes sont envoyées pour savoir si une fenêtre de distribution de classe B ou de classe C peut être lancée pour la session. Si le temps nécessaire à la configuration de la session de multidiffusion dépasse le délai imparti ou si vous annulez la session de multidiffusion, le statut passe à Configuration de multidiffusion terminée.

- En session

Cet état indique qu'une fenêtre de distribution de classe B ou de classe C a été lancée et que votre appareil dispose d'une session de multidiffusion en cours. Pendant ce temps, des messages en liaison descendante peuvent être AWS IoT Core for LoRaWAN envoyés depuis les appareils du groupe de multidiffusion. Si vous mettez à jour l'heure de votre session, elle remplace la session en cours et le statut passe à Tentative de session. À la fin de la session ou si vous annulez la session de multidiffusion, le statut passe à Prêt pour la configuration de multidiffusion.

Étapes suivantes

Maintenant que vous connaissez les différents statuts de votre groupe de multidiffusion et des appareils de votre groupe, et que vous savez comment résoudre les problèmes tels que l'impossibilité de configurer la multidiffusion sur un appareil, vous pouvez planifier l'envoi d'un message de liaison descendante aux appareils et votre groupe de multidiffusion sera en session. Pour plus d'informations sur la planification d'un message de liaison descendante, consultez [Programmez un message de liaison descendante à envoyer aux appareils de votre groupe de multidiffusion \(p. 1361\)](#).

Programmez un message de liaison descendante à envoyer aux appareils de votre groupe de multidiffusion

Après avoir ajouté des appareils à votre groupe de multidiffusion, vous pouvez démarrer une session de multidiffusion et configurer un message de liaison descendante à envoyer à ces appareils. Le message de liaison descendante doit être programmé dans les 48 heures et l'heure de début de la multidiffusion doit être postérieure d'au moins 30 minutes à l'heure actuelle.

Note

Les appareils d'un groupe de multidiffusion ne peuvent pas accuser réception d'un message de liaison descendante.

Prérequis

Avant de pouvoir envoyer un message de liaison descendante, vous devez avoir créé un groupe de multidiffusion et ajouté avec succès des appareils au groupe pour lequel vous souhaitez envoyer un message de liaison descendante. Vous ne pouvez pas ajouter d'autres appareils une fois qu'une heure de début a été planifiée pour votre session de multidiffusion. Pour plus d'informations, veuillez consulter [Créez des groupes de multidiffusion et ajoutez des appareils au groupe \(p. 1356\)](#).

Si l'un des appareils n'a pas été ajouté correctement, le groupe de multidiffusion et l'état de l'appareil contiendront des informations qui vous aideront à résoudre les erreurs. Si les erreurs persistent, pour plus d'informations sur la résolution de ces erreurs, consultez [Surveillez et résolvez les problèmes liés à l'état de votre groupe de multidiffusion et des appareils du groupe \(p. 1359\)](#).

Programmez un message de liaison descendante à l'aide de la console

Pour envoyer un message de liaison descendante à l'aide de la console, accédez à la page [Groupes de multidiffusion](#) de la AWS IoT console et choisissez le groupe de multidiffusion que vous avez créé. Sur la page de détails du groupe de multidiffusion, choisissez Planifier un message de liaison descendante, puis choisissez Planifier une session de liaison descendante.

1. Planifier une fenêtre de message à lien descendant

Vous pouvez définir une fenêtre horaire pour l'envoi d'un message de liaison descendante aux appareils de votre groupe de multidiffusion. Le message de liaison descendante doit être programmé dans les 48 heures.

Pour planifier votre session multidiffusion, spécifiez les paramètres suivants :

- Date et heure de début : La date et l'heure de début doivent se situer au moins 30 minutes après et 48 heures avant l'heure actuelle.

Note

L'heure que vous spécifiez est en UTC. Pensez donc à vérifier le décalage horaire avec votre fuseau horaire lorsque vous planifiez la fenêtre de liaison descendante.

- Délai d'expiration de session : délai après lequel vous souhaitez que la session de multidiffusion expire si aucun message de liaison descendante n'a été reçu. Le délai d'expiration minimum autorisé est 60 secondes. La valeur maximale du délai d'attente est de 2 jours pour les groupes de multidiffusion de classe B et de 18 heures pour les groupes de multidiffusion de classe C.

2. Configurez votre message de liaison descendante

Pour configurer votre message de liaison descendante, spécifiez les paramètres suivants :

- Débit de données : choisissez un débit de données pour votre message de liaison descendante. Le débit de données dépend de la région RF et de la taille de la charge utile. Le débit de données par défaut est de 8 pour la région US915 et de 0 pour la région EU868.
- Fréquence : choisissez une fréquence d'envoi de votre message de liaison descendante. Pour éviter les conflits de messagerie, choisissez une fréquence disponible en fonction de la région RF.
- FPort : choisissez un port de fréquence disponible pour envoyer le message de liaison descendante à vos appareils.
- Charge utile : Spécifiez la taille maximale de votre charge utile en fonction du débit de données. En utilisant le débit de données par défaut, vous pouvez avoir une taille de charge utile maximale de 33 octets dans l'US915 RfRegion et de 51 octets dans l'EU868 RfRegion. En utilisant des débits de données plus élevés, vous pouvez transférer jusqu'à une taille de charge utile maximale de 242 octets.

Pour planifier votre message de liaison descendante, choisissez Planifier.

Programmez un message de lien descendant à l'aide de l'API

Pour planifier un message de liaison descendante à l'aide de l'API, utilisez l'opération [StartMulticastGroupSession](#) API ou la commande [start-multicast-group-session](#) CLI. Vous pouvez obtenir des informations sur le groupe de multidiffusion à l'aide de l'opération [d'GetMulticastGroupSession](#) API ou de la commande [get-multicast-group-session](#) CLI.

Pour envoyer des données à un groupe de multidiffusion après le démarrage de la session, utilisez l'opération [SendDataToMulticastGroupAPI](#) ou la commande [send-data-to-multicast-groupCLI](#).

Étapes suivantes

Une fois que vous avez configuré un message de liaison descendante à envoyer aux appareils, le message est envoyé au début de la session. Les appareils d'un groupe de multidiffusion ne peuvent pas confirmer si le message a été reçu.

Configurer des messages de liaison descendante supplémentaires

Vous pouvez également configurer des messages de liaison descendante supplémentaires à envoyer aux appareils de votre groupe de multidiffusion :

- Pour configurer des messages de liaison descendante supplémentaires depuis la console, procédez comme suit :
 1. Accédez à la page [Groupes de multidiffusion](#) de la AWS IoT console et choisissez le groupe de multidiffusion que vous avez créé.
 2. Sur la page de détails du groupe de multidiffusion, choisissez Planifier un message de liaison descendante, puis choisissez Configurer un message de liaison descendante supplémentaire.
 3. Spécifiez les paramètres Débit de données, Fréquence, FPort et Payload, de la même manière que vous avez configuré ces paramètres pour votre premier message de liaison descendante.
- Pour configurer des messages de liaison descendante supplémentaires à l'aide de l'API ou de la CLI,appelez l'opération d'[SendDataToMulticastGroupAPI](#) ou la commande [send-data-to-multicast-groupCLI](#) pour chaque message de liaison descendante supplémentaire.

Mettre à jour le calendrier des sessions

Vous pouvez également mettre à jour le calendrier des sessions afin d'utiliser une nouvelle date et une nouvelle heure de début pour votre session de multidiffusion. Le nouveau calendrier de session remplacera la session précédemment planifiée.

Note

Mettez à jour votre session de multidiffusion uniquement lorsque cela est nécessaire. Ces mises à jour peuvent provoquer le réveil prolongé d'un groupe d'appareils et épuiser la batterie.

- Pour mettre à jour le calendrier des sessions depuis la console, procédez comme suit :
 1. Accédez à la page [Groupes de multidiffusion](#) de la AWS IoT console et choisissez le groupe de multidiffusion que vous avez créé.
 2. Sur la page de détails du groupe de multidiffusion, choisissez Planifier un message de liaison descendante, puis choisissez Mettre à jour le calendrier de la session.
 3. Spécifiez les paramètres Date d'état, Heure de début et Délai d'expiration de la session, de la même manière que vous avez spécifié ces paramètres pour votre premier message de liaison descendante.
- Pour mettre à jour le calendrier de session à partir de l'API ou de la CLI, utilisez l'opération d'[StartMulticastGroupSessionAPI](#) ou la commande [start-multicast-group-sessionCLI](#).

Mises à jour du micrologiciel en direct (FUOTA) pour lesAWS IoT Core for LoRaWAN appareils

Utilisez les mises à jour du microprogramme par voie hertzienne (FUOTA) pour déployer des mises à jour du microprogrammeAWS IoT Core for LoRaWAN sur les appareils.

À l'aide de FUOTA, vous pouvez envoyer des mises à jour du micrologiciel à des appareils individuels ou à un groupe d'appareils. Vous pouvez également envoyer des mises à jour du microprogramme à plusieurs appareils en créant un groupe de multidiffusion. Ajoutez d'abord vos appareils au groupe de multidiffusion, puis envoyez l'image de mise à jour du micrologiciel à tous ces appareils. Nous vous recommandons de signer numériquement les images du microprogramme afin que les appareils qui reçoivent les images puissent vérifier qu'elles proviennent de la bonne source.

AvecAWS IoT Core for LoRaWAN les mises à jour de FUOTA, vous pouvez :

- Déployez de nouvelles images de microprogramme ou des images delta sur un seul appareil ou un groupe de périphériques.
- Vérifier l'authenticité et l'intégrité du nouveau microprogramme après qu'il a été déployé sur les appareils.
- Surveillez la progression d'un déploiement et résolvez les problèmes en cas d'échec du déploiement.

AWS IoT Core for LoRaWANLa prise en charge du FUOTA et des groupes de multidiffusion repose sur les spécifications suivantes [deLoRa l'Alliance](#) :

- LoRaSpécification de configuration de la multidiffusion à distance WAN, TS005-2.0.0
- LoRaSpécification de transport par blocs de données fragmentés sur un réseau WAN, TS004-2.0.0
- LoRaSpécification de synchronisation de l'horloge de la couche d'application WAN, TS003-2.0.0

Note

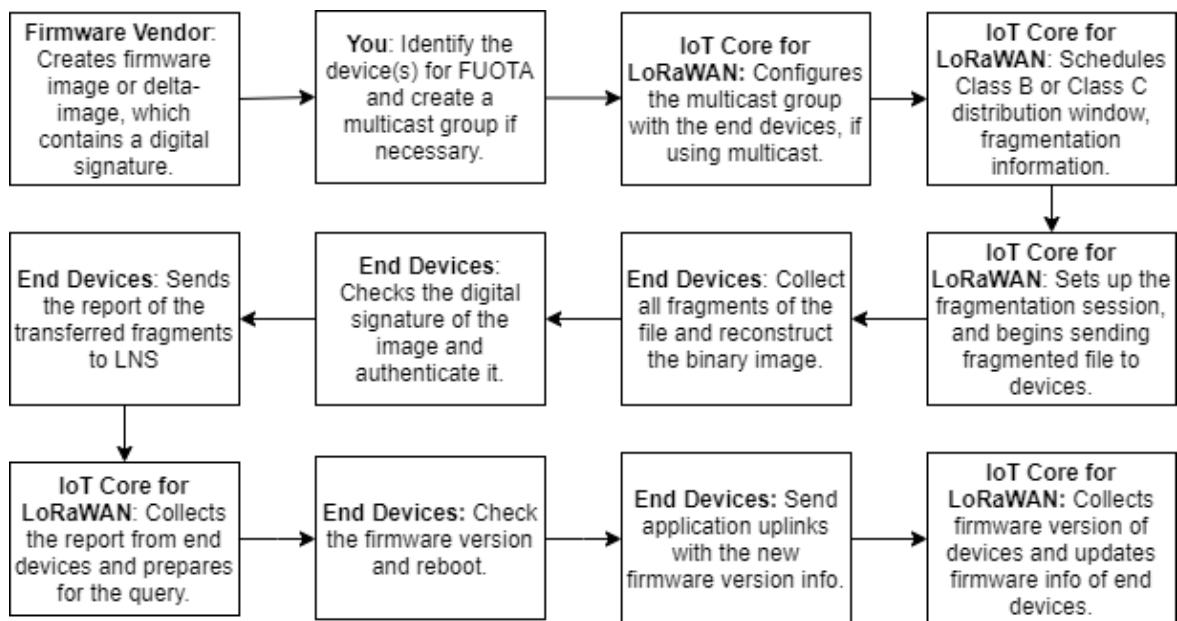
AWS IoT Core for LoRaWANeffectue automatiquement la synchronisation de l'horloge conformément aux spécifications de l' LoRa Alliance. Il utilise cette fonctionAppTimeReq pour répondre à l'heure côté serveur aux appareils qui le demandent à l'aide de la ClockSync signalisation.

L'exemple suivant indique comment effectuer les mises à jour FUOTA.

- [Présentation du processus FUOTA \(p. 1364\)](#)
- [Création d'une tâche FUOTA et fourniture d'une image du microprogramme \(p. 1366\)](#)
- [Ajouter des appareils et des groupes de multidiffusion à une tâche FUOTA et planifier une session FUOTA \(p. 1369\)](#)
- [Surveillez et résolvez les problèmes liés à l'état de votre tâche FUOTA et aux appareils ajoutés à la tâche \(p. 1372\)](#)

Présentation du processus FUOTA

Le schéma suivant montre commentAWS IoT Core for LoRaWAN fonctionne le processus FUOTA pour vos appareils finaux. Si vous ajoutez des appareils individuels à votre session FUOTA, vous pouvez ignorer les étapes de création et de configuration de votre groupe de multidiffusion. Vous pouvez ajouter vos appareils directement à une session FUOTA, puisAWS IoT Core for LoRaWAN lancer le processus de mise à jour du micrologiciel.



Pour effectuer des mises à jour FUOTA pour vos appareils, créez d'abord votre image de microprogramme signée numériquement et configurez les appareils et les groupes de multidiffusion que vous souhaitez ajouter à votre tâche FUOTA. Une fois que vous avez démarré une session FUOTA, vos terminaux collectent tous les fragments, reconstruisent l'image à partir des fragments, signalent l'état à AWS IoT Core for LoRaWAN, puis appliquent la nouvelle image du microprogramme.

Ce qui suit illustre les différentes étapes du processus FUOTA :

- Création d'une image de microprogramme ou d'une image delta avec une signature numérique

Pour effectuer des mises à jour AWS IoT Core for LoRaWAN pour vos appareils LoRaWAN, nous vous recommandons de signer numériquement l'image du microprogramme ou l'image delta lorsque vous envoyez des mises à jour du microprogramme par liaison radio. Les appareils qui reçoivent les images peuvent ensuite vérifier qu'elles proviennent de la bonne source.

La taille de l'image de votre microprogramme ne doit pas dépasser 1 mégaoctet. Plus la taille de votre microprogramme est grande, plus le processus de mise à jour peut prendre du temps. Pour un transfert de données plus rapide ou si la taille de votre nouvelle image est supérieure à 1 mégaoctet, utilisez une image delta, c'est-à-dire la partie de votre nouvelle image qui correspond au delta entre l'image de votre nouveau microprogramme et l'image précédente.

Note

AWS IoT Core for LoRaWAN ne fournit pas l'outil de génération de signature numérique ni le système de gestion des versions du microprogramme. Vous pouvez utiliser n'importe quel outil tiers pour générer la signature numérique de l'image de votre microprogramme. Nous vous recommandons d'utiliser un outil de signature numérique tel que celui intégré au [GitHub référentiel ARM Mbed](#), qui inclut également des outils permettant de générer l'image delta et permettant aux appareils d'utiliser cette image.

- Identifier et configurer les appareils pour FUOTA

Après avoir identifié les appareils pour FUOTA, envoyez les mises à jour du micrologiciel à un ou plusieurs appareils.

- Pour envoyer les mises à jour du microprogramme à plusieurs appareils, créez un groupe de multidiffusion et configurez le groupe de multidiffusion avec les appareils finaux. Pour plus

d'informations, veuillez consulter [Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils \(p. 1352\)](#).

- Pour envoyer des mises à jour du micrologiciel à des appareils individuels, ajoutez ces appareils à votre session FUOTA, puis effectuez la mise à jour du micrologiciel.
3. Planifier une fenêtre de distribution et configurer une session de fragmentation

Si vous avez créé un groupe de multidiffusion, vous pouvez spécifier la fenêtre de distribution de classe B ou de classe C afin de déterminer à quel moment les appareils peuvent recevoir les fragments AWS IoT Core for LoRaWAN. Vos appareils fonctionnaient peut-être en classe A avant de passer en mode classe B ou classe C. Vous devez également spécifier l'heure de début de la session.

Les périphériques de classe B ou de classe C se réveillent à la fenêtre de distribution spécifiée et commencent à recevoir les paquets de liaison descendante. Les appareils fonctionnant en mode de classe C peuvent consommer plus d'énergie que les appareils de classe B. Pour plus d'informations, veuillez consulter [Classes d'appareil \(p. 1338\)](#).

4. Les terminaux signalent l'état AWS IoT Core for LoRaWAN et mettent à jour l'image du microprogramme

Après avoir configuré une session de fragmentation, mettez fin à vos appareils et AWS IoT Core for LoRaWAN effectuez les étapes suivantes pour mettre à jour le micrologiciel de vos appareils.

1. Les périphériques LoRa WAN ayant un faible débit de données, pour démarrer le processus FUOTA, AWS IoT Core for LoRaWAN configure une session de fragmentation afin de fragmenter l'image du microprogramme. Il envoie ensuite ces fragments aux appareils finaux.
2. Après avoir AWS IoT Core for LoRaWAN envoyé les fragments d'image, vos terminaux LoRa WAN effectuent les tâches suivantes.
 - a. Collectez les fragments, puis reconstruisez l'image binaire à partir de ces fragments.
 - b. Vérifiez la signature numérique de l'image reconstruite pour authentifier l'image et vérifier qu'elle provient de la bonne source.
 - c. Comparez la version du micrologiciel AWS IoT Core for LoRaWAN à la version actuelle.
 - d. Signalez l'état des images fragmentées vers lesquelles vous avez transféré AWS IoT Core for LoRaWAN, puis appliquez la nouvelle image du microprogramme.

Note

Dans certains cas, les terminaux signalent l'état des images fragmentées vers lesquelles elles ont été transférées AWS IoT Core for LoRaWAN avant de vérifier la signature numérique de l'image du microprogramme.

Maintenant que vous connaissez le processus FUOTA, vous pouvez créer votre tâche FUOTA et ajouter des appareils à la tâche pour mettre à jour leur micrologiciel. Pour plus d'informations, veuillez consulter [Création d'une tâche FUOTA et fourniture d'une image du microprogramme \(p. 1366\)](#).

Création d'une tâche FUOTA et fourniture d'une image du microprogramme

Pour mettre à jour le microprogramme de vos périphériques LoRa WAN, créez d'abord une tâche FUOTA et fournissez l'image du microprogramme signée numériquement que vous souhaitez utiliser pour la mise à jour. Vous pouvez ensuite ajouter vos appareils et vos groupes de multidiffusion à la tâche et planifier une session FUOTA. Lorsque la session démarre, AWS IoT Core for LoRaWAN configure une session de fragmentation et vos appareils finaux collectent les fragments, reconstruisent l'image et appliquent le nouveau microprogramme. Pour de plus amples informations sur le processus FUOTA, veuillez consulter [Présentation du processus FUOTA \(p. 1364\)](#).

Ce qui suit montre comment créer une tâche FUOTA et télécharger l'image du microprogramme ou l'image delta que vous allez stocker dans un compartiment S3.

Prérequis

Avant de pouvoir effectuer des mises à jour FUOTA, l'image du microprogramme doit être signée numériquement afin que vos appareils puissent vérifier l'authenticité de l'image lors de l'application de l'image. Vous pouvez utiliser n'importe quel outil tiers pour générer la signature numérique de l'image de votre microprogramme. Nous vous recommandons d'utiliser un outil de signature numérique tel que celui intégré au [GitHub référentiel ARM Mbed](#), qui inclut également des outils permettant de générer l'image delta et permettant aux appareils d'utiliser cette image.

Créez une tâche FUOTA et téléchargez l'image du microprogramme à l'aide de la console

Pour créer une tâche FUOTA et télécharger l'image de votre microprogramme à l'aide de la console, accédez à l'onglet des [tâches FUOTA](#) de la console, puis choisissez Créer une tâche FUOTA.

1. Création d'une tâche FUOTA

Pour créer votre tâche FUOTA, spécifiez les propriétés et les balises de la tâche.

1. Spécifier les propriétés d'une tâche FUOTA

Pour spécifier les propriétés de la tâche FUOTA, entrez les informations suivantes pour votre tâche FUOTA.

- Nom : saisissez un nom unique pour votre tâche FUOTA. Le nom ne doit contenir que des lettres, des chiffres, des traits d'union et des traits de soulignement. Il ne doit pas contenir d'espace.
- Description : Vous pouvez fournir une description facultative pour votre groupe de multidiffusion. Le champ de description peut contenir jusqu'à 2 048 caractères
- RFRegion : définissez la bande de fréquences pour votre tâche FUOTA. La bande de fréquences doit correspondre à celle que vous avez utilisée pour approvisionner vos appareils sans fil ou vos groupes de multidiffusion.

2. Tags pour la tâche FUOTA

Vous pouvez éventuellement fournir des paires clé-valeur sous forme de balises pour votre tâche FUOTA. Pour continuer à créer votre tâche, choisissez Suivant.

2. Télécharger l'image du microprogramme

Choisissez le fichier image du microprogramme que vous souhaitez utiliser pour mettre à jour le micrologiciel des appareils que vous ajoutez à la tâche FUOTA. Le fichier image du microprogramme est stocké dans un compartiment S3. Vous pouvez fournir AWS IoT Core for LoRaWAN les autorisations nécessaires pour accéder à l'image du microprogramme en votre nom. Nous vous recommandons de signer numériquement les images du microprogramme afin que son authenticité soit vérifiée lors de la mise à jour du micrologiciel.

1. Choisissez le fichier image du microprogramme

Vous pouvez soit télécharger un nouveau fichier image de microprogramme dans un compartiment S3, soit choisir une image existante qui a déjà été chargée dans un compartiment S3.

Note

La taille du fichier image du microprogramme ne doit pas dépasser 1 mégaoctet. Plus la taille de votre microprogramme est grande, plus le processus de mise à jour peut prendre du temps.

- Pour utiliser une image existante, choisissez Sélectionner une image de microprogramme existante, choisissez Browse S3, puis choisissez le fichier image du microprogramme que vous souhaitez utiliser.

AWS IoT Core for LoRaWAN renseigne l'URL S3, qui est le chemin d'accès au fichier image de votre microprogramme dans le compartiment S3. Le format du chemin est `s3://bucket_name/file_name`. Pour afficher le fichier dans la console [Amazon Simple Storage Service](#), choisissez Afficher.

- Pour charger une nouvelle image de microprogramme.
 - a. Choisissez Charger une nouvelle image de microprogramme et chargez l'image de votre microprogramme. La taille du fichier image ne doit pas dépasser 1 mégaoctet.
 - b. Pour créer un compartiment S3 et saisir un nom de compartiment pour stocker le fichier image de votre microprogramme, choisissez Créer un compartiment S3.

2. Autorisations d'accès au compartiment

Vous pouvez créer un nouveau rôle de service ou choisir un rôle existant pour autoriser l'accès AWS IoT Core for LoRaWAN au fichier image du microprogramme dans le compartiment S3 en votre nom. Choisissez Suivant.

Pour créer un nouveau rôle, vous pouvez saisir un nom de rôle ou le laisser vide pour qu'un nom aléatoire soit généré automatiquement. Pour consulter les autorisations de politique qui accordent l'accès au compartiment S3, choisissez Afficher les autorisations de politique.

Note

Si le chiffrement du compartiment S3 est spécifié comme SSE-KMS, vous devez ajouter le rôle de service utilisé par la tâche FUOTA en tant qu'utilisateur de cette clé KMS. Ce rôle de service est soit généré par la tâche FUOTA, soit attribué à la tâche FUOTA lors de la création de la tâche.

Pour plus d'informations sur l'utilisation d'un compartiment S3 pour stocker votre image et sur l'octroi d'AWS IoT Core for LoRaWAN autorisations pour y accéder, consultez [Téléchargement du fichier du microprogramme dans un compartiment S3 et ajout d'un rôle IAM \(p. 1328\)](#).

3. Vérifier et créer

Pour créer votre tâche FUOTA, passez en revue la tâche FUOTA et les détails de configuration que vous avez spécifiés, puis choisissez Créer une tâche.

Créez une tâche FUOTA et téléchargez l'image du microprogramme à l'aide de l'API

Pour créer une tâche FUOTA et spécifier le fichier image de votre microprogramme à l'aide de l'API, utilisez l'opération [CreateFuotaTask](#) API ou la commande `create-fuota-task` CLI. Vous pouvez fournir `uninput.json` fichier en entrée à la `create-fuota-task` commande. Lorsque vous utilisez l'API ou la CLI, le fichier image du microprogramme que vous fournissez en entrée doit déjà être chargé dans un compartiment S3. Vous spécifiez également le rôle IAM qui donne AWS IoT Core for LoRaWAN accès à l'image du microprogramme dans le compartiment S3.

```
aws iotwireless create-fuota-task \
--cli-input-json file://input.json
```

où :

Contenu du fichier `input.json`

```
{  
    "Description": "FUOTA task to update firmware of devices in multicast group.",  
    "FirmwareUpdateImage": "S3:/firmware_bucket/firmware_image",  
    "FirmwareUpdateRole": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI",  
    "LoRaWAN": {  
        "RfRegion": "US915"  
    },  
    "Name": "FUOTA_Task_MC"  
}
```

Après avoir créé votre tâche FUOTA, vous pouvez utiliser les opérations d'API ou les commandes CLI suivantes pour mettre à jour, supprimer ou obtenir des informations sur votre tâche FUOTA.

- [UpdateFuotaTask](#) ou [update-fuota-task](#)
- [GetFuotaTask](#) ou [get-fuota-task](#)
- [ListFuotaTasks](#) ou [list-fuota-tasks](#)
- [DeleteFuotaTask](#) ou [delete-fuota-task](#)

Étapes suivantes

Maintenant que vous avez créé une tâche FUOTA et fourni l'image du microprogramme, vous pouvez ajouter des appareils à la tâche pour mettre à jour leur microprogramme. Vous pouvez ajouter des appareils individuels ou des groupes de multidiffusion à la tâche. Pour plus d'informations, veuillez consulter [Ajouter des appareils et des groupes de multidiffusion à une tâche FUOTA et planifier une session FUOTA \(p. 1369\)](#).

Ajouter des appareils et des groupes de multidiffusion à une tâche FUOTA et planifier une session FUOTA

Après avoir créé une tâche FUOTA, vous pouvez ajouter à votre tâche des appareils pour lesquels vous souhaitez mettre à jour le micrologiciel. Une fois que vos appareils ont été ajoutés avec succès à la tâche FUOTA, vous pouvez planifier une session FUOTA pour mettre à jour le micrologiciel de l'appareil.

- Si vous ne disposez que d'un petit nombre d'appareils, vous pouvez les ajouter directement à votre tâche FUOTA.
- Si vous souhaitez mettre à jour le microprogramme d'un grand nombre de périphériques, vous pouvez ajouter ces appareils à vos groupes de multidiffusion, puis les ajouter à votre tâche FUOTA. Pour en savoir plus sur la création et l'utilisation de groupes multidiffusion, veuillez consulter [Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils \(p. 1352\)](#).

Note

Vous pouvez ajouter des appareils individuels ou des groupes de multidiffusion à la tâche FUOTA.
Vous ne pouvez pas ajouter à la fois des appareils et des groupes de multidiffusion à la tâche.

Après avoir ajouté vos appareils ou groupes de multidiffusion, vous pouvez démarrer une session de mise à jour du micrologiciel. AWS IoT Core for LoRaWAN collecte l'image du microprogramme, fragmente les images, puis stocke les fragments dans un format crypté. Vos terminaux collectent les fragments et appliquent la nouvelle image du microprogramme. Le temps nécessaire à la mise à jour du micrologiciel dépend de la taille de l'image et de la façon dont les images ont été fragmentées. Une fois la mise à jour du microprogramme terminée, les fragments cryptés de l'image du microprogramme stockés par AWS IoT Core for LoRaWAN sont supprimés. Vous pouvez toujours trouver l'image du microprogramme dans le compartiment S3.

Prérequis

Avant de pouvoir ajouter des appareils ou des groupes de multidiffusion à votre tâche FUOTA, procédez comme suit.

- Vous devez déjà avoir créé la tâche FUOTA et fourni l'image de votre microprogramme. Pour plus d'informations, veuillez consulter [Création d'une tâche FUOTA et fourniture d'une image du microprogramme \(p. 1366\)](#).
- Provisionnez les périphériques sans fil pour lesquels vous souhaitez mettre à jour le microprogramme de l'appareil. Pour en savoir plus sur l'intégration de votre appareil, veuillez consulter [Intégrez vos appareils à AWS IoT Core for LoRaWAN \(p. 1288\)](#).
- Pour mettre à jour le micrologiciel de plusieurs appareils, vous pouvez les ajouter à un groupe de multidiffusion. Pour plus d'informations, veuillez consulter [Créez des groupes de multidiffusion pour envoyer une charge utile en liaison descendante à plusieurs appareils \(p. 1352\)](#).
- Lorsque vous intégrez les appareils à AWS IoT Core for LoRaWAN, spécifiez le paramètre de configuration FUOTAFPorts. Si vous utilisez un périphérique LoRa WAN v1.0.x, vous devez également spécifier leGenAppKey. Pour en savoir plus sur les paramètres de configuration FUOTA, veuillez consulter [Préparer les appareils pour la multidiffusion et la configuration FUOTA \(p. 1353\)](#).

Ajouter des appareils à une tâche FUOTA et planifier une session FUOTA à l'aide de la console

Pour ajouter des appareils ou des groupes de multidiffusion et planifier une session FUOTA à l'aide de la console, accédez à l'onglet des [tâches FUOTA](#) de la console. Choisissez ensuite la tâche FUOTA à laquelle vous souhaitez ajouter des appareils et effectuez la mise à jour du micrologiciel.

Ajouter des appareils et des groupes de multidiffusion

1. Vous pouvez ajouter des appareils individuels ou des groupes de multidiffusion à votre tâche FUOTA. Toutefois, vous ne pouvez pas ajouter à la fois des appareils individuels et des groupes de multidiffusion à la même tâche FUOTA. Pour ajouter des appareils à l'aide de la console par, procédez comme suit.
 1. Dans les détails de la tâche FUOTA, choisissez Ajouter un appareil.
 2. Choisissez la bande de fréquences ou la région RF pour les appareils que vous ajoutez à la tâche. Cette valeur doit correspondre à la RFRegion que vous avez choisie pour la tâche FUOTA.
 3. Choisissez si vous souhaitez ajouter des appareils individuels ou des groupes de multidiffusion à la tâche.
 - Pour ajouter des appareils individuels, choisissez Ajouter des appareils individuels et entrez l'identifiant de chaque appareil que vous souhaitez ajouter à votre tâche FUOTA.
 - Pour ajouter des groupes de multidiffusion, choisissez Ajouter des groupes de multidiffusion et ajoutez vos groupes de multidiffusion à la tâche. Vous pouvez filtrer les groupes de multidiffusion que vous souhaitez ajouter à la tâche à l'aide du profil de l'appareil ou des balises. Lorsque vous filtrez par profil d'appareil, vous pouvez sélectionner des groupes de multidiffusion composés d'appareils dont le profil prend en charge la classe B ou prend en charge la classe C activé.
2. Planifier une session FUOTA

Une fois que vos appareils ou groupes de multidiffusion ont été ajoutés avec succès, vous pouvez planifier une session FUOTA. Pour planifier une session, procédez comme suit.

1. Choisissez la tâche FUOTA pour laquelle vous souhaitez mettre à jour le micrologiciel de l'appareil, puis choisissez Planifier une session FUOTA.
2. Spécifiez une date et une heure de début pour votre session FUOTA. Assurez-vous que l'heure de début est 30 minutes ou plus tard par rapport à l'heure actuelle.

Ajouter des appareils à une tâche FUOTA et planifier une session FUOTA à l'aide de l'API

Vous pouvez utiliser l'AWS IoT WirelessAPI ou la CLI pour ajouter vos appareils sans fil ou vos groupes de multidiffusion à votre tâche FUOTA. Vous pouvez ensuite planifier une session FUOTA.

1. Ajouter des appareils et des groupes de multidiffusion

Vous pouvez associer des appareils sans fil ou des groupes de multidiffusion à votre tâche FUOTA.

- Pour associer des appareils individuels à votre tâche FUOTA, utilisez l'opération [d'AssociateWirelessDeviceWithFuotaTask](#)API ou la commande [associate-wireless-device-with-fuota-task](#)CLI et fournissez-lesWirelessDeviceID en tant qu'entrée.

```
aws iotwireless associate-wireless-device-with-fuota-task \
--id "01a23cde-5678-4a5b-ab1d-33456808ecb2"
--wireless-device-id "ab0c23d3-b001-45ef-6a01-2bc3de4f5333"
```

- Pour associer des groupes de multidiffusion à votre tâche FUOTA, utilisez l'opération [d'AssociateMulticastGroupWithFuotaTask](#)API ou la commande [associate-multicast-group-with-fuota-task](#)CLI et fournissez-lesMulticastGroupID en entrée.

```
aws iotwireless associate-multicast-group-with-FUOTA-task \
--id 01a23cde-5678-4a5b-ab1d-33456808ecb2"
--multicast-group-id
```

Après avoir associé vos appareils sans fil ou votre groupe de multidiffusion à votre tâche FUOTA, utilisez les opérations d'API ou les commandes CLI suivantes pour répertorier vos appareils ou groupes de multidiffusion ou pour les dissocier de votre tâche.

- [DisassociateWirelessDeviceFromFuotaTask](#) ou [disassociate-wireless-device-from-fuota-task](#)
- [DisassociateMulticastGroupFromFuotaTask](#) ou [disassociate-multicast-group-from-fuota-task](#)
- [ListWirelessDevices](#) ou [list-wireless-devices](#)
- [ListMulticastGroups](#) ou [list-multicast-groups-by-fuota-task](#)

Note

L'API :

- [ListWirelessDevices](#) peut répertorier les appareils sans fil en général et les appareils associés à un groupe de multidiffusion, lorsqu'ilMulticastGroupID est utilisé comme filtre. L'API répertorie les appareils sans fil associés à une tâche FUOTA lorsqu'elleFuotaTaskID est utilisée comme filtre.
- [ListMulticastGroups](#) peut répertorier les groupes de multidiffusion en général et les groupes de multidiffusion associés à une tâche FUOTA lorsqu'elleFuotaTaskID est utilisée comme filtre.

2. Planifier une session FUOTA

Une fois que vos appareils ou groupes de multidiffusion ont été ajoutés avec succès à la tâche FUOTA, vous pouvez démarrer une session FUOTA pour mettre à jour le micrologiciel de l'appareil. L'heure de début doit être au moins 30 minutes après l'heure actuelle. Pour planifier une session FUOTA à l'aide de l'API ou de la CLI, utilisez l'opération [StartFuotaTask](#)API ou la commande [start-fuota-task](#)CLI.

Une fois que vous avez démarré une session FUOTA, vous ne pouvez plus ajouter d'appareils ou de groupes de multidiffusion à la tâche. Vous pouvez obtenir des informations sur l'état de votre session FUOTA en utilisant l'opération [GetFuotaTask](#) API ou la commande [get-fuota-task](#) CLI.

Surveillez et résolvez les problèmes liés à l'état de votre tâche FUOTA et aux appareils ajoutés à la tâche

Après avoir provisionné les périphériques sans fil et créé les groupes de multidiffusion que vous souhaitez utiliser, vous pouvez démarrer une session FUOTA en effectuant les étapes suivantes.

État d'une tâche FUOTA

L'un des messages d'état suivants peut s'afficher dans le fichier FUOTA pour votre tâche FUOTAAWS Management Console.

- En suspens

Cet état indique que vous avez créé une tâche FUOTA, mais qu'il n'y a pas encore de session de mise à jour du microprogramme. Ce message d'état s'affichera lorsque votre tâche aura été créée. Pendant ce temps, vous pouvez mettre à jour votre tâche FUOTA et associer ou dissocier des appareils ou des groupes de multidiffusion à votre tâche. Une fois que le statut est passé de En attente, aucun appareil supplémentaire ne peut être ajouté à la tâche.

- Session FUOTA en attente

Une fois que vos appareils ont été ajoutés avec succès à la tâche FUOTA, lorsque votre tâche comporte une session de mise à jour du microprogramme planifiée, ce message d'état s'affiche. Pendant ce temps, vous ne pouvez ni mettre à jour ni ajouter d'appareils à votre session FUOTA. Si vous annulez votre session FUOTA, le statut du groupe passe à En attente.

- En session FUOTA

Lorsque votre session FUOTA commence, ce message d'état s'affiche. La session de fragmentation démarre et vos terminaux collectent les fragments, reconstruisent l'image du microprogramme, comparent la nouvelle version du microprogramme à la version d'origine et appliquent la nouvelle image.

- FUOTA fait

Une fois que vos appareils finaux ont indiquéAWS IoT Core for LoRaWAN que la nouvelle image du microprogramme a été appliquée, ou lorsque la session expire, la session FUOTA est marquée comme terminée et vous verrez ce statut s'afficher.

Cet état s'affichera également dans tous les cas suivants. Assurez-vous donc de vérifier si la mise à jour du micrologiciel a été correctement appliquée aux appareils.

- Lorsque l'état de la tâche FUOTA était « session FUOTA en attente » et qu'une erreur se produit dans le compartiment S3, par exemple, le lien vers le fichier image dans le compartiment S3 est incorrect ouAWS IoT Core for LoRaWAN ne dispose pas des autorisations suffisantes pour accéder au fichier contenu dans le compartiment.
- Lorsque le statut de la tâche FUOTA était « session FUOTA en attente » et qu'une demande de démarrage d'une session FUOTA est reçue, mais qu'aucune réponse n'est reçue des appareils ou des groupes de multidiffusion de votre tâche FUOTA.
- Lorsque l'état de la tâche FUOTA était En session FUOTA et que les appareils ou les groupes de multidiffusion n'ont envoyé aucun fragment depuis un certain temps, ce qui entraîne l'expiration du délai de la session.
- Supprimer l'attente

Si vous supprimez votre tâche FUOTA qui se trouve dans l'un des autres états, ce statut s'affichera. Une action de suppression est permanente et ne peut pas être annulée. Cette action peut prendre du temps et le statut de la tâche sera Supprimer en attendant que la tâche FUOTA ait été supprimée. Une fois que votre tâche FUOTA est entrée dans cet état, elle ne peut pas passer à l'un des autres états.

État des appareils dans une tâche FUOTA

L'un des messages d'état suivants peut s'afficher sur les appareils concernés par votre tâche FUOTA dans leAWS Management Console. Vous pouvez passer la souris sur chaque message d'état pour obtenir plus d'informations sur ce qu'il indique.

- Initial

Lorsque c'est l'heure de début de votre session FUOTA,AWS IoT Core for LoRaWAN vérifie si votre appareil dispose du package compatible pour la mise à jour du micrologiciel. Si votre appareil dispose du package compatible, la session FUOTA de l'appareil démarre. L'image du microprogramme est fragmentée et les fragments sont envoyés à votre appareil. Lorsque cet état s'affiche, cela indique que la session FUOTA de l'appareil n'a pas encore commencé.

- Package non pris en charge

Si l'appareil ne dispose pas du package FUOTA pris en charge, cet état s'affichera. Si le package de mise à jour du microprogramme n'est pas pris en charge, la session FUOTA de votre appareil ne peut pas démarrer. Pour résoudre cette erreur, vérifiez si le microprogramme de votre appareil peut recevoir des mises à jour du microprogramme à l'aide de FUOTA.

- Algorithme de fragmentation non pris en charge

Au début de votre session FUOTA,AWS IoT Core for LoRaWAN configure une session de fragmentation pour votre appareil. Si cet état s'affiche, cela signifie que le type d'algorithme de fragmentation utilisé ne peut pas être appliqué pour la mise à jour du microprogramme de votre appareil. L'erreur se produit parce que votre appareil ne dispose pas du package FUOTA pris en charge. Pour résoudre cette erreur, vérifiez si le microprogramme de votre appareil peut recevoir des mises à jour du microprogramme à l'aide de FUOTA.

- Mémoire insuffisante

Après avoirAWS IoT Core for LoRaWAN envoyé les fragments d'image, vos appareils finaux collectent les fragments d'image et reconstruisent l'image binaire à partir de ces fragments. Cet état s'affiche lorsque votre appareil ne dispose pas de suffisamment de mémoire pour assembler les fragments entrants de l'image du microprogramme, ce qui peut entraîner la fin prémature de votre session de mise à jour du microprogramme. Pour résoudre l'erreur, vérifiez si le matériel de votre appareil peut recevoir cette mise à jour. Si votre appareil ne peut pas recevoir cette mise à jour, utilisez une image delta pour mettre à jour le micrologiciel.

- Indice de fragmentation non pris en charge

L'indice de fragmentation identifie l'une des quatre sessions de fragmentation possibles simultanément. Si votre appareil ne prend pas en charge la valeur d'indice de fragmentation indiquée, cet état s'affiche. Pour résoudre cette erreur, effectuez une ou plusieurs des opérations suivantes.

- Démarrez une nouvelle tâche FUOTA pour l'appareil.
- Si l'erreur persiste, passez du mode monodiffusion au mode multidiffusion.
- Si l'erreur persiste, vérifiez le microprogramme de votre appareil.
- Erreur de mémoire

Cet état indique que votre appareil a rencontré une erreur de mémoire lors de la réception des fragments entrantsAWS IoT Core for LoRaWAN. Si cette erreur se produit, il se peut que votre appareil ne soit pas en mesure de recevoir cette mise à jour. Pour résoudre l'erreur, vérifiez si le matériel de votre appareil

peut recevoir cette mise à jour. Si nécessaire, utilisez une image delta pour mettre à jour le micrologiciel de l'appareil.

- Mauvais descripteur

Votre appareil ne prend pas en charge le descripteur indiqué. Le descripteur est un champ qui décrit le fichier qui sera transporté pendant la session de fragmentation. Si cette erreur s'affiche, contactez le [AWS SupportCentre](#).

- Rediffusion du nombre de sessions

Cet état indique que votre appareil a déjà utilisé ce nombre de sessions. Pour résoudre l'erreur, lancez une nouvelle tâche FUOTA pour l'appareil.

- Fragments manquants

Au fur et à mesure que votre appareil collecte les fragments d'image AWS IoT Core for LoRaWAN, il reconstruit la nouvelle image du microprogramme à partir des fragments codés indépendants. Si votre appareil n'a pas reçu tous les fragments, la nouvelle image ne peut pas être reconstruite et vous verrez ce statut. Pour résoudre l'erreur, lancez une nouvelle tâche FUOTA pour l'appareil.

- Erreur MIC

Lorsque votre appareil reconstruit la nouvelle image du microprogramme à partir des fragments collectés, il effectue un contrôle MIC (Message Integrity Check) pour vérifier l'authenticité de votre image et si elle provient de la bonne source. Si votre appareil détecte une discordance dans le micro après avoir rassemblé les fragments, cet état s'affiche. Pour résoudre l'erreur, lancez une nouvelle tâche FUOTA pour l'appareil.

- Réussite

La session FUOTA pour votre appareil a été réussie.

Note

Bien que ce message d'état indique que les appareils ont reconstruit l'image à partir des fragments et l'ont vérifiée, le microprogramme du périphérique n'a peut-être pas été mis à jour lorsque le périphérique signale l'état à AWS IoT Core for LoRaWAN. Vérifiez si le micrologiciel de votre appareil a été mis à jour.

Étapes suivantes

Vous avez découvert les différents statuts de la tâche FUOTA et de ses appareils et vous savez comment résoudre tout problème. Pour plus d'informations sur chacun de ces états, consultez la [spécification de transport par blocs de données fragmentés sur le réseau LoRa WAN, TS004-1.0.0](#).

Surveillance de votre parc de ressources sans fil en temps réel à l'aide d'un analyseur de réseau

L'analyseur de réseau utilise une WebSocket connexion par défaut pour recevoir des journaux de messages de suivi en temps réel pour vos ressources de connectivité sans fil. À l'aide de l'analyseur de réseau, vous pouvez ajouter les ressources que vous souhaitez surveiller, activer une session de suivi et commencer à recevoir des messages de suivi en temps réel.

Note

Pour surveiller vos ressources, vous pouvez également utiliser Amazon CloudWatch. Pour l'utiliser avec CloudWatch, vous configurez un rôle IAM pour configurer la journalisation, puis vous attendez que les entrées du journal s'affichent dans la console. Pour plus d'informations,

veuillez consulter [Surveillance et journalisation pour AWS IoT Wireless l'utilisation d'Amazon CloudWatch \(p. 1452\)](#).

L'analyseur de réseau réduit considérablement le temps nécessaire pour établir une connexion et commencer à recevoir des messages de suivi, en vous fournissant des informations de just-in-time journal pour votre parc de ressources. En réduisant le temps de configuration et en utilisant les informations des messages de suivi, vous pouvez surveiller vos ressources plus efficacement, obtenir des informations pertinentes et résoudre les erreurs. Les ressources sans fil que vous pouvez surveiller incluent les périphériques LoRa WAN, les passerelles LoRa WAN et les groupes de multidiffusion. Par exemple, vous pouvez rapidement identifier une erreur de jointure lors de l'intégration de l'un de vos appareils LoRa WAN. Pour corriger l'erreur, utilisez les informations du journal des messages de suivi fourni.

Les ressources sans fil que vous pouvez surveiller incluent les périphériques LoRa WAN, les passerelles LoRa WAN et les groupes de multidiffusion. Vous pouvez également utiliser l'analyseur de réseau pour déboguer et résoudre tout problème lié à votre tâche FUOTA. Par exemple, considérez une tâche FUOTA qui possède un groupe de multidiffusion, le groupe de multidiffusion A, et auquel les appareils A et B ont été ajoutés à ce groupe. Vous pouvez ensuite ajouter à la fois le groupe de multidiffusion, le groupe de multidiffusion A et les appareils A et B à la configuration de votre analyseur de réseau pour les surveiller et déboguer votre tâche FUOTA.

Comment utiliser l'analyseur de réseau

Pour surveiller votre parc de ressources et commencer à recevoir des messages de suivi, effectuez les opérations suivantes.

1. Création de la configuration de l'analyseur de réseau et ajout de ressources

Avant de pouvoir activer la messagerie de suivi, créez une configuration d'analyseur de réseau et ajoutez des ressources à votre configuration. Tout d'abord, spécifiez les paramètres de configuration, qui incluent les niveaux de journalisation et les informations relatives aux périphériques sans fil et aux trames de multidiffusion. Ajoutez ensuite les ressources sans fil que vous souhaitez surveiller à l'aide de la passerelle sans fil, du périphérique sans fil et des identifiants de groupe de multidiffusion.

2. Diffusez des messages de suivi avec WebSockets

Vous pouvez générer une URL de demande présignée à l'aide des informations d'identification de votre rôle IAM afin de diffuser les messages de suivi de l'analyseur de réseau à l'aide du protocole WebSocket

3. Activez la session de messagerie de suivi et surveillez les messages de suivi

Pour commencer à recevoir des messages de suivi, activez votre session de suivi. Pour éviter des coûts supplémentaires, vous pouvez désactiver ou fermer la session de suivi de votre analyseur de réseau.

La section suivante montre comment créer votre configuration, ajouter des ressources et activer votre session de messagerie de suivi.

Rubriques

- [Ajouter le rôle IAM nécessaire pour l'analyseur de réseau \(p. 1376\)](#)
- [Créer une configuration d'analyseur de réseau et ajouter des ressources \(p. 1377\)](#)
- [Diffusez les messages de suivi de l'analyseur de réseau avec WebSockets \(p. 1384\)](#)
- [Afficher et surveiller les journaux des messages de suivi de l'analyseur de réseau en temps réel \(p. 1393\)](#)
- [Déboguez et dépannez vos groupes de multidiffusion et vos tâches FUOTA à l'aide d'un analyseur de réseau \(p. 1396\)](#)

Ajouter le rôle IAM nécessaire pour l'analyseur de réseau

Lorsque vous utilisez l'analyseur de réseau, vous devez accorder à un utilisateur l'autorisation d'utiliser les opérations de l'API [UpdateNetworkAnalyzerConfiguration](#) et d'accéder [GetNetworkAnalyzerConfiguration](#) aux ressources de l'analyseur de réseau. Vous trouverez ci-dessous les politiques IAM que vous utilisez pour accorder des autorisations.

Politiques IAM pour l'analyseur de réseau

Utilisez chacune des actions suivantes :

- Politique d'accès sans fil complet

Accordez AWS IoT Core au LoRa WAN la politique d'accès complète en attachant la politique [AWSIoTWirelessFullAccess](#) à votre rôle. Pour en savoir plus, consultez le [résumé AWSIoTWirelessFullAccess de la politique](#).

- Politique IAM étendue pour l'API Get and Update

Créez la politique IAM suivante en accédant à la page [Créer une politique](#) de la console IAM et en accédant à l'onglet Éditeur visuel :

1. Choisissez IoTWireless for Service.
2. Sous Niveau d'accès, ouvrez Lire et choisissez GetNetworkAnalyzerConfiguration, puis cliquez sur Écrire et choisissez UpdateNetworkAnalyzerConfiguration.
3. Choisissez Next:Tags et entrez un nom pour la politique, tel que IoT.WirelessNetworkAnalyzerPolicy. Choisissez Create Policy (Créer une politique).

Ce qui suit montre la politique IoT WirelessNetworkAnalyzerPolicy que vous avez créée. Pour en savoir plus sur la création d'une stratégie, consultez [Création de stratégies IAM](#).

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "iotwireless:GetNetworkAnalyzerConfiguration",  
                "iotwireless:UpdateNetworkAnalyzerConfiguration"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Politique étendue pour accéder à des ressources spécifiques

Pour configurer un contrôle d'accès plus précis, vous devez ajouter les passerelles sans fil, les appareils et tous les groupes de multidiffusion au champ Ressource. La politique suivante utilise l'ARN générique pour accorder l'accès à toutes les passerelles, appareils et groupes de multidiffusion. Vous pouvez contrôler l'accès à des passerelles et à des appareils spécifiques en utilisant le `WirelessGatewayId`, `WirelessDeviceId`, et `leMulticastGroupId`.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "VisualEditor0",  
            "Effect": "Allow",  
            "Action": [  
                "iotwireless:GetNetworkAnalyzerConfiguration",  
                "iotwireless:UpdateNetworkAnalyzerConfiguration"  
            ],  
            "Resource": "arn:aws:iot::WirelessGatewayId:WirelessDeviceId:leMulticastGroupId"  
        }  
    ]  
}
```

```
"Statement": [
    {
        "Sid": "VisualEditor0",
        "Effect": "Allow",
        "Action": [
            "iotwireless:GetNetworkAnalyzerConfiguration",
            "iotwireless:UpdateNetworkAnalyzerConfiguration"
        ],
        "Resource": [
            "arn:aws:iotwireless:*:{accountId}:WirelessDevice/*",
            "arn:aws:iotwireless:*:{accountId}:WirelessGateway/*",
            "arn:aws:iotwireless:*:{accountId}:MulticastGroup/*",
            "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
        ]
    }
]
```

Pour accorder à un utilisateur l'autorisation d'utiliser l'analyseur de réseau sans toutefois utiliser de ressources sans fil ni de groupes de multidiffusion, appliquez la politique suivante. Sauf indication contraire, les autorisations d'utilisation des ressources sont implicitement refusées.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "iotwireless:GetNetworkAnalyzerConfiguration",
                "iotwireless:UpdateNetworkAnalyzerConfiguration"
            ],
            "Resource": [
                "arn:aws:iotwireless:*:{accountId}:NetworkAnalyzerConfiguration/*"
            ]
        }
    ]
}
```

Étapes suivantes

Maintenant que vous avez créé la politique, vous pouvez ajouter des ressources à la configuration de votre analyseur de réseau et recevoir des informations de suivi pour ces ressources. Pour plus d'informations, veuillez consulter [Créer une configuration d'analyseur de réseau et ajouter des ressources \(p. 1377\)](#).

Créer une configuration d'analyseur de réseau et ajouter des ressources

Avant de diffuser des messages de suivi, créez une configuration d'analyseur de réseau et ajoutez-y les ressources que vous souhaitez surveiller. Lorsque vous créez une configuration, vous pouvez :

- Spécifiez un nom de configuration et, éventuellement, une description.
- Personnalisez les paramètres de configuration tels que les informations sur le cadre et le niveau de détail de vos messages de journal.
- Ajoutez les ressources que vous souhaitez contrôler. Les ressources peuvent inclure des appareils sans fil, des passerelles sans fil et des groupes de multidiffusion.

Les paramètres de configuration que vous spécifiez détermineront les informations de messagerie de suivi que vous recevrez pour les ressources que vous ajoutez à la configuration. Vous pouvez également créer plusieurs configurations en fonction de votre cas d'utilisation en matière de surveillance.

L'exemple suivant montre comment créer une configuration et ajouter des ressources.

Rubriques

- [Créer une configuration d'analyseur de réseau \(p. 1378\)](#)
- [Ajouter des ressources et mettre à jour la configuration de l'analyseur de réseau \(p. 1381\)](#)

Créer une configuration d'analyseur de réseau

Avant de pouvoir surveiller vos ressources sans fil ou vos groupes de multidiffusion, vous devez créer une configuration d'analyseur de réseau. Lorsque vous créez la configuration, vous devez uniquement spécifier un nom de configuration. Vous pouvez personnaliser vos paramètres de configuration et ajouter les ressources que vous souhaitez surveiller à votre configuration, même après sa création. Les paramètres de configuration déterminent les informations de messagerie de suivi que vous recevrez pour ces ressources.

Selon les ressources que vous souhaitez surveiller et le niveau d'informations que vous souhaitez recevoir à leur sujet, vous pouvez créer plusieurs configurations. Par exemple, vous pouvez créer une configuration qui affiche uniquement les informations d'erreur pour un ensemble de passerelles de votreCompte AWS. Vous pouvez également créer une configuration qui affiche toutes les informations relatives à un périphérique sans fil que vous souhaitez surveiller.

Les sections suivantes présentent les différents paramètres de configuration et expliquent comment créer votre configuration.

Paramètres configuration

Lors de la création ou de la mise à jour de la configuration de votre analyseur de réseau, vous pouvez également personnaliser les paramètres suivants pour filtrer les informations du flux de log.

- Informations sur le cadre

Ce paramètre contient les informations relatives au cadre des ressources de votre appareil sans fil pour les messages de suivi. Les informations du cadre peuvent être utilisées pour déboguer la communication entre votre serveur réseau et les appareils finaux. Il est activé par défaut.

- Niveaux de journalisation

Vous pouvez consulter les journaux d'informations ou d'erreurs, ou vous pouvez désactiver la journalisation.

- Infos

Les journaux dont le niveau de journalisation est Info sont plus détaillés et contiennent à la fois des flux de journaux d'erreurs et des flux de journaux d'informations. Les journaux d'informations peuvent être utilisés pour visualiser les modifications apportées à l'état d'un appareil ou d'une passerelle.

Note

La collecte de flux de journaux plus détaillés peut entraîner des coûts supplémentaires. Pour de plus amples informations sur la tarification, veuillez consulter [Tarification AWS IoT Core](#).

- Erreur

Les journaux dont le niveau d'erreur est défini sur Error sont moins détaillés et affichent uniquement les informations relatives aux erreurs. Vous pouvez utiliser ces journaux lorsqu'une application présente une erreur, telle qu'une erreur de connexion à un appareil. En utilisant les informations du flux de log, vous pouvez identifier et résoudre les erreurs relatives aux ressources de votre flotte.

- Informations sur la trame multicast

Si vous souhaitez surveiller des groupes de multidiffusion, vous pouvez utiliser les informations de la trame de multidiffusion pour dépanner les appareils qui tentent de rejoindre le groupe.

Créer une configuration à l'aide de la console

Vous pouvez créer une configuration d'analyseur de réseau et personnaliser les paramètres facultatifs à l'aide de la AWS IoT console ou de l'API AWS IoT sans fil. Vous pouvez également créer plusieurs configurations et supprimer ultérieurement celles que vous n'utilisez plus.

Créer une configuration d'analyseur de réseau

1. Ouvrez le [hub Network Analyzer de la AWS IoT console](#) et choisissez Create configuration.

2. Spécifiez les paramètres configuration.

- Nom, description et identifications

Spécifiez un nom de configuration unique, une description facultative et des balises pour ajouter des paires clé-valeur de métadonnées relatives à la configuration. Pour en savoir plus sur la dénomination et la description de vos ressources, consultez [Décrivez vos ressourcesAWS IoT Core pour le LoRa WAN \(p. 1278\)](#)

- Paramètres configuration

Choisissez si vous souhaitez désactiver les informations sur les cadres et utilisez Sélectionner les niveaux de journal pour sélectionner les niveaux de journal que vous souhaitez utiliser pour vos journaux de messages de suivi. Si vous souhaitez surveiller des groupes de multidiffusion, choisissez Message de données de multidiffusion (groupes de multidiffusion uniquement). Choisissez Suivant.

3. Ajoutez des ressources à votre configuration. Vous pouvez soit ajouter vos ressources maintenant, soit choisir Créer, puis ajouter vos ressources ultérieurement. Pour ajouter des ressources ultérieurement, choisissez Create.

Sur la page du hub Network Analyzer, vous verrez la configuration que vous avez créée ainsi que ses paramètres. Pour afficher les détails de la nouvelle configuration, choisissez le nom de la configuration.

Supprimer la configuration de votre analyseur de réseau

Vous pouvez créer plusieurs configurations d'analyseur de réseau en fonction des ressources que vous souhaitez surveiller et du niveau d'informations de suivi que vous souhaitez recevoir à leur sujet.

Pour supprimer des configurations depuis la console

1. Accédez au [hub Network Analyzer de la AWS IoT console](#) et choisissez la configuration que vous souhaitez supprimer.
2. Choisissez Actions, puis Delete (Supprimer).

Création d'une configuration à l'aide de l'API

Pour créer une configuration d'analyseur de réseau à l'aide de l'API, utilisez l'opération [CreateNetworkAnalyzerConfiguration](#)API ou la [create-network-analyzer-configuration](#)commande CLI.

Lorsque vous créez votre configuration, vous devez uniquement spécifier un nom de configuration. Vous pouvez également utiliser cette opération d'API pour spécifier les paramètres de configuration et ajouter des ressources lors de la création de la configuration.

Note

Lorsque vous utilisez l'opération d'`CreateNetworkAnalyzerConfigurationAPI` pour ajouter des ressources, vous ne pouvez spécifier que 99 ressources sans fil au maximum pour chaque demande d'API. Une configuration d'analyseur de réseau unique peut comporter jusqu'à 250 appareils sans fil et 250 passerelles sans fil combinés. Pour ajouter des ressources supplémentaires, utilisez la AWS IoT console, comme décrit dans la section ci-dessus.

- Créez une configuration

Lorsque vous créez votre configuration, vous devez spécifier un nom. Par exemple, la commande suivante crée une configuration en fournissant uniquement un nom et, éventuellement, une description. Par défaut, les informations de cadre sont activées dans la configuration et utilisent un niveau de journalisation de `INFO`.

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_Network_Analyzer_Config \
    --description "My first network analyzer configuration"
```

L'exécution de cette commande affiche l'ARN et l'ID de la configuration de votre analyseur de réseau.

```
{  
    "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

- Créez une configuration avec des ressources

Pour personnaliser les paramètres de configuration, utilisez le `trace-content` paramètre. Pour ajouter des ressources, utilisez les `MulticastGroups` paramètres `WirelessDevices`/`WirelessGateways`, et pour spécifier les passerelles, les appareils et les groupes de multidiffusion que vous souhaitez ajouter à votre configuration. Par exemple, la commande suivante permet de personnaliser les paramètres de configuration et d'ajouter à votre configuration les ressources sans fil.

```
aws iotwireless create-network-analyzer-configuration \
    --configuration-name My_NetworkAnalyzer_Config \
    --trace-content
    WirelessDeviceFrameInfo=DISABLED,MulticastFrameInfo=ENABLED,LogLevel="ERROR" \
    --wireless-gateways "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-de1f-2b3b-4c5c-  
bb1112223cd1"
    --wireless-devices "1ffd32c8-8130-4194-96df-622f072a315f" \
    --multicast-groups "12abd34e-5f67-89c2-9293-593b1bd862e0"
```

L'exemple suivant affiche le résultat de l'exécution de la commande :

```
{  
    "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Répertorier les configurations de l'analyseur réseau

Vous pouvez créer plusieurs configurations d'analyseur de réseau en fonction des ressources que vous souhaitez surveiller et du niveau de détail des informations de messagerie de suivi que vous souhaitez recevoir pour ces ressources. Après avoir créé ces configurations, vous pouvez utiliser l'opération

d'[ListNetworkAnalyzerConfigurations](#)API ou la commande [list-network-analyzer-configuration](#)CLI pour obtenir une liste de ces configurations.

```
aws iotwireless list-network-analyzer-configurations
```

L'exécution de cette commande affiche toutes les configurations de l'analyseur de réseau de votre Compte AWS. Vous pouvez également utiliser le `max-results` paramètre pour spécifier le nombre de configurations que vous souhaitez afficher. L'exemple suivant affiche le résultat de l'exécution de cette commande.

```
{  
    "NetworkAnalyzerConfigurationList": [  
        {  
            "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:NetworkAnalyzerConfiguration/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
            "Name": "My_Network_Analyzer_Config1"  
        },  
        {  
            "Arn": "arn:aws:iotwireless:us-  
east-1:123456789012:NetworkAnalyzerConfiguration/90123456-a1a2-9a87-65b4-c12bf3c2d09a",  
            "Name": "My_Network_Analyzer_Config2"  
        }  
    ]  
}
```

Supprimer la configuration de votre analyseur de réseau

Vous pouvez supprimer une configuration que vous n'utilisez plus à l'aide de l'opération [DeleteNetworkAnalyzerConfiguration](#)API ou de la commande [delete-network-analyzer-configuration](#)CLI.

```
aws iotwireless delete-network-analyzer-configuration \  
    --configuration-name My_NetworkAnalyzer_Config
```

L'exécution de cette commande ne produit aucune sortie. Pour afficher les configurations disponibles, vous pouvez utiliser l'opération [ListNetworkAnalyzerConfigurations](#) API.

Étapes suivantes

Maintenant que vous avez créé une configuration d'analyseur de réseau, vous pouvez ajouter des ressources à votre configuration ou mettre à jour vos paramètres de configuration. Pour plus d'informations, veuillez consulter [Ajouter des ressources et mettre à jour la configuration de l'analyseur de réseau \(p. 1381\)](#).

Ajouter des ressources et mettre à jour la configuration de l'analyseur de réseau

Avant de pouvoir activer la messagerie de suivi, ajoutez les ressources que vous souhaitez surveiller à la configuration de votre analyseur de réseau. Les ressources peuvent être des périphériques LoRa WAN, des passerelles LoRa WAN ou des groupes de multidiffusion.

Prérequis

Avant de pouvoir ajouter des ressources :

1. Vous devez avoir intégré les passerelles et les appareils et ajouté tous les groupes de multidiffusion que vous souhaitez surveiller pour le WAN. AWS IoT Core LoRa Pour plus d'informations, veuillez consulter [Connexion de passerelles et d'appareils à AWS IoT Core for LoRaWAN \(p. 1276\)](#).

2. Vous devez avoir créé une configuration d'analyseur de réseau pour laquelle vous allez ajouter des ressources. Pour plus d'informations, veuillez consulter [Créer une configuration d'analyseur de réseau \(p. 1378\)](#).

Ajouter des ressources et mettre à jour les paramètres de configuration à l'aide de la console

Vous pouvez ajouter des ressources et personnaliser les paramètres facultatifs à l'aide de la AWS IoT console ou de l'API AWS IoT sans fil. Outre les ressources, vous pouvez également modifier vos paramètres de configuration et enregistrer la configuration mise à jour.

Ajoutez des ressources à votre configuration

1. Ouvrez le [hub Network Analyzer de la AWS IoT console](#) et choisissez la configuration pour laquelle vous souhaitez ajouter des ressources.
2. Choisissez Actions, puis Ajouter des ressources.
3. Ajoutez les ressources que vous souhaitez surveiller à l'aide de la passerelle sans fil et des identificateurs de périphériques sans fil. Vous pouvez choisir plusieurs ressources et ajouter jusqu'à 250 passerelles ou appareils sans fil.
4. Après avoir ajouté toutes les ressources, choisissez Ajouter.

Vous verrez le nombre de passerelles et d'appareils que vous avez ajoutés sur la page du hub Network Analyzer. Vous pouvez continuer à ajouter et à supprimer des ressources jusqu'à ce que vous activez la session de messagerie de suivi. Une fois la session activée, pour ajouter des ressources, vous devez désactiver la session.

Mettre à jour vos paramètres de configuration

1. Ouvrez le [hub Network Analyzer de la AWS IoT console](#) et choisissez la configuration pour laquelle vous souhaitez mettre à jour les paramètres.
2. Choisissez Actions, puis Edit (Modifier).
3. Choisissez si vous souhaitez désactiver les informations sur les cadres et utilisez Sélectionner les niveaux de journal pour sélectionner les niveaux de journal que vous souhaitez utiliser pour vos journaux de messages de suivi. Choisissez Save (Enregistrer).

Vous verrez les paramètres de configuration que vous avez spécifiés sur la page de détails de la configuration de votre analyseur de réseau.

Ajoutez des ressources et mettez à jour les paramètres de configuration à l'aide de l'API

Pour ajouter des ressources ou mettre à jour vos paramètres de configuration, utilisez l'[UpdateNetworkAnalyzerConfiguration](#) API ou l'[update-network-analyzer-configuration](#) interface de ligne de commande.

Note

Lorsque vous utilisez l'opération d'UpdateNetworkAnalyzerConfiguration API pour ajouter ou mettre à jour des ressources, vous ne pouvez spécifier que 99 ressources sans fil au maximum pour chaque demande d'API. Une configuration d'analyseur de réseau unique peut comporter jusqu'à 250 appareils sans fil et 250 passerelles sans fil combinés. Pour ajouter des ressources supplémentaires, utilisez la AWS IoT console, comme décrit dans la section ci-dessus.

- Mise à jour des paramètres configuration

Pour mettre à jour vos paramètres de configuration, utilisez le `TraceContent` paramètre pour spécifier le niveau de journalisation et si vous souhaitez activer les informations sur les cadres. Par exemple, la commande suivante met à jour les paramètres de configuration en désactivant les informations du cadre et en définissant le niveau de journalisation sur. `ERROR`

```
aws iotwireless update-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default \
--trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR"
```

- Ajouter des ressources

Pour ajouter des ressources, utilisez les `MulticastGroupsToAdd` paramètres `WirelessDevicesToAdd`, `WirelessGatewaysToAdd`, et pour spécifier les passerelles, les appareils et les groupes de multidiffusion que vous souhaitez ajouter à votre configuration. Par exemple, la commande suivante met à jour les paramètres de configuration et ajoute à votre configuration les ressources sans fil, spécifiées par leur `WirelessGatewayId`, `WirelessDeviceId`, et `MulticastGroupId`.

```
aws iotwireless update-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default \
--trace-content WirelessDeviceFrameInfo=DISABLED,LogLevel="ERROR" \
--wireless-gateways-to-add "12345678-a1b2-3c45-67d8-e90fa1b2c34d" "90123456-
de1f-2b3b-4c5c-bb1112223cd1"
--wireless-devices-to-add "1ffd32c8-8130-4194-96df-622f072a315f" \
--multicast-groups-to-add "12abd34e-5f67-89c2-9293-593b1bd862e0"
```

Pour supprimer des appareils ou des passerelles, utilisez les `MulticastGroupsToRemove` paramètres `WirelessDevicesToRemove`, `WirelessGatewaysToRemove`, et de l'API.

Obtenir des informations sur la configuration

L'exécution de `UpdateNetworkAnalyzerConfiguration` l'API ne produit aucun résultat. Pour afficher vos paramètres de configuration et les passerelles ou appareils que vous avez ajoutés, utilisez l'opération [GetNetworkAnalyzerConfiguration](#) API ou la `get-network-analyzer-configuration` commande. Indiquez le nom de la configuration de l'analyseur de réseau en entrée.

```
aws iotwireless get-network-analyzer-configuration \
--configuration-name NetworkAnalyzerConfig_Default
```

L'exécution de cette commande produit le résultat suivant.

```
{
    "TraceContent": {
        "WirelessDeviceFrameInfo": "DISABLED",
        "LogLevel": "ERROR"
    },
    "WirelessDevices": [],
    "WirelessGateways": [
        "a0dd70e5-8f15-41a5-89cf-310284e691a1",
        "41682155-de4f-4f8f-84bf-bb5557221fc8"
    ]
}
```

Étapes suivantes

Maintenant que vous avez ajouté des ressources et spécifié tous les paramètres de configuration facultatifs pour votre configuration, vous pouvez utiliser le WebSocket protocole AWS IoT Core pour établir une connexion avec le LoRa WAN afin d'utiliser l'analyseur de réseau. Vous pouvez ensuite activer la messagerie de suivi et commencer à recevoir des messages de suivi pour vos ressources. Pour plus d'informations, veuillez consulter [Diffusez les messages de suivi de l'analyseur de réseau avec WebSockets \(p. 1384\)](#).

Diffusez les messages de suivi de l'analyseur de réseau avec WebSockets

Lorsque vous utilisez le WebSocket protocole, vous pouvez diffuser les messages de suivi de l'analyseur de réseau en temps réel. Lorsque vous envoyez une demande, le service répond avec une structure JSON. Après avoir activé la messagerie de suivi, vous pouvez utiliser les journaux des messages pour obtenir des informations sur vos ressources et résoudre les erreurs. Pour plus d'informations, consultez [WebSocketprotocole](#).

La section suivante montre comment diffuser des messages de trace de Network Analyzer avecWebSockets.

Rubriques

- [Générez une demande présignée avec la bibliothèque WebSocket \(p. 1384\)](#)
- [Exemple de code Python pour générer une URL présignée \(p. 1388\)](#)
- [WebSocketmessages et codes d'état \(p. 1392\)](#)

Générez une demande présignée avec la bibliothèque WebSocket

La section suivante montre comment générer une demande présignée afin de pouvoir utiliser la WebSocket bibliothèque pour envoyer des demandes au service.

Ajoutez une politique pour les WebSocket demandes à votre rôle IAM

Pour utiliser le WebSocket protocole afin d'appeler l'analyseur de réseau, attachez la politique suivante au rôle AWS Identity and Access Management (IAM) qui émet cette demande.

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

Créer une URL présignée

Créez une URL pour votre WebSocket demande qui contient les informations nécessaires pour établir la communication entre votre application et l'analyseur de réseau. Pour vérifier l'identité de la demande, le

WebSocket streaming utilise le processus Amazon Signature Version 4 pour la signature des demandes.
[Pour en savoir plus](#)

Pour appeler l'analyseur de réseau, utilisez l'URL de StartNetworkAnalyzerStream demande. La demande sera signée à l'aide des informations d'identification pour le rôle IAM mentionné précédemment. L'URL a le format suivant, avec des sauts de ligne ajoutés pour plus de lisibilité. Vous devez ajouter le nom de la configuration sous la ligne &X-Amz-SignedHeaders=host. Tous les paramètres supplémentaires doivent être ajoutés en dessous de cette ligne, triés par ordre alphabétique.

L'exemple suivant montre comment utiliser cette URL de demande avec le nom de la configuration ***NaConfig***:

```
wss://api.iotwireless.<region>.amazonaws.com/start-network-analyzer-stream?configuration-name=NaConfig  
    &X-Amz-Algorithm=AWS4-HMAC-SHA256  
    &X-Amz-Date=20220427T001057Z&X-Amz-SignedHeaders=host  
    &X-Amz-Expires=300  
    &X-Amz-Credential=<credential_number>/account/<region>/iotwireless/aws4_request  
    &X-Amz-Signature=c123456789098765a012c3a45d6789dd01234af5678bba9bbc0dbc112a3334d
```

Note

Si votre URL n'inclut pas le nom de la configuration, AWS IoT Core for LoRa WAN inclura le nom par défaut de la configuration de l'analyseur de réseau. NetworkAnalyzerConfig_Default

```
GET wss://api.iotwireless.<region>.amazonaws.com/start-network-analyzer-stream?X-Amz-Algorithm=AWS4-HMAC-SHA256  
    &X-Amz-Credential=Signature Version 4 credential scope  
    &X-Amz-Date=date  
    &X-Amz-Expires=time in seconds until expiration  
    &X-Amz-Security-Token=security-token  
    &X-Amz-Signature=Signature Version 4 signature  
    &X-Amz-SignedHeaders=host
```

Utilisez les valeurs suivantes pour les paramètres de la version 4 de Signature :

- Algorithme X-Amz : algorithme que vous utilisez dans le processus de signature. La seule valeur valide est AWS4-HMAC-SHA256.
 - X-Amz-Credential : chaîne séparée par des barres obliques («/») qui est formée en concaténant votre identifiant de clé d'accès et les composants de l'étendue de vos informations d'identification. L'étendue des informations d'identification inclut la date au format YYYYMMDD, la AWS région, le nom du service et une chaîne de terminaison (aws4_request).
 - X-Amz-Date — Date et heure de création de la signature. [Générer](#) [Version 1](#) [Version 2](#) [Version 3](#) [Version 4](#) [Version 5](#) [Version 6](#) [Version 7](#) [Version 8](#) [Version 9](#) [Version 10](#) [Version 11](#) [Version 12](#) [Version 13](#) [Version 14](#) [Version 15](#) [Version 16](#) [Version 17](#) [Version 18](#) [Version 19](#) [Version 20](#) [Version 21](#) [Version 22](#) [Version 23](#) [Version 24](#) [Version 25](#) [Version 26](#) [Version 27](#) [Version 28](#) [Version 29](#) [Version 30](#) [Version 31](#) [Version 32](#) [Version 33](#) [Version 34](#) [Version 35](#) [Version 36](#) [Version 37](#) [Version 38](#) [Version 39](#) [Version 40](#) [Version 41](#) [Version 42](#) [Version 43](#) [Version 44](#) [Version 45](#) [Version 46](#) [Version 47](#) [Version 48](#) [Version 49](#) [Version 50](#) [Version 51](#) [Version 52](#) [Version 53](#) [Version 54](#) [Version 55](#) [Version 56](#) [Version 57](#) [Version 58](#) [Version 59](#) [Version 60](#) [Version 61](#) [Version 62](#) [Version 63](#) [Version 64](#) [Version 65](#) [Version 66](#) [Version 67](#) [Version 68](#) [Version 69](#) [Version 70](#) [Version 71](#) [Version 72](#) [Version 73](#) [Version 74](#) [Version 75](#) [Version 76](#) [Version 77](#) [Version 78](#) [Version 79](#) [Version 80](#) [Version 81](#) [Version 82](#) [Version 83](#) [Version 84](#) [Version 85](#) [Version 86](#) [Version 87](#) [Version 88](#) [Version 89](#) [Version 90](#) [Version 91](#) [Version 92](#) [Version 93](#) [Version 94](#) [Version 95](#) [Version 96](#) [Version 97](#) [Version 98](#) [Version 99](#) [Version 100](#)
 - X-Amz-Expires : délai en secondes avant l'expiration des informations d'identification. La valeur maximale est de 300 secondes (5 minutes).
 - X-Amz-Security-Token : (facultatif) Un jeton de signature version 4 pour les informations d'identification temporaires. Si vous spécifiez ce paramètre, incluez-le dans la requête canonique. Pour plus d'informations, consultez la section [Demande d'informations d'identification de sécurité temporaires](#) dans le Guide de l'utilisateur de la AWS Identity and Access Management.

- X-Amz-Signature : signature de la version 4 que vous avez générée pour la demande.
 - X-Amz-SignedHeaders : en-têtes qui sont signés lors de la création de la signature pour la demande. La seule valeur valide est host.

Construisez l'URL de la demande et créez la signature Signature Version 4

Pour construire l'URL de la demande et créer la signature Signature Version 4, utilisez les étapes suivantes.

Note

Les exemples de cette section sont en pseudocode. Pour un exemple de code Python montrant comment créer la signature, reportez-vous à la section [Exemple de code Python pour générer une URL présignée](#) (p. 1388).

Tâche 1 : créer une demande canonique

Créez une chaîne qui inclut des informations de votre demande dans un format normalisé. Cela garantit qu'à la AWS réception de la demande, il peut calculer la même signature que celle dans laquelle vous effectuez le calcul [Tâche 3 : calculer la signature \(p. 1388\)](#). Pour en savoir plus informations, consultez [Créer une demande de signature](#).

1. Définissez des variables pour la demande dans votre application.

```
# HTTP verb
method = "GET"

# Service name
service = "iotwireless"

# Région AWS
region = "Région AWS"

# Service streaming endpoint
endpoint = "wss://api.iotwireless.{region}.amazonaws.com"

# Host
host = "api.iotwireless.<region>.amazonaws.com"

# Date and time of request
amz-date = 'YYYYMMDD'T'HHMMSS'Z'

# Date without time for credential scope
datestamp = 'YYYYMMDD'
```

- Créez un URI canonique (identifiant de ressource uniforme). L'URI canonique est la partie de l'URI entre le domaine et la chaîne de requête.

```
canonical_uri = "/start-network-analyzer-stream"
```

- Créez les en-têtes canoniques et les en-têtes signés. Notez la barre oblique \n dans les en-têtes canoniques.

- Ajoutez le nom d'en-tête en minuscules suivi de deux points.
 - Ajoutez une liste de valeurs séparées par des virgules pour cet en-tête. Ne triez pas les valeurs dans des en-têtes contenant plusieurs valeurs.
 - Ajoutez une nouvelle ligne (\n).

```
canonical_headers = "host:" + host + "\n"
signed_headers = "host"
```

4. Associez l'algorithme à l'algorithme de hachage. Vous devez utiliser SHA-256.

```
algorithm = "AWS4-HMAC-SHA256"
```

5. Créez la portée des informations d'identification qui déterminera la clé dérivée de la date, de la région et du service auquel la demande est adressée.

```
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"
```

- Créez la chaîne de requête canonique. Les valeurs des chaînes de requête doivent être codées en URI et triées par nom.

- Triez les noms de paramètre selon le point de code de caractère dans l'ordre croissant. Les paramètres avec des noms en double doivent être triés par valeur. Par exemple, un nom de paramètre qui commence par la lettre majuscule F précède un nom de paramètre qui commence par la lettre minuscule b.
 - N'encodez en mode URI aucun des caractères non réservés définis dans la [RFC 3986](#) : A à Z, a à z, 0 à 9, tiret (-), trait de soulignement (_), point (.) et tilde soulignement (~).
 - Encodez de pourcentage tous les autres caractères avec %XY, où X et Y représentent les caractères hexadécimaux (0 à 9 et les lettres majuscules A à F). Par exemple, le caractère d'espace doit être codé sous la forme %20 (sans utiliser « + », comme le font certains schémas de codage) et les caractères UTF-8 étendus doivent être au format %XY%ZA%BC.
 - Encodez deux fois tous les caractères égaux à (=) dans les valeurs de paramètre.

```
canonical_querystring = "X-Amz-Algorithm=" + algorithm
canonical_querystring += "&X-Amz-Credential=" + URI-encode(access_key + "/" +
    credential_scope)
canonical_querystring += "&X-Amz-Date=" + amz_date
canonical_querystring += "&X-Amz-Expires=300"
canonical_querystring += "&X-Amz-Security-Token=" + token
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers
```

7. Créez un hachage de la charge utile. Pour une demande GET, la charge utile est une chaîne vide.

```
payload_hash = HashSHA256("").Encode("utf-8")).HexDigest()
```

8. Combinez tous les éléments pour créer la demande canonique.

```
canonical_request = method + '\n' + canonical_uri + '\n' + canonical_querystring + '\n' + canonical_headers + '\n' + signed_headers + '\n' + payload_hash
```

Tâche 2 : Créer la chaîne à signer

```
string_to_sign=algorithm + "\n"
+ amz_date + "\n"
+ credential_scope + "\n"
+ HashSHA256(canonical_request.Encode("utf-8")).HexDigest()
```

Tâche 3 : calculer la signature

Le code suppose que vous avez mis en œuvre la fonction, `GetSignatureKey`, pour dériver une clé de signature. Pour plus d'informations et des exemples de fonctions, consultez la section [Exemples de dérivation d'une clé de signature pour la version 4](#) de Signature dans la référence générale d'Amazon Web Services.

La fonction `HMAC(key, data)` représente une fonction HMAC-SHA256 qui renvoie les résultats au format binaire.

```
#Create the signing key
signing_key = GetSignatureKey(secret_key, datestamp, region, service)

# Sign the string_to_sign using the signing key
signature = HMAC.new(signing_key, (string_to_sign).Encode("utf-8")), Sha256()).HexDigest
```

Tâche 4 : ajouter des informations de signature à la demande et créer l'URL de la demande

Vous pouvez ensuite utiliser une WebSocket bibliothèque pour demander l'URL présignée. Pour un exemple de WebSocket client à utiliser avec Python, voir [websocket-client](#) 1.4.1.

```
#Add the authentication information to the query string  
canonical_querystring += "&X-Amz-Signature=" + signature  
  
# Sign the string_to_sign using the signing key  
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

Étapes suivantes

Vous pouvez désormais utiliser l'URL de demande associée à votre WebSocket bibliothèque pour envoyer la demande au service et observer les messages. Pour un exemple de code Python montrant comment générer l'URL présignée, consultez [Exemple de code Python pour générer une URL présignée \(p. 1388\)](#).

Exemple de code Python pour générer une URL présignée

Le code suivant montre un exemple de génération de l'URL pré-signée à l'aide de Python comme langage de programmation.

Prérequis

Pour utiliser le langage de programmation Python afin de générer des requêtes, vous devez disposer des éléments suivants :

- Python installé sur votre ordinateur. Vous pouvez exécuter la commande suivante ou télécharger le [programme d'installation Python](#), puis l'exécuter.

```
sudo apt install python3
```

- La bibliothèque de requêtes Python. Vous pouvez exécuter la commande suivante ou télécharger la [bibliothèque Requests](#), qui est utilisée dans l'exemple de script pour effectuer des requêtes Web.

```
pip install requests
```

- Clé d'accès composée de clé d'accès et d'une clé d'accès secrète dans les variables d'environnement nommées AWS_ACCESS_KEY_ID et AWS_SECRET_ACCESS_KEY. Sinon, vous pouvez conserver ces valeurs dans un fichier d'informations d'identification et les lire à partir de ce fichier.

Note

Comme bonne pratique, nous vous recommandons de ne pas incorporer d'informations d'identification dans le code. Pour plus d'informations, veuillez consulter la rubrique [Bonnes pratiques pour les comptes AWS](#) dans le Guide de référence AWS Account Management.

```
$ export AWS_ACCESS_KEY_ID=My_Access_Key
$ export AWS_SECRET_ACCESS_KEY=My_Secret_Key

# Session token is required only if you use temporary access key starting with "ASIA"
$ export AWS_SESSION_TOKEN=My_Session_token
```

- Configuration d'analyseur de réseau créée dans votre compte. Pour exécuter le script, vous devez spécifier le nom de la configuration de votre analyseur de réseau sous forme de variable. Cet exemple fait référence à une configuration par son nom *My_Network_Analyzer_Config*.

Note

Si vous ne spécifiez pas de nom de configuration, la configuration par défaut sera utilisée *NetworkAnalyzerConfig_Default*.

Exemple de code Python

Le code Python génère l'URL pré-signée que la WebSocket bibliothèque peut utiliser pour envoyer des demandes au service. La fonction crée une requête canonique, puis crée la chaîne à signer qui est utilisée pour calculer la signature, puis ajoute la signature à la requête HTTP pour créer l'URL pré-signée. Vous pouvez ensuite utiliser la WebSocket bibliothèque pour demander l'URL pré-signée.

Pour exécuter le script `generate_presigned_url.py`, exécutez la commande suivante si vous l'exéutez depuis le même chemin que celui où se trouve le script.

```
python generate_presigned_url.py
```

L'exemple suivant affiche le contenu du `generate_presigned_url.py` script.

Contenu de `generate_presigned_url.py`

```
# Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
# SPDX-License-Identifier: Apache-2.0
```

```
# Version 4 signing example

"""
Sample Python code to generate the pre-signed URL. You can
change the parameters in this code to your own values, such
as the variables that are required for the request URL, the
network analyzer configuration name, and Region.
"""

# -----
# Step 1. Import the required libraries and define the functions
# sign and getSignatureKey that will be used to derive a signing key.
# -----
import sys, os, base64, datetime, hashlib, hmac, urllib.parse
import requests      # pip install requests

def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()

def getSignatureKey(key, dateStamp, regionName, serviceName):
    kDate = sign(("AWS4" + key).encode("utf-8"), dateStamp)
    kRegion = sign(kDate, regionName)
    kService = sign(kRegion, serviceName)
    kSigning = sign(kService, "aws4_request")
    return kSigning

# -----
# Step 2. Define the variables required for the request URL. Replace
# values for the variables, such as region, with your own values.
# -----
method = "GET"
service = "iotwireless"
region = "us-east-1"

# Host and endpoint information.
host = "api.iotwireless." + region + ".amazonaws.com"
endpoint = "wss://" + host

# Create a date for headers and the credential string.
t = datetime.datetime.utcnow()
amz_date = t.strftime("%Y%m%dT%H%M%SZ")

# For date stamp, the date without time is used in credential scope.
datestamp = t.strftime("%Y%m%d")

# -----
# Step 3. Create the canonical URI and canonical headers for the request.
# -----
canonical_uri = "/start-network-analyzer-stream"
configuration_name = "My_Network_Analyzer_Config"

canonical_headers = "host:" + host + "\n"
signed_headers = "host"
algorithm = "AWS4-HMAC-SHA256"
credential_scope = datestamp + "/" + region + "/" + service + "/" + "aws4_request"

# -----
# Step 4. Read the credentials that are required for the request
# from environment variables or configuration file.
# -----
# IMPORTANT: Best practice is NOT to embed credentials in code.

access_key = os.environ.get("AWS_ACCESS_KEY_ID")
secret_key = os.environ.get("AWS_SECRET_ACCESS_KEY")
token = os.environ.get("AWS_SESSION_TOKEN")
```

```
if access_key is None or secret_key is None:  
    print("No access key is available.")  
    sys.exit()  
  
if access_key.startswith("ASIA") and token is None:  
    print("Detected temporary credentials. You must specify a token.")  
    sys.exit()  
  
# -----  
# Step 5. Create the canonical query string. Query string values must be  
# URI-encoded and sorted by name. Query headers must in alphabetical order.  
# -----  
canonical_querystring = "X-Amz-Algorithm=" + algorithm  
  
canonical_querystring += "&X-Amz-Credential=" + \  
urllib.parse.quote(access_key + "/" + credential_scope, safe="-_.~")  
  
canonical_querystring += "&X-Amz-Date=" + amz_date  
canonical_querystring += "&X-Amz-Expires=300"  
  
if access_key.startswith("ASIA"):  
    # percent encode the token and double encode "="  
    canonical_querystring += "&X-Amz-Security-Token=" + \  
urllib.parse.quote(token, safe="-_.~").replace("=", "%253D")  
  
canonical_querystring += "&X-Amz-SignedHeaders=" + signed_headers  
canonical_querystring += "&configuration-name=" + configuration_name  
  
# -----  
# Step 6. Create a hash of the payload.  
# -----  
payload_hash = hashlib.sha256("").encode("utf-8")).hexdigest()  
  
# -----  
# Step 7. Combine the elements, which includes the query string, the  
# headers, and the payload hash, to form the canonical request.  
# -----  
canonical_request = method + "\n" + canonical_uri + "\n" + canonical_querystring \  
+ "\n" + canonical_headers + "\n" + signed_headers + "\n" + payload_hash  
  
# -----  
# Step 8. Create the metadata string to store the information required to  
# calculate the signature in the following step.  
# -----  
string_to_sign = algorithm + "\n" + amz_date + "\n" + \  
credential_scope + "\n" + hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()  
  
# -----  
# Step 9. Calculate the signature by using a signing key that's obtained  
# from your secret key.  
# -----  
# Create the signing key from your secret key.  
signing_key = getSignatureKey(secret_key, datestamp, region, service)  
  
# Sign the string_to_sign using the signing key.  
signature = hmac.new(signing_key, (string_to_sign).encode("utf-8"),  
hashlib.sha256).hexdigest()  
  
# -----  
# Step 10. Create the request URL using the calculated signature and by  
# combining it with the canonical URI and the query string.  
# -----  
canonical_querystring += "&X-Amz-Signature=" + signature  
  
request_url = endpoint + canonical_uri + "?" + canonical_querystring
```

```
print("\n-----PRESIGNED URL-----")
print(request_url)
```

Étapes suivantes

Vous pouvez désormais utiliser l'URL de demande associée à votre WebSocket bibliothèque pour envoyer la demande au service et observer les messages.

Pour installer une WebSocket bibliothèque à utiliser avec Python, exécutez la commande suivante.

Pour plus d'informations sur l'utilisation d'un WebSocket client avec Python, consultez la section

[WebSocketClient pour Python avec des options d'API de bas niveau](#).

```
pip install websocket-client
```

Une fois que vous avez installé le client et effectué la demande, vous verrez des messages et des codes d'état indiquant l'état de votre demande. Pour plus d'informations, veuillez consulter [WebSocketmessages et codes d'état \(p. 1392\)](#).

WebSocketmessages et codes d'état

Après avoir créé une demande présignée, vous pouvez utiliser l'URL de la demande avec votre WebSocket bibliothèque, ou une bibliothèque adaptée à votre langage de programmation, pour envoyer des demandes au service. Pour en savoir plus sur la génération de cette demande présignée, consultez [Générez une demande présignée avec la bibliothèque WebSocket \(p. 1384\)](#).

WebSocketmessages

Le WebSocket protocole peut être utilisé pour établir une connexion bidirectionnelle. Les messages peuvent être transmis du client au serveur et du serveur au client. Toutefois, l'analyseur de réseau prend uniquement en charge les messages envoyés du serveur au client. Tout message reçu du client est inattendu et le serveur ferme automatiquement la WebSocket connexion si un message est reçu du client.

Lorsque la demande est reçue et qu'une session de messagerie de suivi est lancée, le serveur répond avec une structure JSON, qui constitue la charge utile. Pour plus d'informations sur la charge utile et sur la manière dont vous pouvez activer la messagerie de suivi à partir du AWS Management Console, voir [Afficher et surveiller les journaux des messages de suivi de l'analyseur de réseau en temps réel \(p. 1393\)](#).

WebSocketcodes d'état

Ce qui suit montre les codes d'WebSocketétat de la communication entre le serveur et le client. Les codes WebSocket d'état sont conformes à la [norme RFC relative à la fermeture normale des connexions](#).

L'exemple suivant affiche les codes d'état pris en charge :

- 1 000

Ce code d'état indique une fermeture normale, ce qui signifie que la WebSocket connexion a été établie et que la demande a été satisfaite. Cet état peut être observé lorsqu'une session est inactive, ce qui entraîne l'expiration du délai de connexion.

- 1 002

Ce code d'état indique que le point de terminaison met fin à la connexion en raison d'une erreur de protocole.

- 1003

Ce code d'état indique un état d'erreur selon lequel le point de terminaison a mis fin à la connexion parce qu'il a reçu des données dans un format qu'il ne peut pas accepter. Le point de terminaison ne prend

en charge que les données texte et peut afficher ce code d'état s'il reçoit un message binaire ou un message du client utilisant un format non pris en charge.

- 1008

Ce code d'état indique un état d'erreur selon lequel le point de terminaison a mis fin à la connexion parce qu'il a reçu un message violant sa politique. Ce statut est générique et s'affiche lorsque les autres codes d'état, tels que 1003 ou 1009, ne sont pas applicables. Ce statut s'affiche également s'il est nécessaire de masquer la politique ou en cas d'échec d'autorisation, tel qu'une signature expirée.

- 1011

Ce code d'état indique un état d'erreur dans lequel le serveur met fin à la connexion parce qu'il a rencontré une condition inattendue ou une erreur interne qui l'a empêché de répondre à la demande.

Étapes suivantes

Maintenant que vous savez comment générer une demande présignée et comment observer les messages du serveur à l'aide de la WebSocket connexion, vous pouvez activer la messagerie de suivi et commencer à recevoir des journaux de messages pour votre passerelle sans fil et les ressources de votre appareil sans fil. Pour plus d'informations, veuillez consulter [Afficher et surveiller les journaux des messages de suivi de l'analyseur de réseau en temps réel \(p. 1393\)](#).

Afficher et surveiller les journaux des messages de suivi de l'analyseur de réseau en temps réel

Si vous avez ajouté des ressources à la configuration de votre analyseur de réseau, vous pouvez activer la messagerie de suivi pour commencer à recevoir des messages de suivi pour vos ressources. Vous pouvez utiliser AWS Management Console l'API AWS IoT Wireless ou AWS CLI.

Prérequis

Avant de pouvoir activer la messagerie de suivi à l'aide de l'analyseur de réseau, vous devez disposer des éléments suivants :

- Les ressources que vous souhaitez surveiller ont été ajoutées à la configuration par défaut de votre analyseur de réseau. Pour plus d'informations, veuillez consulter [Ajouter des ressources et mettre à jour la configuration de l'analyseur de réseau \(p. 1381\)](#).
- Génération d'une demande présignée à l'aide de l'URL de la StartNetworkAnalyzerStream demande. La demande sera signée à l'aide des informations d'identification du AWS Identity and Access Management rôle à l'origine de cette demande. Pour de plus amples informations, consultez [Créer une URL présignée \(p. 1384\)](#).

Activer la messagerie de suivi à l'aide de la console

Pour activer la messagerie de suivi

1. Ouvrez le [hub Network Analyzer de la AWS IoT console](#) et choisissez la configuration de votre analyseur de réseau, NetworkAnalyzerConfig_Default.
2. Sur la page de détails de la configuration de votre analyseur de réseau, choisissez Activer la messagerie de suivi, puis sélectionnez Activer.

Vous allez commencer à recevoir des messages de suivi lorsque le message de suivi le plus récent apparaît en premier dans la console.

Note

Une fois la session de messagerie lancée, la réception de messages de suivi peut entraîner des coûts supplémentaires jusqu'à ce que vous désactiviez la session ou que vous la quittiez. Pour de plus amples informations sur la tarification, veuillez consulter [Tarification AWS IoT Core](#).

Afficher et surveiller les messages de suivi

Une fois que vous avez activé la messagerie de suivi, la WebSocket connexion est établie et les messages de suivi commencent à apparaître en temps réel, les plus récents en premier. Vous pouvez personnaliser les préférences pour spécifier le nombre de messages de suivi à afficher sur chaque page et pour afficher uniquement les champs pertinents pour chaque message. Par exemple, vous pouvez personnaliser le journal des messages de suivi pour n'afficher que les journaux des ressources de passerelle sans fil dont le niveau de journalisation est défini surERROR, afin de pouvoir rapidement identifier et corriger les erreurs liées à vos passerelles. Les messages de suivi contiennent les types d'informations suivants.

- Numéro du message : numéro unique indiquant le dernier message reçu en premier.
- ID de ressource : ID de passerelle ou de périphérique sans fil de la ressource.
- Horodatage : L'heure à laquelle le message a été reçu.
- ID du message : identifiant que AWS IoT Core le LoRa WAN attribue à chaque message reçu.
- fPort : port de fréquence permettant de communiquer avec l'appareil à l'aide de la WebSocket connexion.
- DevEui: L'identifiant unique étendu (EUI) de votre appareil sans fil.
- Ressource : indique si la ressource surveillée est un périphérique sans fil ou une passerelle sans fil.
- Événement : événement correspondant à un message de journal pour un appareil sans fil, qui peut être Join, Rejoin, Uplink_Data, Downlink_Data ou Registration.
- Niveau de journalisation : informations concernant ou INFO enregistrant les ERROR flux de votre appareil.

Message du journal JSON de l'analyseur de réseau

Vous pouvez également choisir un message de suivi à la fois pour afficher la charge utile JSON de ce message. Selon le message que vous sélectionnez dans les journaux des messages de suivi, vous verrez apparaître des informations dans la charge utile JSON qui indiquent qu'il contient deux parties : CustomerLog et LoRaFrame.

CustomerLog

La CustomerLogpartie du JSON affiche le type et l'identifiant de la ressource qui a reçu le message, le niveau du journal et le contenu du message. L'exemple suivant montre un message de journal CustomerLog. Vous pouvez utiliser le message champ du fichier JSON pour obtenir plus d'informations sur l'erreur et sur la manière de la résoudre.

LoRaFrame

La LoRaFramepartie du JSON possède un ID de message et contient des informations sur la charge utile physique du périphérique et les métadonnées sans fil.

L'exemple suivant affiche la structure du message de suivi.

Note

Si vos appareils envoient un message de liaison montante sans valeur AWS IoT Core pourFport, car l'analyseur de réseau LoRa WAN affichera la fPort valeur 225 dans le message de suivi reçu.

```
export type TraceMessage = {
  ResourceId: string;
  Timestamp: string;
  LoRaFrame:
  {
    MessageId: string;
    PhysicalPayload: any;
    WirelessMetadata:
    {
      fPort: number;
      dataRate: number;
      devEui: string;
      frequency: number,
      timestamp: string;
    },
  }
  CustomerLog:
  {
    resource: string;
    wirelessDeviceId: string;
    wirelessDeviceType: string;
    event: string;
    logLevel: string;
    messageId: string;
    message: string;
  },
};
```

Révision et Étapes suivantes

Dans cette section, vous avez consulté les messages de suivi et appris comment utiliser ces informations pour corriger les erreurs. Après avoir consulté tous les messages, vous pouvez :

- Désactiver la messagerie de suivi

Pour éviter des coûts supplémentaires, vous pouvez désactiver la session de messagerie de suivi. La désactivation de la session déconnecte votre WebSocket connexion et vous ne recevrez aucun message de suivi supplémentaire. Vous pouvez toujours continuer à consulter les messages existants dans la console.

- Modifier les informations du cadre pour votre configuration

Vous pouvez modifier la configuration de l'analyseur de réseau et choisir de désactiver les informations de cadre et de choisir les niveaux de journalisation de vos messages. Avant de mettre à jour votre configuration, pensez à désactiver votre session de messagerie de suivi. Pour effectuer ces modifications, ouvrez la [page de détails de Network Analyzer dans la AWS IoT console](#) et choisissez Modifier. Vous pouvez ensuite mettre à jour votre configuration avec les nouveaux paramètres de configuration et activer la messagerie de suivi pour voir les messages mis à jour.

- Ajoutez des ressources à votre configuration

Vous pouvez également ajouter des ressources supplémentaires à la configuration de votre analyseur de réseau et les surveiller en temps réel. Vous pouvez ajouter jusqu'à 250 ressources de passerelle sans fil et d'appareil sans fil. Pour ajouter des ressources, sur la [page de détails de Network Analyzer de la AWS IoT console](#), choisissez l'onglet Ressources, puis Ajouter des ressources. Vous pouvez ensuite mettre à jour votre configuration avec les nouvelles ressources et activer la messagerie de suivi pour voir les messages mis à jour pour les ressources supplémentaires.

Pour plus d'informations sur la mise à jour de la configuration de votre analyseur de réseau en modifiant les paramètres de configuration et en ajoutant des ressources, consultez [Ajouter des ressources et mettre à jour la configuration de l'analyseur de réseau \(p. 1381\)](#).

Déboguez et dépannez vos groupes de multidiffusion et vos tâches FUOTA à l'aide d'un analyseur de réseau

Les ressources sans fil que vous pouvez surveiller incluent les périphériques LoRa WAN, les passerelles LoRa WAN et les groupes de multidiffusion. Vous pouvez également utiliser l'analyseur de réseau pour déboguer et résoudre tout problème lié à votre tâche FUOTA. Vous pouvez également surveiller et suivre les messages liés à la configuration, à la transmission de données et à la requête d'état lorsque la tâche FUOTA est en cours.

Pour surveiller votre tâche FUOTA, si la tâche contient des groupes de multidiffusion, vous devez ajouter à la fois le groupe de multidiffusion et les appareils du groupe à la configuration de votre analyseur de réseau. Vous devez également activer les informations sur les trames et les informations sur les trames de multidiffusion pour suivre les messages de liaison montante et descendante de monodiffusion et de multidiffusion qui sont échangés avec le groupe de multidiffusion et les appareils pendant que la tâche FUOTA est en cours.

Pour surveiller les groupes de multidiffusion, vous pouvez les ajouter à la configuration de votre analyseur de réseau et utiliser les informations de trame de multidiffusion pour résoudre les messages de multidiffusion en liaison descendante envoyés à ces groupes. Pour dépanner les appareils qui tentent de rejoindre un groupe où la communication monodiffusion est utilisée, vous devez également inclure ces appareils dans la configuration de l'analyseur de réseau. Pour surveiller uniquement la communication monodiffusion avec les appareils du groupe, activez les informations de cadre pour vos appareils sans fil. Cette approche garantit une surveillance et des diagnostics complets à la fois pour les groupes de multidiffusion et pour les appareils qui rejoignent le groupe.

Les sections suivantes décrivent comment déboguer et dépanner vos groupes de multidiffusion et vos tâches FUOTA à l'aide de l'analyseur de réseau.

Rubriques

- [Déboguer les tâches FUOTA qui ne contiennent que des appareils \(p. 1396\)](#)
- [Déboguer les tâches FUOTA avec des groupes de multidiffusion \(p. 1397\)](#)
- [Déboguer les appareils qui tentent de rejoindre un groupe de multidiffusion \(p. 1397\)](#)
- [Démarrage d'une session de groupe multicast \(p. 1398\)](#)

Déboguer les tâches FUOTA qui ne contiennent que des appareils

Vous pouvez utiliser l'analyseur de réseau pour déboguer une tâche FUOTA à laquelle seuls des périphériques LoRa WAN sont ajoutés. Pour plus d'informations sur l'ajout de périphériques à une tâche FUOTA, consultez [Ajouter des appareils et des groupes de multidiffusion à une tâche FUOTA et planifier une session FUOTA \(p. 1369\)](#). Pour débugger la tâche FUOTA, effectuez les opérations suivantes :

1. Créez une configuration d'analyseur de réseau en activant les informations de trame pour les appareils sans fil afin de pouvoir surveiller les messages de liaison montante et descendante FUOTA qui sont échangés avec les appareils pendant que la tâche est en cours.
2. Ajoutez les appareils de votre tâche FUOTA à la configuration de l'analyseur de réseau en utilisant leurs identifiants d'appareils sans fil.
3. Activez la messagerie de suivi pour commencer à recevoir des messages de suivi pour les appareils de la configuration de votre analyseur de réseau.

Dans la PkgCmdType colonne des informations du message de suivi, vous allez commencer à recevoir des messages de liaison descendante monodiffusion relatifs à la configuration de la transmission et de la fragmentation des données.

Note

Si vous ne voyez pas la PkgCmdType colonne dans le tableau des messages de suivi, vous pouvez ajuster les paramètres pour qu'elle apparaisse dans le tableau.

Vous pouvez également consulter les messages PkgCmdType et d'autres messages détaillés dans le message du journal JSON sous WirelessMetadata> ApplicationInfo.

Déboguer les tâches FUOTA avec des groupes de multidiffusion

Vous pouvez utiliser l'analyseur de réseau pour déboguer une tâche FUOTA à laquelle des groupes de multidiffusion et des périphériques LoRa WAN ont été ajoutés au groupe. Pour plus d'informations sur l'ajout de périphériques à une tâche FUOTA, consultez[Ajouter des appareils et des groupes de multidiffusion à une tâche FUOTA et planifier une session FUOTA \(p. 1369\)](#). Pour débugger la tâche FUOTA, effectuez les opérations suivantes :

1. Créez une configuration d'analyseur de réseau en activant les paramètres d'informations sur les trames et d'informations sur les trames de multidiffusion pour les appareils sans fil et les groupes de multidiffusion.
2. Ajoutez le groupe de multidiffusion de votre tâche FUOTA à la configuration de l'analyseur de réseau en utilisant son identifiant de groupe de multidiffusion. En activant les informations sur les trames de multidiffusion, vous pouvez déboguer les messages de données du microprogramme et les messages de requête d'état FUOTA qui sont envoyés au groupe pendant que la tâche FUOTA est en cours.
3. Ajoutez les appareils de votre groupe de multidiffusion à la configuration de l'analyseur de réseau à l'aide de leurs identifiants d'appareils sans fil. En activant les informations du cadre, vous pouvez surveiller les messages en liaison montante et descendante qui sont échangés directement avec les appareils pendant que la tâche FUOTA est en cours.
4. Activez la messagerie de suivi pour commencer à recevoir des messages de suivi pour les appareils et les groupes de multidiffusion dans la configuration de votre analyseur de réseau.

Vous pouvez ensuite consulter les messages de suivi et les déboguer en utilisant la PkgCmdTpye colonne du tableau des messages de suivi et en utilisant les détails du message du journal JSON, comme décrit dans[Déboguer les tâches FUOTA qui ne contiennent que des appareils \(p. 1396\)](#).

Déboguer les appareils qui tentent de rejoindre un groupe de multidiffusion

Vous pouvez utiliser l'analyseur de réseau pour déboguer les appareils qui tentent de rejoindre un groupe de multidiffusion. Pour plus d'informations sur l'ajout d'appareils à un groupe de multidiffusion, consultez[Créez des groupes de multidiffusion et ajoutez des appareils au groupe \(p. 1356\)](#). Pour débugger le groupe multicast, effectuez les opérations suivantes :

1. Créez une configuration d'analyseur de réseau en activant les informations de trame pour les appareils sans fil.
2. Ajoutez les appareils que vous souhaitez surveiller à la configuration de l'analyseur de réseau à l'aide de leurs identifiants d'appareils sans fil.
3. Activez la messagerie de suivi pour commencer à recevoir des messages de suivi pour les appareils de la configuration de votre analyseur de réseau.
4. Commencez à associer les appareils au groupe de multidiffusion une fois que la messagerie de suivi a été activée pour les appareils du groupe.

Démarrage d'une session de groupe multicast

Vous pouvez utiliser l'analyseur de réseau pour déboguer une session de groupe de multidiffusion. Pour plus d'informations, veuillez consulter [Programmez un message de liaison descendante à envoyer aux appareils de votre groupe de multidiffusion \(p. 1361\)](#). Pour débugger une session de groupe multicast, effectuez les opérations suivantes :

1. Créez une configuration d'analyseur de réseau en activant les informations de trame de multidiffusion pour le groupe de multidiffusion.
2. Ajoutez le groupe de multidiffusion que vous souhaitez surveiller à la configuration de l'analyseur de réseau à l'aide de son identifiant de groupe de multidiffusion.
3. Avant le début de la session de multidiffusion, activez la messagerie de suivi pour commencer à recevoir des messages de suivi pour la session de groupe de multidiffusion.
4. Démarrez la session de groupe de multidiffusion et surveillez son état en consultant les messages affichés dans le tableau des messages de suivi et dans le message du journal JSON.

Dans le tableau des messages de suivi, le `MulticastAddr` sera affiché dans la `DevAddr` colonne. Dans le message du journal JSON, vous pouvez consulter des informations détaillées, telles que le `MulticastGroupId` lien ci-dessous `WirelessMetadata`> `ApplicationInfo`.

Sécurité des données avec AWS IoT Core for LoRa WAN

Deux méthodes sécurisent les données AWS IoT Core de vos quatre appareils LoRa WAN :

- La sécurité utilisée par les appareils sans fil pour communiquer avec les passerelles.

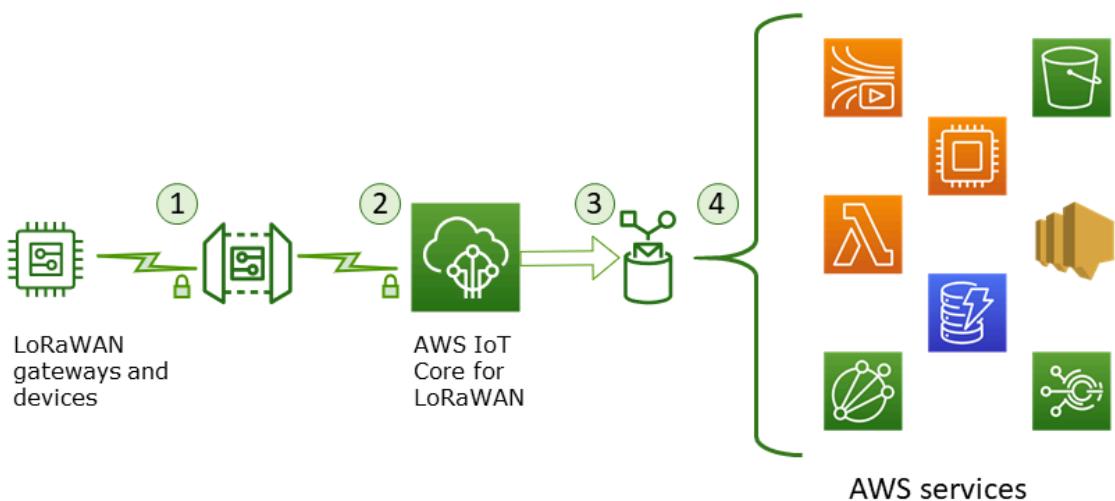
Les dispositifs LoRa WAN suivent les pratiques de sécurité décrites dans [LoRaWAN™ SECURITY : un livre blanc préparé pour l' LoRa Alliance™ par Gemalto, Actility et Semtech](#) pour communiquer avec les passerelles.

- La sécurité AWS IoT Core utilisée pour connecter les passerelles à AWS IoT Core un LoRa réseau WAN et envoyer les données à d'autres AWS services.

AWS IoT Core la sécurité est décrite dans [Protection des données dans AWS IoT Core \(p. 408\)](#).

Comment les données sont sécurisées dans l'ensemble du système

Ce diagramme identifie les éléments clés d'un système LoRa WAN connecté à un réseau LoRa WAN afin AWS IoT Core d'identifier la manière dont les données sont sécurisées dans l'ensemble.



1. Le périphérique sans fil LoRa WAN chiffre ses messages binaires à l'aide du mode AES128 CTR avant de les transmettre.
2. Les connexions AWS IoT Core de passerelle vers le LoRa WAN sont sécurisées par TLS, comme décrit dans [Sécurité du transport dans AWS IoT Core \(p. 409\)](#). AWS IoT Core for LoRa WAN déchiffre le message binaire et code la charge utile du message binaire déchiffré sous la forme d'une chaîne base64.
3. Le message codé en base64 qui en résulte est envoyé en tant que charge utile du message à la AWS IoT règle décrite dans la destination attribuée au périphérique. Les données qu'AWS il contient sont AWS cryptées à l'aide de clés détenues.
4. La AWS IoT règle dirige les données du message vers les services décrits dans la configuration de la règle. Les données qu'AWS il contient sont AWS cryptées à l'aide de clés détenues.

LoRaSécurité du transport des périphériques et des passerelles WAN

Les périphériques WAN et AWS IoT Core pour le LoRa WAN stockent des clés racines pré-partagées. Les clés de session sont dérivées à la fois par les périphériques LoRa WAN et AWS IoT Core pour le LoRa WAN selon 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 des messages (MIC) à 4 octets est également utilisé pour vérifier l'intégrité des données selon un algorithme CMAC AES-128 standard. Les clés de session peuvent être mises à jour à l'aide du processus Join/Rejoin.

Les pratiques de sécurité pour les LoRa passerelles sont décrites dans les spécifications du LoRa WAN. Les passerelles se connectent au LoRa WAN via un socket Web à l'aide d'un [AWS IoT Core Basics Station](#). AWS IoT Core pour le LoRa WAN ne prend en charge que la version 2.0.4 et les versions ultérieures.

Avant que la connexion au socket Web ne soit établie, AWS IoT Core for LoRa WAN utilise le [mode d'authentification du serveur TLS et du client \(p. 409\)](#) pour authentifier la passerelle. AWS IoT Core for LoRa WAN 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.

AWS IoT Core pour Amazon Sidewalk

AWS IoT Core pour Amazon Sidewalk fournit les services cloud que vous pouvez utiliser pour connecter vos terminaux Sidewalk à AWS Cloud et en utiliser d'autres services AWS.

Amazon Sidewalk est un réseau partagé sécurisé qui permet aux appareils de votre communauté de se connecter et de rester connectés. Amazon Sidewalk transfère des données entre les terminaux Sidewalk et les passerelles Sidewalk, ainsi qu'entre les passerelles Sidewalk et le cloud Sidewalk.

Comment l'utiliserAWS IoT Core pour Amazon Sidewalk ?

Vous pouvez y intégrer vos terminaux Sidewalk à l'aide AWS IoT de la console ou des opérations de l'API AWS IoT Wireless. Une fois vos appareils intégrés, leurs messages sont envoyés à AWS IoT Core. Vous pouvez ensuite commencer à développer vos applications professionnelles sur le AWS cloud, qui utilise les données de vos terminaux Amazon Sidewalk.

Utilisation de la console

Pour intégrer vos terminaux Sidewalk, connectez-vous à la page [Appareils](#) de la AWS IoT console AWS Management Console et accédez à celle-ci. Une fois vos appareils intégrés, vous pouvez les consulter et les gérer sur cette page de la console IoT.

Utilisation de l'API ou de la CLI

Vous pouvez intégrer à la fois des appareils Sidewalk et LoRa WAN à l'aide des [opérations de l'API AWS IoT sans fil](#). L'API AWS IoT Wireless qui AWS IoT Core est intégrée est prise en charge par le AWS SDK. Pour plus d'informations, consultez Kits et boîtes à boîtes [AWS à outils et boîtes à outils Kits](#) et boîtes à

Vous pouvez utiliser le AWS CLI pour exécuter des commandes d'intégration et de gestion de vos terminaux Sidewalk. Pour plus d'informations, consultez [Référence de l'interface ligne ligne AWS IoT ligne ligne ligne ligne ligne ligne commande](#)

AWS IoT Core pour Amazon Sidewalk Régions et points de terminaison

Amazon Sidewalk est uniquement disponible dans leus-east-1 Région AWS. AWS IoT Core pour Amazon Sidewalk prend en charge les points de terminaison des API du plan de contrôle et du plan de données dans cette région. Les points de terminaison de l'API Data Plane sont spécifiques à votre Compte AWS. Pour plus d'informations, consultez la section [Points de terminaison des services AWS IoT sans fil](#) dans la Référence AWS générale.

AWS IoT Core pour Amazon Sidewalk possède des quotas qui s'appliquent aux données de l'appareil transmises entre l'appareil et le AWS Cloud, ainsi que le TPS maximal pour les opérations de l'API AWS IoT sans fil. Pour plus d'informations, consultez la section [Quotas AWS IoT sans fil](#) dans la Référence AWS générale.

Tarification d'AWS IoT Core pour Amazon Sidewalk

Lorsque vous vous inscrivez à AWS, vous pouvez démarrer AWS IoT Core pour Amazon Sidewalk gratuitement avec l'offre [AWSgratuite](#).

Pour plus d'informations sur l'offre générale et la tarification des produits et tarifs de, consultez Tarification et [AWS IoT Coretarification des](#) produits et tarifs de

Qu'est-ce que AWS IoT Core pour Amazon Sidewalk ?

Avec AWS IoT Core pour Amazon Sidewalk, vous pouvez intégrer vos terminaux Amazon Sidewalk, les gérer AWS IoT et les surveiller. Il gère également les destinations qui envoient les données de l'appareil à d'autres services AWS.

Fonctionnalités d'AWS IoT Core pour Amazon Sidewalk

AWS IoT Core pour Amazon Sidewalk vous permet d'effectuer les opérations suivantes :

- Intégrez vos terminaux Sidewalk à AWS IoT l'utilisation de la AWS IoT console, des opérations de AWS IoT Core pour Amazon Sidewalk l'API ou des AWS CLI commandes.
- Tirez parti des fonctionnalités offertes par le AWS Cloud.
- Créez une destination qui utilise des AWS IoT règles pour traiter les messages de charge utile entrants et pour interagir avec d'autres services AWS.
- Activez les notifications d'événements pour recevoir des messages concernant des événements tels que la mise en service ou l'enregistrement de votre appareil Sidewalk, ou si un message de liaison descendante a été correctement transmis à votre appareil.
- Enregistrez et surveillez vos terminaux Sidewalk en temps réel, obtenez des informations utiles, identifiez et résolvez les erreurs.
- Associez vos terminaux Sidewalk à n'importe quel AWS IoT objet, ce qui vous permet de stocker une représentation de votre appareil sur le cloud. Les éléments qui s'y trouvent AWS IoT facilitent la recherche et la gestion de vos fonctionnalités, ainsi que l'accès à d'autres AWS IoT Core fonctionnalités.

Rubriques

- [Qu'est-ce qu'Amazon Sidewalk ? \(p. 1401\)](#)
- [Fonctionnement d'AWS IoT Core pour Amazon Sidewalk \(p. 1402\)](#)

Qu'est-ce qu'Amazon Sidewalk ?

Amazon Sidewalk est un réseau communautaire sécurisé qui utilise Amazon Sidewalk Bridges, tels que les appareils Amazon Echo et Ring compatibles, pour fournir une connectivité cloud aux appareils IoT. Amazon Sidewalk permet une connectivité à faible bande passante et à longue portée à la maison et au-delà grâce au Bluetooth LE pour les communications à courte distance LoRa et aux protocoles radio FSK à des fréquences de 900 MHz pour couvrir de plus longues distances.

Lorsque Amazon Sidewalk est activé, ce réseau peut prendre en charge d'autres terminaux Sidewalk de votre communauté et peut être utilisé pour des applications telles que la détection de votre environnement. Amazon Sidewalk permet à vos appareils d'être connectés et de rester connectés.

Fonctions d'Amazon Sidewalk

Voici les principales caractéristiques d'Amazon Sidewalk.

- Amazon Sidewalk crée un réseau à faible bande passante à l'aide de passerelles Sidewalk qui incluent Ring et certains appareils Echo. À l'aide de passerelles, vous pouvez partager une partie de votre bande passante Internet, qui est ensuite utilisée pour connecter vos appareils finaux au réseau.
- Amazon Sidewalk propose un mécanisme de mise en réseau sécurisé doté de plusieurs niveaux de cryptage et de sécurité.
- Amazon Sidewalk propose un mécanisme simple pour activer ou désactiver la participation à Sidewalk.

Concepts Amazon Sidewalk

Voici quelques concepts clés d'Amazon Sidewalk.

Passerelles de trottoir

Les passerelles Sidewalk, ou ponts Amazon Sidewalk, acheminent les données entre vos terminaux Sidewalk et le cloud. Les passerelles sont des appareils Amazon, tels que l'appareil Echo ou la Ring Floodlight Cam, qui prennent en charge les protocoles SubG-CSS (asynchrone, LDR), SubG-FSK (synchrone, HDR) ou Bluetooth LE pour les communications sur le trottoir. Les passerelles Sidewalk partagent une partie de votre bande passante Internet avec la communauté Sidewalk afin de fournir une connectivité à un groupe d'appareils compatibles Sidewalk.

Terminaux de trottoir

Les terminaux Sidewalk se déplacent sur Amazon Sidewalk en se connectant aux passerelles Sidewalk. Les appareils finaux sont des produits intelligents à faible bande passante et à faible consommation d'énergie, tels que des éclairages ou des serrures de porte compatibles Sidewalk.

Note

Certaines passerelles Sidewalk peuvent également servir de terminaux.

Serveur réseau Sidewalk

Le serveur réseau Sidewalk, géré par Amazon, vérifie les paquets entrants et achemine les messages en liaison montante et descendante vers la destination souhaitée, tout en maintenant la synchronisation temporelle du réseau Sidewalk.

En savoir plus sur Amazon Sidewalk

Pour plus d'informations sur Amazon Sidewalk, veuillez consulter les pages web suivantes :

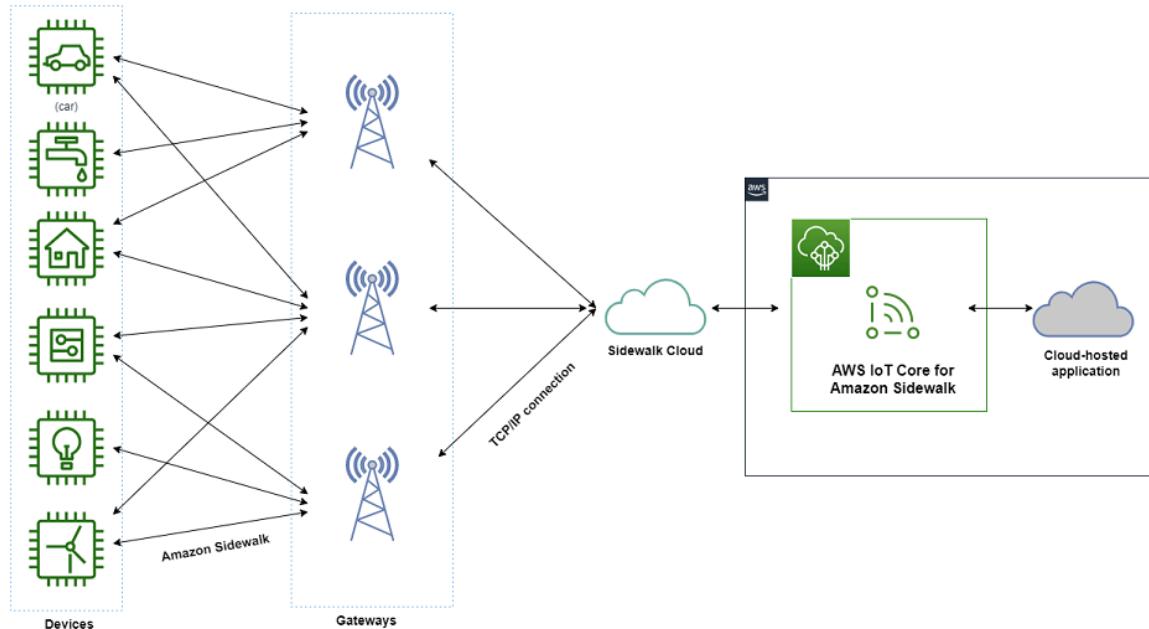
- [Trottoir Amazon](#)
- [Documentation sur Amazon Sidewalk](#)
- [AWS IoT Core pour Amazon Sidewalk](#)

Fonctionnement d'AWS IoT Core pour Amazon Sidewalk

Avec AWS IoT Core pour Amazon Sidewalk, vous pouvez intégrer vos terminaux Amazon Sidewalk, les gérer AWS IoT et les surveiller. Il gère également les destinations qui envoient les données de l'appareil à d'autres services AWS.

AWS IoT Core pour Amazon Sidewalk fournit les services cloud que vous pouvez utiliser pour connecter vos terminaux Sidewalk à AWS Cloud et en utiliser d'autres services AWS. Vous pouvez également l'utiliser avec AWS IoT Core pour Amazon Sidewalk pour gérer vos appareils Sidewalk, les surveiller et créer des applications sur ceux-ci.

Les terminaux Sidewalk communiquent avec eux-mêmes via des passerelles Sidewalk. AWS IoT Core pour Amazon Sidewalk gère les politiques relatives aux services et aux appareils nécessaires à la gestion et à la communication avec les terminaux et les passerelles Sidewalk. Il gère également les destinations qui envoient les données de l'appareil à d'autres services AWS.



Commencez à utiliser AWS IoT Core pour Amazon Sidewalk

Vous pouvez utiliser la AWS IoT console, l'API AWS IoT Core pour Amazon Sidewalk ou le CLI AWS pour créer et intégrer des terminaux Sidewalk et les connecter au réseau Sidewalk. Pour de plus amples informations sur la mise en route avec Amazon Sidewalk et sur la mise en route avec Amazon Sidewalk AWS IoT, veuillez consulter les rubriques suivantes.

- [Démarrer avec AWS IoT Core pour Amazon Sidewalk \(p. 1404\)](#)

Cette rubrique décrit les conditions préalables à l'intégration de vos terminaux Sidewalk, illustre le flux de travail à l'aide d'une application de surveillance des capteurs et fournit une vue d'ensemble de la manière d'intégrer votre appareil à l'aide de AWS CLI commandes.

- [Connexion d'appareils Sidewalk à AWS IoT Core pour Amazon Sidewalk \(p. 1411\)](#)

Cette section décrit les différentes étapes de l'introduction du flux de travail d'intégration, décrit l'intégration de vos appareils finaux à l'aide de la console et les opérations de l'API. Vous allez également connecter votre appareil et consulter les messages échangés entre votre appareil et AWS IoT Core pour Amazon Sidewalk.

- [Approvisionnement en masse d'appareils avec AWS IoT Core pour Amazon Sidewalk \(p. 1426\)](#)

Cette section fournit un step-by-step didacticiel détaillé pour le provisionnement en masse de vos terminaux Sidewalk à l'aide de AWS IoT Core pour Amazon Sidewalk. Vous découvrirez le flux de travail de provisionnement en masse et comment intégrer un grand nombre d'appareils Sidewalk.

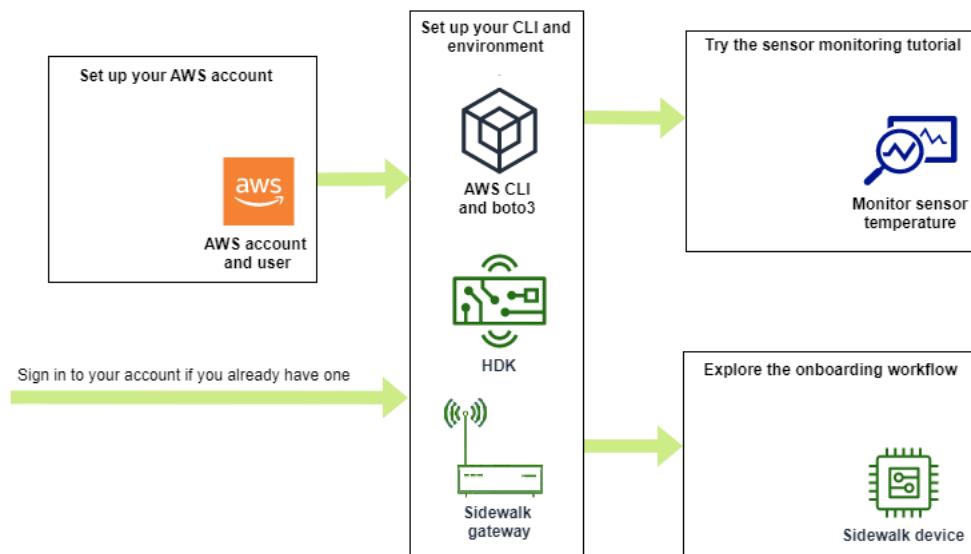
En savoir plus sur AWS IoT Core pour Amazon Sidewalk

Pour plus d'informations sur AWS IoT Core pour Amazon Sidewalk, veuillez consulter les pages web suivantes :

- [Trottoir Amazon](#)
- [Documentation sur Amazon Sidewalk](#)
- [AWS IoT Core pour Amazon Sidewalk](#)

Démarrer avec AWS IoT Core pour Amazon Sidewalk

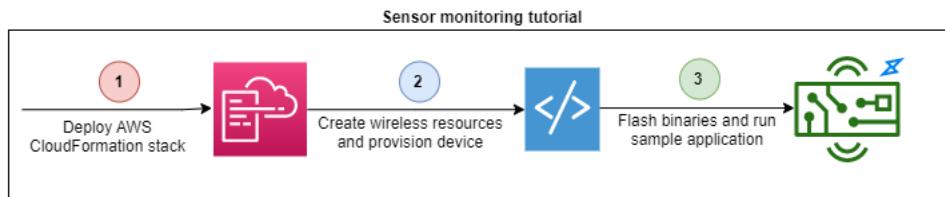
Cette section vous montre comment commencer à connecter vos terminaux Sidewalk à AWS IoT Core pour Amazon Sidewalk. Il explique comment connecter un terminal à Amazon Sidewalk et transmettre des messages entre eux. Vous découvrirez également l'exemple d'application Sidewalk et un aperçu de la manière de surveiller les capteurs à l'aide de AWS IoT Core pour Amazon Sidewalk. L'exemple d'application vous fournit un tableau de bord pour visualiser et surveiller les modifications de la température du capteur.



Essayez le didacticiel de surveillance des capteurs

Cette section fournit un aperçu de l'exemple d'application Amazon Sidewalk GitHub qui vous montre comment surveiller la température d'un capteur. Dans ce didacticiel, vous utilisez des scripts qui créent par programmation les ressources sans fil requises, approvisionnent le terminal et flashent les fichiers binaires, puis connectent votre terminal à l'application. Les scripts qui utilisent les commandes Python AWS CLI et créent une AWS CloudFormation pile et des ressources sans fil, puis flashent les fichiers binaires et déplacent l'application sur votre kit de développement matériel (HDK).

Le schéma suivant montre les étapes à suivre lorsque vous exécutez l'[exemple d'application](#) et connectez votre terminal Sidewalk à l'application. Pour obtenir des instructions détaillées, y compris les prérequis et la configuration de ce didacticiel, consultez le [document README](#) dans GitHub.



Les opérations suivantes sont effectuées dans cette section :

Rubriques

- [Configuration en vue de l'utilisation AWS IoT Core pour Amazon Sidewalk \(p. 1405\)](#)
- [Décrire vos ressources Sidewalk \(p. 1407\)](#)
- [Présentation de l'intégration de vos appareils Sidewalk \(p. 1408\)](#)

Configuration en vue de l'utilisation AWS IoT Core pour Amazon Sidewalk

Avant de connecter votre terminal Sidewalk, vous devez configurer votre compte AWS CLI.

Important

Pour effectuer l'intégralité du processus d'intégration pour le provisionnement et l'enregistrement de votre terminal Sidewalk, vous devez également configurer votre passerelle Sidewalk et le HDK.

Pour obtenir des instructions, consultez [Configuration du kit de développement matériel \(HDK\)](#) et [Configuration d'une passerelle Sidewalk](#) dans la documentation Amazon Sidewalk.

Rubriques

- [Installez Python et Python3-PIP \(p. 1405\)](#)
- [Configuration de votre compte et AWS CLI \(p. 1406\)](#)

Installez Python et Python3-PIP

Pour utiliser AWS CLI et boto3 comme décrit dans la section suivante, vous devez utiliser Python version 3.6 ou ultérieure. Si vous souhaitez intégrer vos terminaux à l'aide de la AWS IoT console, vous pouvez ignorer cette section et continuer à configurer votre compte AWS. Pour vérifier si vous avez déjà installé Python et Python3-PIP, exécutez les commandes suivantes. Si l'exécution de ces commandes renvoie la version, cela signifie que Python et Python3-PIP ont été correctement installés.

```
python3 -V  
pip3 --version
```

Si cette commande renvoie une erreur, c'est peut-être parce que Python n'est pas installé ou que votre système d'exploitation appelle l'exécutable Python v3.x Python3. Dans ce cas, remplacez toutes les instances de python par python3 lorsque vous exécutez les commandes. Si l'erreur persiste, téléchargez et exécutez le programme d'[installation Python](#) ou installez Python en fonction de votre système d'exploitation, comme décrit ci-dessous.

Windows

Sur votre machine Windows, téléchargez Python depuis le [site Web de Python](#), puis exécutez le programme d'installation pour installer Python sur votre machine.

Linux

Sur votre machine Ubuntu, exécutez la commande suivante pour installer Python.

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

macOS

Sur votre ordinateur Mac, utilisez Homebrew pour installer Python. Homebrew installe également pip, qui pointe ensuite vers la version Python3 installée.

```
$ brew install python
```

Configuration de votre compte et AWS CLI

Les étapes suivantes vous montrent comment configurer votreCompte AWSAWS CLI, et boto3 (AWSSDK pour Python).

1. Configurez votreCompte AWS

Pour utiliserAWS IoT Core pour Amazon Sidewalk, créez unCompte AWS utilisateur administratif. Pour des instructions, consultez [Configurez votre Compte AWS \(p. 19\)](#).

2. Installation et configuration de l'AWS CLI

Vous pouvez utiliser leAWS CLI pour intégrer par programme vos terminaux Sidewalk àAWS IoT Core pour Amazon Sidewalk. Si vous souhaitez intégrer vos appareils à l'aide de laAWS IoT console, vous pouvez ignorer cette section. Ouvrez la [AWS IoT Coreconsole](#), puis passez à la section suivante pour commencer à connecter vos appareils àAWS IoT Core pour Amazon Sidewalk. Pour obtenir des instructions sur la configuration duAWS CLI, voir [Installation et configuration duAWS CLI](#).

3. Installez boto3 (AWSSDK pour Python)

Les commandes suivantes vous indiquent comment installer boto3 (AWSSDK pour Python) et leAWS CLI. Vous allez également installer Botocore, qui est requis pour exécuter boto3. Pour obtenir des instructions détaillées, consultez la section [Installation de Boto3](#) dans le guide de documentation de Boto3.

Note

awscliLa version1.26.6 nécessite une version PyYAML 3.10 ou ultérieure, mais pas ultérieure à 5.5.

```
python3 -m pip install botocore-<version>-py3-none-any.whl
python3 -m pip install boto3-<version>-py3-none-any.whl
```

4. Configurez vos informations d'identification et votre région par défaut

Configurez vos informations d'identification et la région par défaut dans les~/.aws/config fichiers~/.aws/credentials et. La bibliothèque boto3 utilise ces informations d'identification pour identifierCompte AWS et autoriser vos appels d'API. Pour obtenir, consultez :

- [Configuration](#) dans le guide de documentation Boto3
- [Paramètres du fichier de configuration et d'informations d'identification](#) dans le guide deAWS CLI documentation

Décrire vos ressources Sidewalk

Avant de commencer à créer des ressources, nous vous recommandons de prendre en compte la convention d'affectation de noms de noms de noms de noms de noms de ressources. Vos ressources sans fil peuvent inclure des terminaux Sidewalk, des profils d'appareils ou des destinations. AWS IoT Core pour Amazon Sidewalk attribue un identifiant unique aux ressources que vous créez. Vous pouvez toutefois leur donner des noms plus descriptifs, ajouter une description ou ajouter des balises facultatives pour les identifier et les gérer.

Les sections suivantes présentent les différentes ressources de Sidewalk et leurs contraintes de longueur.

Rubriques

- [Nom et description des ressources \(p. 1407\)](#)
- [Étiquettes de ressources \(p. 1407\)](#)

Nom et description des ressources

Les ressources de Sidewalk peuvent avoir un nom comportant jusqu'à 256 caractères. Pour les terminaux et les profils Sidewalk, le nom de la ressource est facultatif et peut être modifié une fois la ressource créée. Pour les destinations, indiquez un nom propre à votre Compte AWS et Région AWS. Le nom de destination ne peut pas être modifié une fois que vous l'avez créée. Nous vous recommandons d'utiliser des noms courts et significatifs pour vous aider à identifier votre ressource. Par exemple, si votre terminal Sidewalk est utilisé pour la localisation d'objets, vous pouvez spécifier un nom tel que *Sidewalk_Object_Locator_Device*.

Dans la AWS IoT console, le champ de nom apparaît dans la liste des ressources du hub de ressources. Lorsque vous sélectionnez les noms de vos ressources, réfléchissez à la manière dont vous souhaitez les identifier et les afficher sur la console. L'espace étant limité dans la console, seuls les 15 à 30 premiers caractères peuvent être visibles.

Contraintes relatives au champ « nom du trottoir »

Nom de la ressource	Contraintes de noms de noms de
Destination	Le nom est un identifiant unique qui ne peut pas être modifié.
Terminal Sidewalk	Le nom est un descripteur facultatif qui peut être modifié une fois que la ressource a été créée.
Profil de l'appareil	Le nom est un descripteur facultatif qui peut être modifié une fois que la ressource a été créée.

Les destinations, les terminaux et les profils prennent également en charge un champ de description, qui peut contenir jusqu'à 2 048 caractères. Bien que le champ de description puisse contenir de nombreuses informations, il n'apparaît que sur la page de détails de votre ressource dans la AWS IoT console et peut ne pas être pratique pour une analyse dans le contexte de plusieurs ressources. Pour vous aider à identifier et à gérer vos ressources, nous vous recommandons de fournir une description courte et significative. Par exemple, si votre destination republie les données des capteurs dans une rubrique IoT, vous pouvez spécifier une description telle que *Sidewalk destination to republish sensor data*.

Étiquettes de ressources

Les balises sont des mots ou des expressions qui constituent des métadonnées qui vous permettent d'identifier et d'organiser vos AWS ressources. Les balises vous aident à classer vos ressources. Vous choisissez les clés de balise et les valeurs correspondantes.

Les destinations et les profils peuvent être associés à un maximum de 50AWS balises. Les dispositifs Sidewalk ne sont pas compatibles avec les étiquettes. Par exemple, vous pouvez définir un ensemble d'identifications pour un ensemble d'étiquettes. Par exemple, vous pouvez définir un ensemble d'étiquettes. Pour gérer plus facilement vos ressources, nous vous recommandons de créer un ensemble d'identifications de clés d'balises répondant à vos besoins pour chaque type de ressource. Le tableau suivant affiche les balises prises en charge pour vos ressources Sidewalk.

Limitation des balises de balises de balises

Nom de la ressource	Limites de balises
Destination	Vous pouvez ajouter jusqu'à 50AWS balises à chaque ressource.
Appareil de trottoir	Cette ressource ne prend pas en charge les balises.
Profil de l'appareil	Vous pouvez ajouter jusqu'à 50AWS balises à chaque ressource.

Vous pouvez considérer la clé de balise comme une catégorie d'informations et la valeur de la balise comme une valeur spécifique de cette catégorie. Par exemple, vous pouvez avoir une valeur de balise de, *color* puis attribuer à certaines ressources une valeur de *blue* pour cette balise et à d'autres une valeur de *red*. Vous pouvez ainsi utiliser l'éditeur de balises duAWS Management Console pour rechercher les ressources dont la valeur de balise de couleur est *blue*.

Pour de plus amples informations sur le balisage et les stratégies d'balises, veuillez consulter [Balisage d'étiquettes](#).

Présentation de l'intégration de vos appareils Sidewalk

Cette section vous montre comment intégrer vos terminaux Sidewalk àAWS IoT Core pour Amazon Sidewalk. Pour intégrer vos appareils, ajoutez d'abord votre appareil Sidewalk, puis configurez et enregistrez votre appareil, puis connectez votre matériel à l'application cloud. Avant de lancer ce didacticiel, révisez-le et terminez-le[Configuration en vue de l'utilisationAWS IoT Core pour Amazon Sidewalk \(p. 1405\)](#).

Les étapes suivantes vous montrent comment intégrer et connecter vos terminaux Sidewalk àAWS IoT Core pour Amazon Sidewalk. Si vous souhaitez intégrer des appareils à l'aide duAWS CLI, vous pouvez vous référer aux exemples de commandes fournis dans cette section. Pour plus d'informations sur l'intégration des appareils à l'aide de laAWS IoT console, consultez[Connexion d'appareils Sidewalk àAWS IoT Core pour Amazon Sidewalk \(p. 1411\)](#).

Important

Pour effectuer l'intégralité du processus d'intégration, vous devez également configurer et enregistrer votre terminal, et connecter votre kit de développement matériel (HDK). Pour plus d'informations, consultez la section [Provisionnement et enregistrement de votre terminal](#) dans la documentation Amazon Sidewalk.

Rubriques

- [Étape 1 : Ajouter un appareil Sidewalk àAWS IoT Core pour Amazon Sidewalk \(p. 1409\)](#)
- [Étape 2 : Création d'une destination pour votre appareil Sidewalk \(p. 1410\)](#)
- [Étape 3 : provisionner et enregistrer le terminal \(p. 1410\)](#)
- [Étape 4 : Connect à l'appareil final Sidewalk et échanger des messages \(p. 1410\)](#)

Étape 1 : Ajouter un appareil Sidewalk àAWS IoT Core pour Amazon Sidewalk

Voici une vue d'ensemble des étapes que vous allez suivre pour y ajouter un ensemble d'étiquettes AWS IoT Core pour Amazon Sidewalk. Stockez les informations que vous obtenez concernant le profil de l'appareil et le périphérique sans fil que vous créez. Vous utiliserez ces informations pour configurer et enregistrer le terminal. Pour de plus amples informations sur ces étapes, veuillez consulter [Ajoutez votre appareil àAWS IoT Core pour Amazon Sidewalk \(p. 1412\)](#).

1. Création d'un profil d'appareil

Créez un profil d'appareil contenant les configurations partagées pour vos appareils Sidewalk. Lors de la création du profil, spécifiez un *name* pour le profil sous forme de chaîne alphanumérique. Pour créer un profil, accédez à l'[onglet Sidewalk du hub Profiles](#) de la AWS IoT console et choisissez Créer un profil, ou utilisez l'opération [CreateDeviceProfile](#) API ou la commande [create-device-profile](#) CLI comme indiqué dans cet exemple.

```
// Add your device profile using a name and the sidewalk object.  
aws iotwireless create-device-profile --name sidewalk_profile --sidewalk []
```

2. Créez votre terminal Sidewalk

Créez votre terminal Sidewalk avec AWS IoT Core pour Amazon Sidewalk. Spécifiez un nom de destination et l'ID du profil d'appareil obtenu à l'étape précédente. Pour ajouter un appareil, accédez à l'[onglet Sidewalk du hub Devices](#) de la AWS IoT console et choisissez Provisionner un appareil, ou utilisez l'opération [CreateWirelessDevice](#) API ou la commande [create-wireless-device](#) CLI comme indiqué dans cet exemple.

Note

Spécifiez un nom propre à votre Compte AWS destination Région AWS. Vous utiliserez le même nom de destination lorsque vous ajouterez votre destination à AWS IoT Core pour Amazon Sidewalk.

```
// Add your Sidewalk device by using the device profile ID.  
aws iotwireless create-wireless-device --type "Sidewalk" --name sidewalk_device \  
--destination-name SidewalkDestination \  
--sidewalk DeviceProfileId="12345678-234a-45bc-67de-e8901234f0a1"
```

3. Obtenir le profil de l'appareil et les informations sur l'appareil sans fil

Obtenez le profil de l'appareil et les informations relatives à l'appareil sans fil au format JSON. Le JSON contiendra des informations sur les détails de l'appareil, les certificats de l'appareil DeviceType Id, les clés privées et le numéro de série de fabrication du Sidewalk (SMSN).

- Si vous utilisez la AWS IoT console, vous pouvez utiliser l'[onglet Sidewalk du hub Devices](#) pour télécharger un fichier JSON combiné pour votre terminal Sidewalk.
- Si vous utilisez les opérations d'API, stockez les réponses obtenues à partir des opérations d'API [GetDeviceProfile](#) et [GetWirelessDevice](#) dans des fichiers JSON distincts, tels que *device_profile.json* et *wireless_device.json*.

```
// Store device profile information as a JSON file.  
aws iotwireless get-device-profile \  
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json  
  
// Store wireless device information as a JSON file.  
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \  
--identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

Étape 2 : Création d'une destination pour votre appareil Sidewalk

Voici une vue d'ensemble des étapes que vous allez suivre pour ajouter une destination AWS IoT Core pour Amazon Sidewalk. À l'aide du AWS Management Console, ou des opérations de l'API AWS IoT Wireless AWS CLI, ou du, vous exécutez les étapes suivantes pour créer une AWS IoT règle et une destination. Vous pouvez ensuite vous connecter à la plate-forme matérielle, consulter et échanger des messages. Pour un exemple de rôle et de AWS IoT règle IAM utilisés pour les AWS CLI exemples de cette section, consultez [Créer un rôle IAM et une règle IoT pour votre destination \(p. 1421\)](#).

1. Créez un rôle IAM

Créez un rôle IAM qui AWS IoT Core pour Amazon Sidewalk autorise l'envoi de données à la AWS IoT règle. Pour créer le rôle, utilisez l'opération [CreateRole](#) API ou la commande [create-role](#) CLI. Vous pouvez nommer le rôle comme `SidewalkRole`.

```
aws iam create-role --role-name lambda-ex \
--assume-role-policy-document file://lambda-trust-policy.json
```

2. Création d'une règle pour la destination

Créez une AWS IoT règle qui traitera les données de l'appareil et spécifiera le sujet dans lequel les messages sont publiés. Vous observerez les messages à ce sujet après la connexion à la plate-forme matérielle. Utilisez l'opération d'AWS IoT Core API [CreateTopicRule](#), ou la AWS CLI commande [create-topic-rule](#), pour créer une règle pour la destination.

```
aws iot create-topic-rule --rule-name Sidewalkrule \
--topic-rule-payload file://myrule.json
```

3. Créer une destination

Créez une destination qui associe votre appareil Sidewalk à la règle IoT qui le traite pour l'utiliser avec d'autres services AWS. Vous pouvez ajouter une destination à l'aide du [hub Destinations](#) de la AWS IoT console, de l'opération [CreateDestination](#) API ou de la commande [create-destination](#) CLI.

```
aws iotwireless create-destination --name SidewalkDestination \
--expression-type RuleName --expression SidewalkRule \
--role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

Étape 3 : provisionner et enregistrer le terminal

À l'aide de commandes Python, vous pouvez configurer et enregistrer votre terminal. Le script de provisionnement utilise les données JSON du périphérique que vous avez obtenues pour générer une image binaire de fabrication, qui est ensuite flashée sur la carte mère. Vous enregistrez ensuite votre terminal pour le connecter à la plate-forme matérielle. Pour plus d'informations, consultez la section [Provisionnement et enregistrement de votre terminal](#) dans la documentation Amazon Sidewalk.

Note

Lorsque vous enregistrez votre terminal Sidewalk, votre passerelle doit être activée sur Amazon Sidewalk, et votre passerelle et votre appareil doivent être à portée l'un de l'autre.

Étape 4 : Connect à l'appareil final Sidewalk et échanger des messages

Après avoir enregistré votre terminal, vous pouvez le connecter et commencer à échanger des messages et des données sur l'appareil.

1. Connect votre terminal Sidewalk

Connectez le HDK à votre ordinateur et suivez les instructions fournies dans la documentation du fournisseur pour vous connecter à votre HDK. Pour plus d'informations, consultez la section [Provisionnement et enregistrement de votre terminal](#) dans la documentation Amazon Sidewalk.

2. Afficher et échanger des messages

Utilisez le client MQTT pour vous abonner à la rubrique spécifiée dans la règle et consulter le message reçu. Vous pouvez également utiliser l'opération [SendDataToWirelessDevice](#) API ou la commande [send-data-to-wireless-device](#) CLI pour envoyer un message de liaison descendante à votre appareil et vérifier l'état de la connectivité.

(Facultatif) Vous pouvez activer l'événement d'état de remise du message pour vérifier si le message de liaison descendante a bien été reçu.

```
aws iotwireless send-data-to-wireless-device \
--id "<Wireless_Device_ID>" \
--payload-data "SGVsbgG8gVG8gRGV2c2lt" \
--wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

Connexion d'appareils Sidewalk à AWS IoT Core pour Amazon Sidewalk

Cette section explique comment intégrer votre terminal Sidewalk, puis connecter votre appareil au réseau Sidewalk. Il décrit les étapes que vous effectuez dans le didacticiel d'intégration, comme indiqué dans [Présentation de l'intégration de vos appareils Sidewalk \(p. 1408\)](#). Vous apprendrez à intégrer des appareils à l'aide de la AWS IoT console et des opérations de l'AWS IoT Core pour Amazon Sidewalk API. Vous découvrirez également les AWS CLI commandes qui exécutent ces opérations.

Prérequis

Pour ajouter votre terminal et votre destination à AWS IoT Core pour Amazon Sidewalk, vous devez configurer votre compte AWS. Pour effectuer ces opérations à l'aide de l'API AWS IoT sans fil ou des AWS CLI commandes, vous devez également configurer le AWS CLI. Pour plus d'informations sur les prérequis et la configuration, consultez la section [Configuration en vue de l'utilisation AWS IoT Core pour Amazon Sidewalk \(p. 1405\)](#).

Note

Pour effectuer l'intégralité du flux de travail d'intégration concernant le provisionnement et l'enregistrement de votre terminal, ainsi que la connexion à votre kit de développement matériel (HDK), vous devez également configurer votre passerelle Sidewalk et votre HDK. Pour plus d'informations, consultez [Configuration du kit de développement matériel \(HDK\)](#) et [Configuration d'une passerelle Sidewalk](#) dans la documentation Amazon Sidewalk.

Décrire vos ressources Sidewalk

Avant de commencer et de créer les ressources, nous vous recommandons de prendre en compte la convention de dénomination de vos terminaux Sidewalk, de leurs profils d'appareils et de leurs destinations. AWS IoT Core pour Amazon Sidewalk attribue un identifiant unique aux ressources que vous créez. Vous pouvez toutefois leur donner des noms plus descriptifs, ajouter une description ou ajouter des balises facultatives pour les identifier et les gérer.

Note

Le nom de destination ne peut pas être modifié après avoir été créé. Utilisez un nom propre à votreCompte AWS mainRégion AWS.

Pour plus d'informations, veuillez consulter [Décrire vos ressources Sidewalk \(p. 1407\)](#).

Les rubriques suivantes vous montrent comment intégrer votre appareil Sidewalk àAWS IoT Core pour Amazon Sidewalk

Rubriques

- [Ajoutez votre appareil àAWS IoT Core pour Amazon Sidewalk \(p. 1412\)](#)
- [Ajoutez une destination pour votre terminal Sidewalk \(p. 1419\)](#)
- [Connectez votre appareil Sidewalk et affichez le format de métadonnées Uplink \(p. 1424\)](#)

Ajoutez votre appareil àAWS IoT Core pour Amazon Sidewalk

Avant de créer un appareil sans fil, créez d'abord un profil d'appareil. Les profils des appareils définissent les capacités des appareils et les autres paramètres de vos appareils Sidewalk. Un profil d'appareil unique peut être associé à plusieurs appareils.

Une fois que vous avez créé un profil d'appareil, lorsque vous récupérez des informations le concernant, celui-ci renvoie unDeviceTypeId. Lorsque vous approvisionnez votre terminal, vous utilisez cet identifiant, les certificats de l'appareil, la clé publique du serveur d'applications et le SMSN.

Comment créer et ajouter votre appareil

1. Créez un profil d'appareil pour vos terminaux Sidewalk. Spécifiez un nom de profil à utiliser pour vos appareils Sidewalk sous forme de chaîne alphanumérique. Le profil permettra d'identifier les appareils auxquels l'associer.
 - (Console) Lorsque vous ajoutez votre appareil Sidewalk, vous pouvez également créer un nouveau profil. Cela vous permet d'ajouter rapidement votre appareil à un profilAWS IoT Core pour Amazon Sidewalk et de l'associer à celui-ci.
 - (API) Utilisez l'opérationCreateDeviceProfile d'API en spécifiant un nom de profil et l'objet Sidewalk,sidewalk {}. La réponse de l'API contiendra un ID de profil et un ARN (Amazon Resource Name).
2. Ajoutez votre appareil sans fil àAWS IoT Core pour Amazon Sidewalk. Spécification du nom de destination et choisissez le profil d'appareil créé à l'étape précédente.
 - (Console) Lorsque vous ajoutez votre appareil Sidewalk, entrez un nom de destination et choisissez le profil que vous avez créé.
 - (API) Utilisez l'opérationCreateWirelessDevice API. Spécifiez un nom de destination et l'ID du profil d'appareil obtenu précédemment.

Paramètres de l'appareil Wireless

Paramètre	Description	Remarques
Nom de destination	Le nom de la destination qui décrit lesAWS IoT règles de traitement des données de l'appareil que d'autresService AWS utilisateurs utiliseront.	Si vous n'avez pas encore créé de destination, vous pouvez fournir n'importe quelle valeur de chaîne. AWS IoT Core pour Amazon Sidewalkcréera une destination vide lors de la création de l'appareil, que vous

Paramètre	Description	Remarques
		pourrez ensuite mettre à jour lors de l'ajout de votre destination.
Profil de l'appareil	Le profil d'appareil que vous avez créé précédemment.	–

- Obtenez le fichier JSON contenant les informations requises pour l'approvisionnement de votre appareil final.
 - (Console) Téléchargez ce fichier depuis la page de détails de l'appareil Sidewalk que vous avez créé.
 - (API) Utilisez les opérations de l'GetWirelessDeviceAPIGetDeviceProfile et pour récupérer des informations sur le profil de votre appareil et votre appareil sans fil. Stockez les informations de réponse de l'API sous forme de fichiers JSON, tels que `device_profile.json` et `wireless_device.json`.

Ajoutez le profil de votre appareil et votre terminal Sidewalk

Cette section explique comment créer un profil d'appareil. Il montre également comment utiliser la AWS IoT console et la AWS CLI pour ajouter votre terminal Sidewalk à AWS IoT Core pour Amazon Sidewalk.

Ajouter votre appareil Sidewalk (console)

Pour ajouter votre appareil Sidewalk à l'aide de la AWS IoT console, accédez à l'[onglet Sidewalk du hub Appareils](#), choisissez Provisioner l'appareil, puis effectuez les étapes suivantes.

The screenshot shows the AWS IoT Console interface for provisioning a Sidewalk device. At the top, there are tabs for 'LoRaWAN' and 'Sidewalk', with 'Sidewalk' being active. Below the tabs, there's a section titled 'How it works' with a brief description. The main area is divided into three columns:

- Step 1. Add your Sidewalk device**: An icon of a device with a key. Description: First, create a device profile and retrieve the application server public key. Next, create your Sidewalk device and retrieve information about it, including device certificates and private keys.
- Step 2. Provision & register your Sidewalk device**: An icon of a document with a key. Description: Provision your hardware as a Sidewalk endpoint by flashing the device certificates and the application server public key that you have generated. Register your device so that it can connect to AWS IoT Core for Amazon Sidewalk.
- Step 3. Connect your Sidewalk endpoint to the cloud**: An icon of a cloud with a device. Description: Create a destination and use AWS IoT Rules to process and route data to other AWS services. Your endpoint can now exchange messages with your cloud application.

At the bottom, there's a table titled 'Sidewalk devices (2)'. The table has columns for 'Edit', 'Delete', and 'Provision device'. A red box highlights the 'Provision device' button. The table also includes a search bar and navigation controls.

1. Spécification des détails de l'appareil

Spécifiez les informations de configuration de votre appareil Sidewalk. Vous pouvez également créer un nouveau profil d'appareil ou choisir un profil existant pour votre appareil Sidewalk.

- Spécification du nom de l'appareil et, éventuellement, de sa description. La description peut comporter jusqu'à 2 048 caractères. Ces champs peuvent être modifiés après avoir créé l'appareil.
- Choisissez un profil d'appareil à associer à votre appareil Sidewalk. Si vous avez déjà un profil d'appareil, vous pouvez choisir votre profil. Pour créer un nouveau profil, choisissez Créez un nouveau profil, puis entrez un nom pour le profil.

Note

Pour associer des balises au profil de votre appareil, après avoir créé votre profil, accédez au [hub Profils](#), puis modifiez votre profil pour ajouter ces informations.

- c. Spécifiez le nom de votre destination qui acheminera les messages de votre appareil vers un autreServices AWS. Si vous n'avez pas encore créé de destination, rendez-vous sur le [hub Destinations](#) pour créer votre destination. Vous pouvez ensuite choisir cette destination pour votre appareil Sidewalk. Pour plus d'informations, veuillez consulter [Ajoutez une destination pour votre terminal Sidewalk \(p. 1419\)](#).
 - d. Choisissez Suivant pour continuer à ajouter votre appareil Sidewalk.
2. Associer un appareil Sidewalk à unAWS IoT objet (facultatif)

Vous pouvez éventuellement associer votre appareil Sidewalk à n'importe quelAWS IoT objet. Les objets IoT sont des entrées dans le registre d'appareils AWS IoT. Les choses facilitent la recherche et la gestion de vos appareils. L'association d'un objet à votre appareil permet à celui-ci d'accéder à d'autresAWS IoT Core fonctionnalités.

Pour associer votre appareil à un objet, choisissez Enregistrement automatique de l'objet.

- a. Entrez un nom unique pour l'objet IoT que vous souhaitez associer à votre appareil Sidewalk. Les noms d'objets distinguent les majuscules des minuscules et doivent être uniques à portée deCompte AWS mainRégion AWS.
- b. Fournissez toutes les configurations supplémentaires pour votre objet IoT, par exemple en utilisant un type d'objet ou des attributs consultables pouvant être utilisés pour filtrer à partir d'une liste d'objets.
- c. Choisissez Suivant et vérifiez les informations relatives à votre appareil Sidewalk, puis choisissez Créer.

Ajoutez votre appareil Sidewalk (CLI)

Pour ajouter votre appareil Sidewalk et télécharger les fichiers JSON qui seront utilisés pour approvisionner votre appareil Sidewalk, effectuez les opérations d'API suivantes.

Rubriques

- [Étape 1 : Création d'un profil d'appareil \(p. 1414\)](#)
- [Étape 2 : Ajouter votre appareil Sidewalk \(p. 1415\)](#)

Étape 1 : Création d'un profil d'appareil

Pour créer un profil d'appareil dans votreCompte AWS, utilisez l'opération [CreateDeviceProfile](#)API ou la commande [create-device-profile](#)CLI. Lorsque vous créez le profil de votre appareil, spécifiez le nom et fournissez toutes les balises facultatives sous forme de paires nom-valeur.

Par exemple, la commande suivante crée un profil d'appareil pour vos appareils Sidewalk :

```
aws iotwireless create-device-profile \
--name sidewalk_profile --sidewalk {}
```

L'exécution de cette commande renvoie l'Amazon Resource Name (ARN) et l'ID du profil de l'appareil en sortie.

```
{  
  "DeviceProfileArn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",
```

```
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Étape 2 : Ajouter votre appareil Sidewalk

Pour ajouter votre appareil Sidewalk à votre compte AWS IoT pour Amazon Sidewalk, utilisez l'opération [CreateWirelessDevice](#) API ou la commande [create-wireless-device](#) CLI. Lorsque vous créez votre appareil, spécifiez les paramètres suivants, en plus du nom et de la description facultatifs de votre appareil Sidewalk.

Note

Si vous souhaitez associer votre appareil Sidewalk à un AWS IoT objet, utilisez l'opération [AssociateWirelessDeviceWithThing](#) API ou la commande [associate-wireless-device-with-thing](#) CLI. Pour plus d'informations, veuillez consulter [Associez les terminaux Sidewalk de votre Compte AWS entreprise à un objet IoT \(p. 1443\)](#).

La commande suivante montre un exemple de création d'un appareil Sidewalk :

```
aws iotwireless create-wireless-device \  
  --cli-input-json "file://device.json"
```

L'exemple suivant affiche le contenu du fichier device.json.

Contenu du fichier device.json

```
{  
  "Type": "Sidewalk",  
  "Name": "SidewalkDevice",  
  "DestinationName": "SidewalkDestination",  
  "Sidewalk": {  
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
  }  
}
```

L'exécution de cette commande renvoie l'ID de l'appareil et l'Amazon Resource Name (ARN) en sortie.

```
{  
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-abcd-0123-  
bcde-fabc012345678",  
  "Id": "23456789-abcd-0123-bcde-fabc012345678"  
}
```

Obtenir les fichiers JSON de l'appareil pour le provisionnement

Après avoir ajouté votre appareil Sidewalk à AWS IoT Core pour Amazon Sidewalk, téléchargez le fichier JSON qui contient les informations requises pour approvisionner votre terminal. Vous pouvez récupérer ces informations à l'aide de la AWS IoT console ou du AWS CLI. Pour plus d'informations sur la mise en service de l'appareil, consultez la section [Provisionnement et enregistrement de votre terminal](#) dans la documentation Amazon Sidewalk.

Obtenir un fichier JSON (console)

Pour obtenir le fichier JSON permettant d'approvisionner votre appareil Sidewalk :

1. Accédez au [hub des appareils Sidewalk](#).

2. Choisissez l'appareil auquel vous l'avez ajouté AWS IoT Core pour Amazon Sidewalk pour en afficher les détails.
3. Obtenez le fichier JSON en choisissant Télécharger le fichier JSON de l'appareil sur la page de détails de l'appareil que vous avez ajouté.

Uncertificate.json fichier contenant les informations requises pour le provisionnement de votre terminal sera téléchargé. Voici un exemple de fichier JSON. Il contient les certificats de l'appareil, les clés privées, le numéro de série de fabrication du Sidewalk (SMSN) et leDeviceTypeID.

```
{  
    "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEbbH ... DANKkOKoNT3bUGz+/f/pyTE  
+xMRdIUBZ1Bw==",  
    "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEbbHD ... UiZmntHiUr1GfkT0FMYqRB+Aw==",  
    "metadata": {  
        "devicetypeid": "fe98",  
        "applicationDeviceArn": "arn:aws:iotwireless:us-  
east-1:123456789012:WirelessDevice/897ce68e-3ca2-4ed0-85a2-30b0666c4052",  
        "applicationDeviceId": "897ce68e-3ca2-4ed0-85a2-30b0666c4052",  
        "smsn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A4A",  
        "devicePrivKeyP256R1":  
            "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",  
            "devicePrivKeyEd25519":  
                "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"  
            },  
        "applicationServerPublicKey":  
            "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"  
    }  
}
```

Sur la page de détails de votre appareil Sidewalk, vous trouverez également des informations sur :

- L'ID de l'appareil, son Amazon Resource Name (ARN) et les détails deAWS IoT tout élément associé à l'appareil.
- Le profil de l'appareil et les détails de la destination.
- Heure à laquelle le dernier message de liaison montante a été reçu de l'appareil.
- État qui indique si votre appareil a été approvisionné ou enregistré.

Obtenir un fichier JSON (CLI)

Pour obtenir les fichiers JSON nécessaires au provisionnement de votre terminal Sidewalk à l'aide de l'AWS IoT Core pour Amazon Sidewalk API ou duAWS CLI, enregistrez la réponse de l'API provenant de la récupération des informations relatives au profil de votre appareil et à votre appareil sans fil sous forme de fichiers JSON, par exemple `wireless_device.json` et `device_profile.json` temporairement. Vous les utiliserez pour approvisionner votre appareil Sidewalk.

Ce qui suit comment extraire les fichiers JSON.

Rubriques

- [Étape 1 : Obtenir les informations du profil de l'appareil sous forme de fichier JSON \(p. 1416\)](#)
- [Étape 2 : Obtenir les informations sur l'appareil Sidewalk sous forme de fichier JSON \(p. 1417\)](#)

Étape 1 : Obtenir les informations du profil de l'appareil sous forme de fichier JSON

Utilisez l'opération `GetDeviceProfile` API ou la commande `get-device-profile` CLI pour obtenir des informations sur le profil de l'appareil pour lequel vous avez ajouté à votre compteAWS IoT Core pour Amazon Sidewalk. Pour récupérer des informations sur le profil de votre appareil, spécifiez l'ID du profil.

L'API renverra ensuite des informations sur le profil de l'appareil correspondant à l'identifiant spécifié et à l'identifiant de l'appareil. Vous enregistrez ces informations de réponse sous forme de fichier et vous leur attribuez un nom tel que [device_profile.json](#).

Voici un exemple de commande CLI :

```
aws iotwireless get-device-profile \
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" > device_profile.json
```

L'exécution de cette commande renvoie les paramètres du profil de votre appareil, la clé publique du serveur d'applications et leDeviceTypeID. Ce qui suit montre un fichier JSON qui contient un exemple d'informations de réponse provenant de l'API. Pour plus d'informations sur les paramètres de la réponse de l'API, consultez [GetDeviceProfile](#).

GetDeviceProfileRéponse de l'API (Contenu de [device_profile.json](#))

```
{
  "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-
a1b2-3c45-67d8-e90fa1b2c34d",
  "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",
  "Name": "Sidewalk_profile",
  "LoRaWAN": null,
  "Sidewalk": {
    {
      "ApplicationServerPublicKey": "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",
      "DAKCertificateMetadata": [
        {
          "DeviceTypeId": "fe98",
          "CertificateId": "43564A6D2D50524F544F54595045",
          "FactorySupport": false,
          "MaxAllowedSignature": 1000
        }
      ],
      "QualificationStatus": false
    }
  }
}
```

Étape 2 : Obtenir les informations sur l'appareil Sidewalk sous forme de fichier JSON

Utilisez l'opération [GetWirelessDevice](#)API ou la commande [get-wireless-device](#)CLI pour obtenir des informations sur l'appareil Sidewalk pour lequel vous avez ajouté à votre compte AWS IoT Core pour Amazon Sidewalk. Pour obtenir des informations sur votre terminal, fournissez l'identifiant de l'appareil sans fil que vous avez obtenu lors de l'ajout de votre appareil.

L'API renverra ensuite des informations sur l'appareil correspondant à l'identifiant spécifié et à l'identifiant de l'appareil. Enregistrez ces informations de réponse dans un fichier JSON. Donnez au fichier un nom pertinent comme [wireless_device.json](#).

Voici un exemple d'exécution de la commande à l'aide de la CLI :

```
aws iotwireless get-wireless-device --identifier-type WirelessDeviceId \
--identifier "23456789-abcd-0123-bcde-fabc012345678" > wireless_device.json
```

L'exécution de cette commande renvoie les détails de l'appareil, les certificats de l'appareil, les clés privées et le numéro de série de fabrication du Sidewalk (SMSN). Voici un exemple de sortie de l'exécution de cette commande. Pour plus d'informations sur les paramètres de la réponse de l'API, consultez [GetWirelessDevice](#).

GetWirelessDeviceRéponse de l'API (Contenu de [wireless_device.json](#))

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/23456789-abcd-0123-bcde-fabc012345678",
    "Id": "23456789-abcd-0123-bcde-fabc012345678",
    "DestinationName": "SidewalkDestination",
    "Type": "Sidewalk",
    "Sidewalk": {
        "CertificateId": "4C7438772D50524F544F54595045",
        "DeviceCertificates": [
            {
                "SigningAlg": "Ed25519",
                "Value": "hDdkJw9L2uMCORjImjMHqzNR6nYYh6QKncSl5GthQNl7NKe4ounb5UMQtLjnm7z0UPY0qghCeVOLCBUiqe2ZiMBEW18F+GeltcafZcFKhS+05NPcVNR/FHYaf/cn5iUBrlz/T+0DXvGdwkBkgDyfgoUJgn7JdzFjaneE5qzTWXUbL79i1sXToGGjP8hiD9jJhidPWhIswleydAwg010ZGA4CjzIaSGVM1VtaLB0VDphuUMMBfgAeL8Tdv5LkFIPB3ZX9zt8zzmAuFRzI4MuNjWfIDn0F6AKu37WWU6/QYhZoQrW9D/wndiCcsRGl+ANn367r/HE02Re4D0iCfs9f2rjc4LT1Lkt7g/KW2ii+W+9HYvvY0bBAI+AHz6Cx4j+djabTsrgW2k6NU2zUSM7bdP3z2a2+Z4Wzbji/jYwt/OP8ipsy5Ee4ywXUfcfQ0rK0r0zay6yh27p3I3MZle2oC04J1lqK0VbIQqsXzSSyp6XXS0lhmaGugZ1AAADgZ+gFBeX/ZNN8VJwnsNfgzj4me1hgVjdUo4W9kvx9cr2jHwkC30j/bdBTh1+yBjOC53yH1QK/l1GHrEWiWPnPnE434LRxnWkw8EHD4oieJxC8fkIxkQfj+gHhU79Z+oAAVAAAzsnf9SDIZPoDXF0Tdc9P0qTgld0oXDl2XPaVD4CvvLearr0SLFv+lsNbC4rgZn23MtIBM/7YQmJwmQ+FXRup6Tkubg1hpz04J/09dxg8UiZmntHiUr1GfkTOFMYqRB+Aw=="
            },
            {
                "SigningAlg": "P256r1",
                "Value": "hDdkJw9L2uMCORjImjMHqzNR6nYYh6QKncSl5GthQNnmHmGU8a+S0qDXWwDNT3VSntpbTTQ17cMIusqweQo+JPXXWE1bGh7eaxPGz4ZeF5yM2cqVNUrQr1lX/6LZ+0LuyrcFrLzzB9APi0NIMLqv/Rt7XJssHQs2RPcT1ul/2XVpa6ztULJeQi2JwhTb/k48wbh/EvafG/ibrIBIx9v7/dwGRAPKhq7Uwb9hHnpha8qNOUtjeUdIwJNh9vCBFX9s22t4PdortoFxbXo9C149PDDD4wqUHJGYlCsVX/Sqqj7Aug3h5wdYN6cDgsuu10m0+aBcXBGpkh70xVxlwXkIP+11dt23TkrSUKd0B01sc9Mc/0yEBCzx5RutKBwsefzy014vQX3AHgV7oD/XV73THMgGiDxQ55CPaaxN/prm791VkJ76BSZaBeF+Su6tg0k/eQneklt8Du5uqkyBHvx8MvxsBIMZ73vIFwUrLhjDeq3+n00yQqSBMnxHKU2mAwN3zb2LoLwjPkKN0h1+NNnv99L2pBcNCnhnoBULWn+BgewzYNdWrXyKkp403ZDa4f+5SVWvbY5eyDDXcohvz/0cCtuRjAkzKBCvIjBDnCv1McjVdC03+utizGntfhAo1Rzstn0oRkgVF2WuMT9IrUmzYximuTXUmWtjyFStqgNBZwHWUTlMmjlpLCVzzcs4HPTKr3dazdvEkhwGAAAIFByCjSp/5WHc4AhsgyjMvKCsZQiKgiI8ECwjfxBaSzDy4zYsRl03FC428H1atrFChFCZT0Bqt5LPXD38+vuAUJiP8XqiEdXeqf2mYMJ5ykoDpwkve/cuQfpjzFQlQfvwjBwiJDANKKOkoNT3bUGz+/f/pyTE+xMRdIUBZ1Bw=="
            }
        ],
        "DeviceProfileId": "0ff5b0c6-f149-4498-af34-21993acd52a7",
        "PrivateKeys": [
            {
                "SigningAlg": "Ed25519",
                "Value": "2c24d4572327f23b9bef38097137c29224a9e979081b3d90124ac9dfa477934e"
            },
            {
                "SigningAlg": "P256r1",
                "Value": "38d526f29cfaf142f596deca187bd809ef71bc13435eedc885b63bb825d63def"
            }
        ],
        "SidewalkManufacturingSn": "843764270F4BDAE3023918C89A3307AB3351EA761887A40A9DC4A5E46B6140D9",
        "Status": "PROVISIONED"
    },
    ...
}
```

Étapes suivantes

Stockez les fichiers JSON, `wireless_device.json` et `device_profile.json` temporairement, car vous les utiliserez à l'étape suivante pour configurer et enregistrer votre terminal en vue de la connexion à

la plate-forme matérielle. Pour plus d'informations, consultez la section [Provisionnement et enregistrement de votre terminal](#) dans la documentation Amazon Sidewalk.

Ajoutez une destination pour votre terminal Sidewalk

Utilisez AWS IoT des règles pour traiter les données et les messages de l'appareil et les acheminer vers d'autres services. Vous pouvez également définir des règles pour traiter les messages binaires reçus d'un appareil et les convertir dans d'autres formats afin de faciliter leur utilisation par d'autres services. Les destinations associent votre terminal Sidewalk à la règle qui traite les données de l'appareil à envoyer à d'autres services AWS utilisateurs.

Comment créer et utiliser une destination

1. Créez une AWS IoT règle et un rôle IAM pour la destination. La AWS IoT règle spécifie les règles qui traiteront les données de l'appareil et les achemineront pour qu'elles soient utilisées par d'autres services AWS utilisateurs et par vos applications. Le rôle IAM autorise l'accès à la règle.
2. Créez une destination pour vos appareils Sidewalk à l'aide de l'opération `CreateDestination` API. Spécifiez le nom de destination, le nom de la règle, le nom du rôle et tous les paramètres facultatifs. L'API renvoie un identifiant unique pour la destination, que vous pouvez spécifier lors de l'ajout de votre terminal à celle-ci AWS IoT Core pour Amazon Sidewalk.

Ce qui suit montre comment créer une destination, ainsi qu'une AWS IoT règle et un rôle IAM pour la destination.

Rubriques

- [Créer une destination pour votre appareil Sidewalk \(p. 1419\)](#)
- [Créer un rôle IAM et une règle IoT pour votre destination \(p. 1421\)](#)

Créez une destination pour votre appareil Sidewalk

Vous pouvez ajouter une destination à votre compte AWS IoT Core pour Amazon Sidewalk soit à l'aide du [hub Destinations](#), soit à l'aide du `CreateDestination`. Lorsque vous créez votre destination, spécifiez :

- Un nom unique pour la destination à utiliser pour votre terminal Sidewalk.

Note

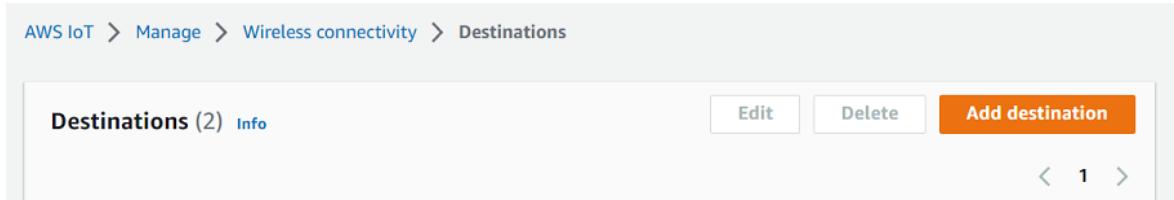
Si vous avez déjà ajouté votre appareil à l'aide d'un nom de destination, vous devez utiliser ce nom lors de la création de votre destination. Pour plus d'informations, veuillez consulter [Étape 2 : Ajouter votre appareil Sidewalk \(p. 1415\)](#).

- Nom de la AWS IoT règle qui traitera les données de l'appareil et sujet dans lequel les messages sont publiés.
- Rôle IAM qui accorde aux données de l'appareil l'autorisation d'accéder à la règle.

Les sections suivantes décrivent comment créer la AWS IoT règle et le rôle IAM pour votre destination.

Créer une destination (console)

Pour créer une destination à l'aide de la AWS IoT console, accédez au [hub Destinations](#) et choisissez Ajouter une destination.



Pour traiter les données d'un appareil, spécifiez les champs suivants lors de la création d'une destination, puis choisissez Ajouter une destination.

- Détails de destination

Entrez un nom de destination et une description facultative pour votre destination.

- Nom de la règle

AWS IoTRègle configurée pour évaluer les messages envoyés par votre appareil et traiter les données de l'appareil. Le nom de la règle sera associé à votre destination. La destination a besoin de la règle pour traiter les messages qu'elle reçoit. Vous pouvez choisir les messages à traiter en invoquant uneAWS IoT règle ou en les publiant sur le courtier deAWS IoT messages.

- Si vous choisissez Entrez un nom de règle, entrez un nom, puis choisissez Copier pour copier le nom de règle que vous allez saisir lors de la création de laAWS IoT règle. Vous pouvez soit choisir Créer une règle pour créer la règle maintenant, soit accéder au hub de [règles](#) de laAWS IoT console et créer une règle portant ce nom.

Vous pouvez également saisir une règle et utiliser le paramètre Avancé pour spécifier un nom de rubrique. Le nom de la rubrique est fourni lors de l'invocation de la règle et est accessible à l'aide de l'topicexpression contenue dans la règle. Pour plus d'informations surAWS IoT les règles, consultez la section [AWS IoTRègles](#).

- Si vous choisissez Publier surAWS IoT Message Broker, entrez un nom de rubrique. Vous pouvez ensuite copier le nom du sujet MQTT et plusieurs abonnés peuvent s'abonner à ce sujet pour recevoir les messages publiés sur ce sujet. Pour plus d'informations, consultez les [rubriques MQTT](#).

Pour plus d'informations surAWS IoT les règles relatives aux destinations, voir [Création de règles pour traiter les messages des périphériques LoRa WAN](#).

- Nom du rôle

Rôle IAM qui autorise les données de l'appareil à accéder à la règle nommée dans Nom de la règle.

Dans la console, vous pouvez créer un nouveau rôle de service ou sélectionner un rôle de service existant. Si vous créez un nouveau rôle de service, vous pouvez soit entrer un nom de rôle (par exemple,**SidewalkDestinationRole**), soit le laisser videAWS IoT Core pour que LoRa WAN génère un nouveau nom de rôle. AWS IoT Corefor LoRa WAN créera alors automatiquement le rôle IAM avec les autorisations appropriées en votre nom.

Créer une destination (CLI)

Pour créer une destination, utilisez l'opération [CreateDestination](#)d'API ou la commande [create-destination](#)CLI. Par exemple, la commande suivante crée une destination pour votre terminal Sidewalk :

```
aws iotwireless create-destination --name SidewalkDestination \
--expression-type RuleName --expression SidewalkRule \
--role-arn arn:aws:iam::123456789012:role/SidewalkRole
```

L'exécution de cette commande renvoie les détails de destination, notamment l'Amazon Resource Name (ARN) et le nom de la destination.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/SidewalkDestination",  
    "Name": "SidewalkDestination"  
}
```

Pour plus d'informations sur la création d'une destination, voir [Création de règles pour traiter les messages des périphériques LoRa WAN](#).

Créer un rôle IAM et une règle IoT pour votre destination

AWS IoTLes règles envoient des messages de l'appareil à d'autres services. AWS IoTLes règles peuvent également traiter les messages binaires reçus d'un terminal Sidewalk pour que d'autres services puissent les utiliser. AWS IoT Core pour Amazon Sidewalkles destinations associent un périphérique sans fil à la règle qui traite les données de message du périphérique pour les envoyer à d'autres services. La règle agit sur les données de l'appareil dèsAWS IoT Core pour Amazon Sidewalk leur réception. Pour tous les appareils qui envoient leurs données au même service, vous pouvez créer une destination qui peut être partagée par tous les appareils. Vous devez également créer un rôle IAM qui autorise l'envoi de données à la règle.

Créer un rôle IAM pour votre destination

Créez un rôle IAM quiAWS IoT Core pour Amazon Sidewalk autorise l'envoi de données à laAWS IoT règle. Pour créer le rôle, utilisez l'opération [CreateRole](#)API ou la commande [create-role](#)CLI. Vous pouvez nommer le rôle comme *SidewalkRole*.

```
aws iam create-role --role-name SidewalkRole \  
    --assume-role-policy-document '{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "lambda.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }'
```

Vous pouvez également définir la politique d'approbation pour le rôle à l'aide d'un fichier JSON.

```
aws iam create-role --role-name SidewalkRole \  
    --assume-role-policy-document file://trust-policy.json
```

L'exemple suivant affiche le contenu du fichier JSON.

Contenu de trust-policy.json

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

Créez une règle pour votre destination

Utilisez l'opération d'AWS IoT CoreAPI [CreateTopicRule](#), ou laAWS CLI commande [create-topic-rule](#), pour créer une règle. La règle thématique sera utilisée par votre destination pour acheminer les données reçues de votre terminal Sidewalk vers un autreServices AWS. Par exemple, vous pouvez créer

une action de règle qui envoie un message à une fonction Lambda. Vous pouvez définir la fonction Lambda de telle sorte qu'elle reçoive les données d'application de votre appareil et utilise la base 64 pour décoder les données de charge utile afin qu'elles puissent être utilisées par d'autres applications.

Les étapes suivantes montrent comment créer la fonction Lambda, puis une règle de rubrique qui envoie un message à cette fonction.

1. Créer un rôle et une stratégie d'exécution

Créez un rôle IAM qui accorde à votre fonction l'autorisation d'accéder aux AWS ressources. Vous pouvez également définir la politique d'approbation pour le rôle à l'aide d'un fichier JSON.

```
aws iam create-role --role-name lambda-ex \
--assume-role-policy-document file://lambda-trust-policy.json
```

L'exemple suivant affiche le contenu du fichier JSON.

Contenu du lambda-trust-policy fichier .json

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

2. Crédit et test d'une fonction Lambda

Procédez comme suit pour créer une AWS Lambda fonction qui décode les données de la charge utile en base64.

- Écrivez le code pour décoder les données de la charge utile. Par exemple, vous pouvez utiliser l'exemple de code Python suivant. Spécifiez un nom pour le script, tel que *base64_decode.py*.

Contenu du fichier *base64_decode.py*

```
// -----
// ----- Python script to decode incoming binary payload -----
// -----
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))
```

- Créez un package de déploiement sous la forme d'un fichier zip contenant le fichier Python et nommez-le ainsi *base64_decode.zip*. Utilisez l'CreateFunctionAPI ou la commande *create-function* CLI pour créer une fonction Lambda pour l'exemple de code, *base64_decode.py*.

c.

```
aws lambda create-function --function-name my-function \
--zip-file fileb://base64_decode.zip --handler index.handler \
--runtime python3.9 --role arn:iam::123456789012:role/Lambda-ex
```

Vous devriez voir la sortie suivante. Vous utiliserez la valeur Amazon Resource Name (ARN) issue de la sortie lorsque vous créez la règle du sujet.FunctionArn

```
{  
    "FunctionName": "my-function",  
    "FunctionArn": "arn:aws:lambda:us-east-1:123456789012:function:my-function",  
    "Runtime": "python3.9",  
    "Role": "arn:aws:iam::123456789012:role/Lambda-ex",  
    "Handler": "index.handler",  
    "CodeSha256": "FpFMvUhayLkOoVBpNuNiIVML/tuGv2iJQ7t0yWVTU8c=",  
    "Version": "$LATEST",  
    "TracingConfig": {  
        "Mode": "PassThrough"  
    },  
    "RevisionId": "88ebe1e1-bfdf-4dc3-84de-3017268fa1ff",  
    ...  
}
```

- d. Pour obtenir les journaux pour un appel à partir de la ligne de commande, utilisez l'--log-type option associée à la `invoke` commande. La réponse inclut un LogResult champ qui contient jusqu'à 4 Ko de journaux codés en base64 provenant de l'appel.

```
aws lambda invoke --function-name my-function out --log-type Tail
```

Vous devez recevoir une réponse StatusCode de 200. Pour plus d'informations sur la création et l'utilisation de fonctions Lambda à partir du AWS CLI, consultez la section [Utilisation de Lambda avec AWS CLI](#).

3. Créer une règle de rubrique

Utilisez l'`CreateTopicRule` API ou la commande `create-topic-rule` CLI pour créer une règle de rubrique qui envoie un message à cette fonction Lambda. Vous pouvez également ajouter une deuxième action de règle qui sera republiée dans une AWS IoT rubrique. Nommez cette règle de rubrique comme suit *Sidewalkrule*.

```
aws iot create-topic-rule --rule-name Sidewalkrule \
--topic-rule-payload file://myrule.json
```

Vous pouvez utiliser le `myrule.json` fichier pour spécifier plus de détails sur la règle. Par exemple, le fichier JSON suivant montre comment republier dans une AWS IoT rubrique et envoyer un message à une fonction Lambda.

```
{  
    "sql": "SELECT * ",  
    "actions": [  
        {  
            // You obtained this functionArn when creating the Lambda function using  
            the  
            // create-function command.  
            "lambda": {  
                "functionArn": "arn:aws:lambda:us-east-1:123456789012:function:my-  
                function"  
            }  
        },  
        {  
    }
```

```
        // This topic can be used to observe messages exchanged between the device
        and
        // AWS IoT Core pour Amazon Sidewalk after the device is connected.
        "republish": {
            "roleArn": "arn:aws:iam::123456789012:role/service-
role/SidewalkRepublishRole",
            "topic": "project/sensor/observed"
        }
    ],
}
```

Connect votre appareil Sidewalk et affichez le format de métadonnées Uplink

Dans ce didacticiel, vous allez utiliser le client de test MQTT pour tester la connectivité et voir les messages échangés entre votre terminal et leAWS Cloud. Pour recevoir des messages, dans le client de test MQTT, abonnez-vous à la rubrique spécifiée lors de la création de la règle IoT pour votre destination. Vous pouvez également envoyer un message de liaison descendante depuis votre appareilAWS IoT Core pour Amazon Sidewalk à l'aide de l'opérationSendDataToWirelessDevice API. Vous pouvez vérifier que le message a été remis en activant la notification d'événement d'état de remise du message.

Note

Pour plus d'informations sur la connexion de votre plate-forme matérielle et sa configuration, consultez [Provisionnement et enregistrement de votre terminal](#) et [Configuration du kit de développement matériel \(HDK\)](#) dans la documentation Amazon Sidewalk.

Envoyez des messages en liaison descendante à votre terminal

Utilisez l'opération [SendDataToWirelessDevice](#)API ou la commande [send-data-to-wireless-device](#)CLI pour envoyer des messages de liaison descendante depuis AWS IoT Core pour Amazon Sidewalk votre terminal Sidewalk. Ce qui suit un exemple de l'exécution de cette commande. Les données utiles sont le binaire à envoyer, codé en base64.

```
aws iotwireless send-data-to-wireless-device \
--id "<Wireless_Device_ID>" \
--payload-data "SGVsbG8gVG8gRGV2c2lt" \
--wireless-metadata Sidewalk={Seq=1,AckModeRetryDurationSecs=10}
```

Ce qui suit montre un exemple de sortie de l'exécution de cette commande, qui est un identifiant du message de liaison descendante envoyé au périphérique.

```
{
    MessageId: "6011dd36-0043d6eb-0072-0008"
}
```

Note

L'[SendDataToWirelessDevice](#)API peut renvoyer un identifiant de message, mais le message risque de ne pas être livré correctement. Pour vérifier l'état du message envoyé à l'appareil, vous pouvez activer les événements d'état de remise des messages pour vos comptes et appareils Sidewalk. Pour plus d'informations sur l'activation de cet événement, consultez la section [Activer les événements pour les appareils Sidewalk \(p. 1449\)](#). Pour plus d'informations sur ce type d'événement, consultez la section [Événements de remise de messages](#).

Afficher le format des messages de liaison montante depuis l'appareil

Une fois que vous avez connecté votre appareil, vous pouvez vous abonner à la rubrique (par exemple [project/sensor/observed](#)) que vous avez spécifiée lors de la création de la règle de destination et consulter les messages en provenance de l'appareil.

Si vous avez indiqué un nom de rubrique lors de la création de votre destination, vous pouvez vous abonner à la rubrique pour surveiller les messages en provenance de votre terminal. Accédez au [client de test MQTT](#) sur la page Test de la AWS IoT console, entrez le nom de la rubrique (par exemple, [project/sensor/observed](#)), puis choisissez S'abonner.

L'exemple suivant affiche le format des messages de liaison montante envoyés par les appareils Sidewalk à AWS IoT. WirelessMetadata contient les métadonnées relatives à la demande de message.

```
{  
    "PayloadData": "ZjRlNjY1ZWNlNw==",  
    "WirelessDeviceId": "wireless_device_id",  
    "WirelessMetadata": {  
        "Sidewalk": {  
            "CmdExStatus": "Cmd",  
            "SidewalkId": "device_id",  
            "Seq": 0,  
            "MessageType": "messageType"  
        }  
    }  
}
```

Le tableau suivant présente une définition des différents paramètres des métadonnées de liaison montante. **device-id** Il s'agit de l'identifiant du périphérique sans fil, tel que [ABCDEF1234](#) et **messageType** type de message de liaison montante reçu de l'appareil.

Paramètres des métadonnées de Sidewalk Uplink

Paramètre	Description	Type	Obligatoire
PayloadData	Charge utile du message envoyé depuis le périphérique sans fil.	Chaîne	Oui
WirelessDeviceID	L'identifiant de l'appareil sans fil qui envoie les données	Chaîne	Oui
Sidewalk.CmdExStatus	État d'exécution de la commande. Les messages de type réponse doivent inclure 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, qui peut être RADIO_TX_ERROR ou MEMORY_ERROR.	Tableau de chaînes	Non

Approvisionnement en masse d'appareils avec AWS IoT Core pour Amazon Sidewalk

Vous pouvez utiliser le provisionnement en masse pour intégrer un grand nombre de terminaux AWS IoT Core pour Amazon Sidewalk en masse. Le provisionnement en masse est particulièrement utile lorsque vous fabriquez un grand nombre d'appareils dans une usine et que vous souhaitez y intégrer ces appareils AWS IoT. Pour plus d'informations sur la fabrication d'appareils, consultez la section [Fabrication d'appareils Amazon Sidewalk](#) dans la documentation Amazon Sidewalk.

Les rubriques suivantes vous montrent comment fonctionne le provisionnement en bloc.

- [Flux de travail de provisionnement en masse d'Amazon Sidewalk \(p. 1426\)](#)

Cette rubrique présente certains concepts clés du provisionnement en masse et explique son fonctionnement. Il indique également les étapes à suivre pour que vos appareils Sidewalk puissent être importés AWS IoT Core pour Amazon Sidewalk.

- [Création de profils d'appareils avec prise en charge en usine \(p. 1429\)](#)

Cette rubrique explique la création d'un profil d'un appareil et sa prise en charge en usine. Vous apprendrez également à récupérer la clé YubiHSM et à l'envoyer à votre fabricant pour obtenir le journal de contrôle une fois les appareils fabriqués.

- [Provisionnement d'appareils Sidewalk à l'aide de tâches d'importation \(p. 1432\)](#)

Cette rubrique explique comment approvisionner en bloc vos appareils Sidewalk en créant et en utilisant des tâches d'importation. Vous apprendrez également à mettre à jour ou à supprimer vos tâches d'importation et à consulter l'état de la tâche d'importation et des appareils qu'elle contient.

Rubriques

- [Flux de travail de provisionnement en masse d'Amazon Sidewalk \(p. 1426\)](#)
- [Création de profils d'appareils avec prise en charge en usine \(p. 1429\)](#)
- [Provisionnement d'appareils Sidewalk à l'aide de tâches d'importation \(p. 1432\)](#)

Flux de travail de provisionnement en masse d'Amazon Sidewalk

Les sections suivantes vous montrent les concepts clés du provisionnement en masse et son fonctionnement. Les étapes impliquées dans le provisionnement en masse sont les suivantes :

1. Créez un profil d'appareil à l'aide de AWS IoT Core pour Amazon Sidewalk.
2. Demandez à l'équipe Amazon Sidewalk de vous fournir une clé YubiHSM et de mettre à jour le profil de votre appareil avec le support d'usine.
3. Envoyez la clé YubiHSM à votre fabricant afin qu'il AWS IoT Core pour Amazon Sidewalk puisse obtenir le journal de contrôle une fois les appareils fabriqués.
4. Créez une tâche d'importation et fournissez les numéros de série (SMSN) des appareils auxquels vous souhaitez les intégrer AWS IoT Core pour Amazon Sidewalk.

Composants du provisionnement en masse

Les concepts suivants vous présentent certains composants clés du provisionnement en masse et comment les utiliser dans le cadre du provisionnement en masse de vos appareils Sidewalk.

Clé YubiHSM

Amazon crée un ou plusieurs HSM (modules de sécurité matériels) pour chacun de vos produits Sidewalk. Chaque HSM possède un numéro de série unique, appelé clé YubiHSM, qui est imprimé sur le module matériel. Cette clé peut être achetée sur le [site Web de Yubico](#).

La clé est unique à chaque HSM et liée à chaque profil d'appareil que vous créez AWS IoT Core pour Amazon Sidewalk. Pour obtenir la clé YubiHSM, contactez l'équipe Amazon Sidewalk. Si vous envoyez la clé YubiHSM au fabricant, une fois les appareils Sidewalk fabriqués en usine, vous AWS IoT Core pour Amazon Sidewalk recevrez un fichier journal de contrôle contenant les numéros de série des appareils. Il compare ensuite ces informations avec votre fichier CSV d'entrée pour l'intégration des appareils AWS IoT.

Clé d'attestation de l'appareil (DAK)

Lorsqu'un terminal Sidewalk rejoint le réseau Sidewalk, il doit être doté d'un certificat d'appareil Sidewalk. Les certificats utilisés pour configurer votre appareil incluent un certificat privé spécifique à l'appareil et les certificats d'appareil publics, qui correspondent à la chaîne de certificats Sidewalk. Lorsque vos appareils Sidewalk sont fabriqués, le YubiHSM signe les certificats des appareils.

Vous trouverez ci-dessous un exemple de fichier JSON contenant les certificats de l'appareil et les clés privées. Pour plus d'informations, veuillez consulter [Obtenir les fichiers JSON de l'appareil pour le provisionnement \(p. 1415\)](#).

```
{  
    "p256R1": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbH ... DANKk0KoNT3bUGz+/f/pyTE+xMRdIUBZ1Bw==",  
    "eD25519": "grg8izXoVvQ86cPVm0GMyWuZYHEBbbHD ... UizmntHiUr1GfkT0FMYqRB+Aw==",  
    "metadata": {  
        "devicetypeid": "fe98",  
        ...  
        "devicePrivKeyP256R1":  
        "3e704bf8d319b3a475179f1d68c60737b28c708f845d0198f2d00d00c88ee018",  
        "devicePrivKeyEd25519":  
        "17dacb3a46ad9a42d5c520ca5f47f0167f59ce54d740aa13918465faf533b8d0"  
    },  
    "applicationServerPublicKey":  
    "5ce29b89c2e3ce6183b41e75fe54e45f61b8bb320efbdd2abd7aefa5957a316b"  
}
```

La clé d'attestation de l'appareil (DAK) est une clé privée que vous obtenez lors de la création du profil de votre appareil. Il correspond au certificat du produit, qui est un certificat unique délivré à chaque produit Sidewalk. Lorsque vous contactez l'équipe Amazon Sidewalk, vous recevez la chaîne de certificats Sidewalk, la clé YubiHSM et un HSM approvisionné avec la clé d'attestation du produit (DAK).

Le profil de votre appareil est également mis à jour à l'aide de la nouvelle clé d'attestation de l'appareil (DAK) et lorsque le support en usine est activé. Les informations de métadonnées DAK du profil de l'appareil fournissent des informations telles que le nom DAK, l'identifiant du certificat, le ApId (identifiant du produit annoncé), si le support d'usine est activé et le nombre maximum de signatures que le DAK peut signer.

Identifiant du produit annoncé (ApId)

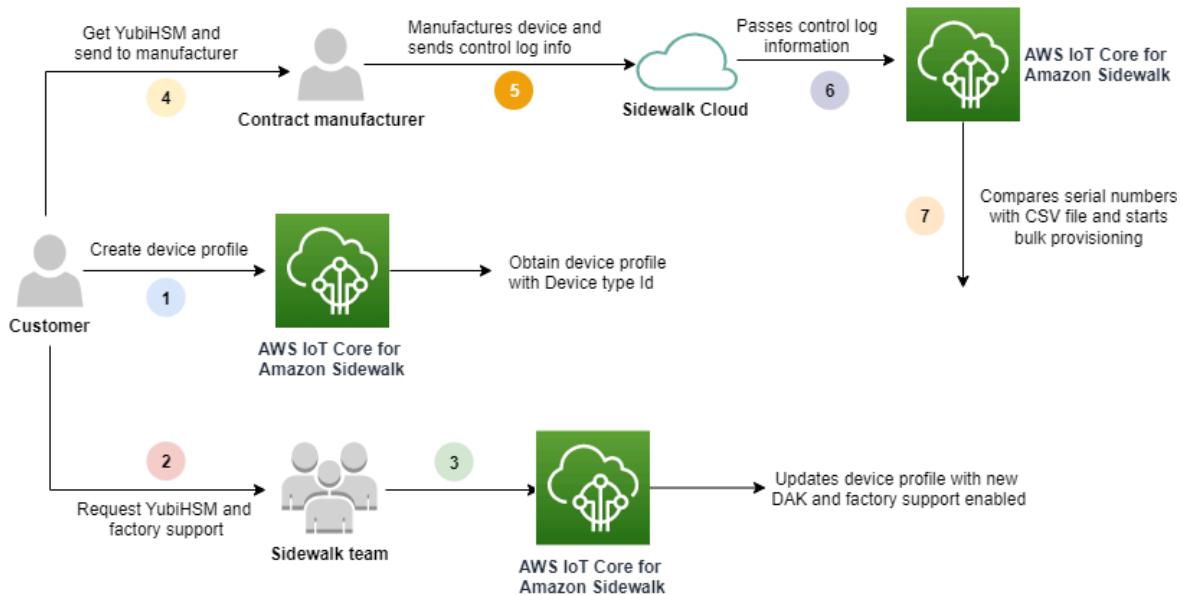
Le ApId paramètre est une chaîne alphanumérique qui identifie le produit annoncé. Ce champ doit être spécifié lorsque vous souhaitez utiliser un profil d'appareil donné pour les appareils Sidewalk que vous approvisionnez en bloc. AWS IoT Core pour Amazon Sidewalk génère ensuite le DAK et vous le fournit via la clé YubiHSM. Les informations DAK associées seront présentées dans le profil de l'appareil.

Pour obtenir le ApId, après avoir récupéré les informations relatives au profil d'appareil que vous avez créé, contactez l'équipe d'Support Amazon Sidewalk. Vous pouvez obtenir les informations de profil de l'appareil

à partir de la AWS IoT console, à l'aide de l'opération d'[GetDeviceProfile](#)API ou de la commande [get-device-profile](#)CLI.

Comment fonctionne le provisionnement en masse

Cet organigramme montre comment fonctionne la fourniture groupée AWS IoT Core pour Amazon Sidewalk.



La procédure suivante illustre les différentes étapes du processus de provisionnement en bloc.

1. Créer un profil d'appareil pour l'appareil Sidewalk

Avant de mettre votre terminal en usine, créez d'abord un profil d'appareil. Vous pouvez utiliser ce profil pour approvisionner des appareils individuels comme décrit dans [Ajoutez le profil de votre appareil et votre terminal Sidewalk \(p. 1413\)](#).

2. Demandez l'assistance de l'usine pour votre profil

Lorsque vous êtes prêt à mettre votre appareil en usine, demandez à l'équipe Amazon Sidewalk de vous fournir la clé YubiHSM et de demander l'assistance en usine pour le profil de votre appareil.

3. Obtenir le DAK et le profil pris en charge par l'usine

L'équipe d'assistance d'Amazon Sidewalk mettra ensuite à jour le profil de votre appareil à l'aide de la clé d'attestation du produit (DAK) et du support technique du fabricant. Le profil de votre appareil sera automatiquement mis à jour avec un identifiant de produit annoncé (APID), ainsi qu'un nouveau DAK et des informations de certificat, telles que l'identifiant du certificat. Les appareils Sidewalk qui utilisent ce profil sont autorisés à être utilisés avec un approvisionnement en masse.

4. Envoyer la clé YubiHSM au fabricant (CM)

Votre appareil final est désormais qualifié. Vous pouvez donc envoyer votre clé YubiHSM au fabricant sous contrat (CM) pour démarrer le processus de fabrication. Pour plus d'informations, consultez la section [Fabrication d'appareils Amazon Sidewalk](#) dans la documentation Amazon Sidewalk.

5. Fabriquez des appareils et envoyez des journaux de contrôle et des numéros de série

Le CM fabrique les appareils et génère des journaux de contrôle. Le CM vous fournit également un fichier CSV contenant une liste des appareils à fabriquer et leurs numéros de série Sidewalk Manufacturing (SMSN). Le code suivant montre un exemple de journal de contrôle. Il contient les numéros de série de l'appareil, l'APID et les certificats publics de l'appareil.

```
{  
    "controlLogs": [  
        {  
            "version": "4-0-1",  
            "device": {  
                "serialNumber": "device1",  
                "productIdentifier": {  
                    "advertisedProductId": "abCD"  
                },  
                "sidewalkData": {  
                    "SidewalkED25519CertificateChain": "...",  
                    "SidewalkP256R1CertificateChain": "..."  
                }  
            }  
        }  
    ]  
}
```

6. Transmettez les informations du journal de contrôle àAWS IoT Core pour Amazon Sidewalk

Le cloud Amazon Sidewalk récupère les informations du journal de contrôle auprès du fabricant et transmet ces informations àAWS IoT Core pour Amazon Sidewalk. Les appareils peuvent ensuite être créés avec leurs numéros de série.

7. Vérifiez la correspondance des numéros de série et lancez le provisionnement groupé

Lorsque vous utilisez laAWS IoT console ou l'opérationAWS IoT Core pour Amazon Sidewalk APIStartWirelessDeviceImportTask,AWS IoT Core pour Amazon Sidewalk comparez le numéro de série de fabrication (SMSN) de chaque appareil obtenu auprès d'Amazon Sidewalk avec les numéros de série correspondants de votre fichier CSV. Si ces informations correspondent, le processus de provisionnement en bloc démarre et les appareils vers lesquels importer sont créésAWS IoT Core pour Amazon Sidewalk.

Création de profils d'appareils avec prise en charge en usine

Avant de pouvoir approvisionner en bloc vos appareils Amazon Sidewalk, vous devez créer un profil d'appareil, puis contacter l'équipe d'assistance Amazon Sidewalk pour demander une assistance en usine. L'équipe Amazon Sidewalk mettra ensuite à jour le profil de votre appareil à l'aide d'une nouvelle clé d'attestation d'appareil (DAK) et y ajoutera le support d'usine. Les appareils Sidewalk qui utilisent ce profil sont ensuite qualifiés pour être utilisés avecAWS IoT Core pour Amazon Sidewalk et peuvent être intégrés pour un approvisionnement en masse.

Les étapes suivantes vous montrent la création d'un profil de périphérique pris en charge en usine.

1. Création d'un profil d'un appareil

Créez d'abord un profil d'appareil. Lorsque vous créez un profil, spécifiez un nom et des balises facultatives sous forme de paires nom-valeur. Pour plus d'informations sur les paramètres requis, ainsi que sur la création et l'utilisation de profils, consultez[Comment créer et ajouter votre appareil \(p. 1412\)](#).

2. Obtenir l'assistance du fabricant pour le profil

Obtenez ensuite le support d'usine pour votre profil d'appareil afin que les appareils utilisant ce profil puissent être qualifiés. Pour vous qualifier, créez un ticket auprès de l'équipe Amazon Sidewalk. Une fois confirmé par l'équipe, vous recevrez un Apld (identifiant de produit annoncé) et votre profil sera

mis à jour à l'aide d'un DAK émis par l'usine. Les terminaux de trottoir qui utilisent ce profil seront qualifiés.

Vous pouvez créer un profil d'appareil à l'aide de laAWS IoT console, des opérations de l'AWS IoT Core pour Amazon SidewalkAPI ou duAWS CLI.

Rubriques

- [Création d'un profil \(console\) \(p. 1430\)](#)
- [Création d'un profil \(CLI\) \(p. 1431\)](#)
- [Étapes suivantes \(p. 1432\)](#)

Création d'un profil (console)

Pour créer un profil d'appareil à l'aide de laAWS IoT console, accédez à l'[onglet Sidewalk du hub Profiles](#) et choisissez Créer un profil.

Name	Profile ID	Qualification status
New_profile3	b627bc56-97c3-475e-90b7-b...	Not Qualified

Pour créer un profil, spécifiez les champs suivants, puis choisissez Soumettre.

- Nom

Entrez un nom pour votre profil.

- Étiquettes

Entrez des balises facultatives sous forme de paires nom-valeur pour vous aider à identifier plus facilement votre profil. Les balises facilitent également le suivi des frais de facturation.

Afficher les informations du profil et qualifier les profils

Vous verrez le profil que vous avez créé dans le [hub de profils](#). Choisissez le profil pour en afficher les détails. Vous y trouverez des informations sur :

- Le nom du profil et l'identifiant unique de l'appareil, ainsi que toutes les balises facultatives que vous avez spécifiées sous forme de paires nom-valeur.
- La clé publique du serveur d'applications et l'ID de type d'appareil du profil.
- Le statut de qualification, qui indique que vous utilisez un profil d'appareil qui n'est pas pris en charge en usine. Pour qualifier le profil de votre appareil afin qu'il soit compatible en usine, contactez l'Support Amazon Sidewalk.
- Informations relatives à la clé d'attestation de l'appareil (DAK). Une fois le profil de votre appareil qualifié, un nouveau DAK sera émis et votre profil sera automatiquement mis à jour avec les nouvelles informations DAK.

Création d'un profil (CLI)

Pour créer un profil d'appareil, utilisez l'opération [CreateDeviceProfile](#)API ou la commande [create-device-profile](#)CLI. Par exemple, la commande suivante crée un profil pour votre terminal Sidewalk.

```
aws iotwireless create-device-profile \
--name sidewalk_device_profile --sidewalk {}
```

L'exécution de cette commande renvoie les détails du profil, qui incluent l'ARN et l'ID du profil.

```
{  
    "DeviceProfileArn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "DeviceProfileId": "12345678-a1b2-3c45-67d8-e90fa1b2c34d"  
}
```

Afficher les informations du profil et qualifier les profils

Utilisez l'opération [GetDeviceProfileAPI](#) ou la commande `get-device-profile` CLI pour obtenir des informations sur le profil de l'appareil pour lequel vous avez ajouté à votre compte AWS IoT Core pour Amazon Sidewalk. Pour récupérer des informations sur le profil de votre appareil, spécifiez l'ID du profil. L'API renverra ensuite des informations sur le profil de l'appareil correspondant à l'identifiant spécifié.

```
aws iotwireless get-device-profile \
--id "12345678-234a-45bc-67de-e8901234f0a1" > device_profile.json
```

L'exécution de cette commande renvoie les paramètres du profil de votre appareil, la clé publique du serveur d'applicationsDeviceTypeDeviceIdApId, l'état de qualification et les DAKCertificate informations.

Dans cet exemple, l'état de qualification et les informations DAK indiquent que le profil de votre appareil n'est pas qualifié. Pour qualifier votre profil, contactez l'Support Amazon Sidewalk, et votre profil recevra un nouveau DAK sans limite d'appareils.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Name": "Sidewalk_profile",  
    "LoRaWAN": null,  
    "Sidewalk":  
    {  
        "ApplicationServerPublicKey":  
        "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",  
        "DAKCertificateMetadata": [  
            {  
                "DeviceTypeId": "fe98",  
                "CertificateId": "43564A6D2D50524F544F54595045",  
                "FactorySupport": false,  
                "MaxAllowedSignature": 1000  
            }  
        ],  
        "QualificationStatus": false  
    }  
}
```

Une fois que l'équipe d'Support d'Amazon Sidewalk aura confirmé ces informations, vous recevrez l'APID et un DAK compatible en usine, comme indiqué dans l'exemple suivant.

Note

Le MaxAllowedSignature symbole of -1 indique que le DAK n'a pas de limite d'appareils. Pour plus d'informations sur les paramètres DAK, voir [DAKCertificateMetadata](#).

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d",  
    "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
    "Name": "Sidewalk_profile",  
    "LoRaWAN": null,  
    "Sidewalk":  
    {  
        "ApplicationServerPublicKey":  
        "a123b45c6d78e9f012a34cd5e6a7890b12c3d45e6f78a1b234c56d7e890a1234",  
        "DAKCertificateMetadata": [  
            {  
                "ApiId": "GZBd",  
                "CertificateId": "43564A6D2D50524F544F54595045",  
                "FactorySupport": true,  
                "MaxAllowedSignature": -1  
            }  
        ],  
        "QualificationStatus": true  
    }  
}
```

Étapes suivantes

Maintenant que vous avez créé un profil d'appareil doté d'un DAK compatible en usine, fournissez à votre fabricant la clé YubiHSM que vous avez obtenue auprès de l'équipe. Vos appareils seront ensuite fabriqués en usine et les informations du journal de contrôle seront ensuite transmises à Amazon Sidewalk, qui contient les numéros de série (SMSN) des appareils. Pour plus d'informations sur ce flux de travail, consultez la section [Fabrication d'appareils Amazon Sidewalk](#) dans la documentation Amazon Sidewalk.

Vous pouvez ensuite approvisionner en bloc vos appareils Sidewalk en fournissant AWS IoT Core pour Amazon Sidewalk les numéros de série des appareils à intégrer. Lorsqu'il AWS IoT Core pour Amazon Sidewalk reçoit le journal de contrôle, il compare les numéros de série du journal de contrôle avec les numéros de série que vous avez fournis. Si les numéros de série correspondent, la tâche d'importation commence à intégrer vos appareils à AWS IoT Core pour Amazon Sidewalk. Pour plus d'informations, veuillez consulter [Provisionnement d'appareils Sidewalk à l'aide de tâches d'importation \(p. 1432\)](#).

Provisionnement d'appareils Sidewalk à l'aide de tâches d'importation

Cette section explique comment approvisionner des appareils Sidewalk en masse à l'aide de la AWS IoT console, des opérations de l'AWS IoT Core pour Amazon Sidewalk API ou du AWS CLI. Les sections suivantes vous montrent comment approvisionner vos appareils Sidewalk.

Rubriques

- [Comment fonctionne l'approvisionnement groupé de Sidewalk \(p. 1433\)](#)
- [Considérations clés relatives à l'approvisionnement en vrac de Sidewalk \(p. 1433\)](#)
- [Format de fichier CSV \(p. 1434\)](#)
- [Comment utiliser le provisionnement groupé de Sidewalk \(p. 1434\)](#)
- [Fourniture de dispositifs de trottoir en vrac \(p. 1435\)](#)
- [Afficher la tâche d'importation et l'état d'intégration de l'appareil \(p. 1439\)](#)

Comment fonctionne l'approvisionnement groupé de Sidewalk

Les étapes suivantes illustrent le fonctionnement du provisionnement en masse.

1. Démarrage de la tâche d'importation d'appareils sans fil

Pour approvisionner des appareils Sidewalk en masse, vous devez créer une tâche d'importation et fournir le numéro de série de fabrication Sidewalk (SMSN) des appareils auxquels vous souhaitez intégrer AWS IoT Core pour Amazon Sidewalk. Vous avez obtenu le numéro de série de fabrication (SMSN) des appareils sous forme de fichier CSV dans votre e-mail après que le fabricant a chargé les journaux de contrôle sur Amazon Sidewalk. Pour plus d'informations sur le flux de travail et la manière d'obtenir le journal de contrôle, consultez la section [Fabrication d'appareils Amazon Sidewalk](#) dans la documentation Amazon Sidewalk.

2. Exécution du processus d'importation en arrière-plan

Lorsqu'il AWS IoT Core pour Amazon Sidewalk reçoit la demande de tâche d'importation, il commence à configurer les choses et lance un processus en arrière-plan qui interroge fréquemment le système. Une fois que le processus d'arrière-plan reçoit l'instruction de la tâche d'importation, il commence à lire le fichier CSV. AWS IoT Core pour Amazon Sidewalk vérifie simultanément si les journaux de contrôle ont été reçus d'Amazon Sidewalk.

3. Création d'enregistrements de périphériques sans fil

Lorsque le journal de contrôle est reçu d'Amazon Sidewalk, AWS IoT Core pour Amazon Sidewalk vérifie si les numéros de série du journal de contrôle correspondent aux valeurs SMSN du fichier CSV. Si les numéros de série correspondent, AWS IoT Core pour Amazon Sidewalk commencera à créer des enregistrements d'appareils sans fil pour les appareils Sidewalk qui correspondent à ces numéros de série. Une fois que tous les appareils ont été intégrés, la tâche d'importation est marquée comme terminée.

Considérations clés relatives à l'approvisionnement en vrac de Sidewalk

Lorsque vous approvisionnez vos appareils Sidewalk en masse AWS IoT Core pour Amazon Sidewalk, voici quelques considérations clés.

- Vous devez effectuer le provisionnement en bloc à l'aide de la AWS IoT console ou des opérations d'AWS IoT Core pour Amazon Sidewalk API à l'endroit Compte AWS où vous avez créé le profil de l'appareil.
- Avant de procéder au provisionnement groupé de vos appareils Sidewalk, le profil de votre appareil doit déjà contenir des informations DAK indiquant la prise en charge par l'usine. Dans le cas contraire, le provisionnement en masse à l'aide de la AWS IoT console ou les opérations de l'API de provisionnement en masse peuvent échouer.
- Une fois que vous avez lancé une tâche d'importation, le traitement du fichier CSV, l'importation des appareils sans fil et leur intégration peuvent prendre au moins 10 minutes AWS IoT Core pour Amazon Sidewalk.
- La tâche d'importation de périphériques sans fil s'exécutera pendant 90 jours, une fois démarrée. Pendant ce temps, il vérifie si les journaux de contrôle ont été reçus d'Amazon Sidewalk. Si le journal de contrôle n'est pas reçu d'Amazon Sidewalk dans les 90 jours, la tâche sera marquée comme terminée et un message indiquant qu'elle a expiré lorsque vous consulterez les détails de la tâche. L'état d'intégration des appareils concernés par la tâche d'importation qui attendaient le journal de contrôle sera marqué comme Échec.
- Lorsque vous essayez de mettre à jour une tâche d'importation que vous avez déjà créée, vous pouvez uniquement ajouter des appareils supplémentaires à la tâche. Vous pouvez ajouter de nouveaux appareils à tout moment après avoir créé une tâche d'importation et avant le début de la tâche sur

les appareils qui ont déjà été ajoutés à la tâche d'importation. Si le fichier de mise à jour contient les numéros de série des appareils qui existent déjà dans la tâche d'importation d'origine, ces numéros de série seront ignorés.

- Lorsque vous demandez une opération de mise à jour, le même rôle IAM que celui que vous avez utilisé lors de la création de la tâche d'importation est supposé accéder au fichier CSV dans le compartiment Amazon S3.
- Une tâche d'importation ne peut être supprimée que si elle s'est déjà terminée avec succès ou si la mise à jour de la tâche n'a pas pu être effectuée. La mise à jour d'une tâche peut échouer, par exemple lorsqu'un rôle IAM incorrect a été fourni ou lorsqu'aucun fichier de compartiment Amazon S3 n'a été trouvé. Une tâche d'importation ne peut pas être mise à jour ou supprimée si elle est dans l'PENDING état.
- Le fichier CSV que vous importez dans la tâche doit utiliser le format décrit dans la section suivante.

Format de fichier CSV

Le fichier CSV contenu dans un compartiment Amazon S3 que vous spécifiez pour la tâche d'importation doit utiliser le format suivant :

- La ligne 1 doit utiliser le mot clé `smsn`, qui indique que le fichier CSV importé contient le SMSN des appareils à importer.
- Les lignes 2 et suivantes doivent contenir le SMSN des appareils à intégrer. Le SMSN de l'appareil doit être au format 64 caractères hexadécimaux.

Ce fichier JSON présente un exemple de format de fichier CSV.

```
smsn
1C1A10B0AC0A200C012BBAC2CBB1B21CB12C0CA2AC1C1BB22CAA01C1B0B01122
B122C2B1121BACA2221001AC1B22012AAC11112C11C2A100C1C2B012A1100C10
02B222C110B0A210B0A0C2C112CCAC21C1C0B0AA1221AB1022A2CC11B1B1122
C2C021CA1C111CCAB1221C0021C1C2AAA0AA1A2A01ABC10CBAACCA2A0121022A
0CB22C01BBC2CA2C0B11001121ACB2ABB0BB0121C2BA101C012CC2B20C011AC0
```

Comment utiliser le provisionnement groupé de Sidewalk

Les étapes suivantes vous montrent comment utiliser le provisionnement en masse d'Amazon Sidewalk.

1. Fournissez les numéros de série de l'appareil

Pour approvisionner vos appareils Sidewalk, vous devez fournir les numéros de série des appareils à intégrer. Vous pouvez approvisionner vos appareils à l'aide de l'une des méthodes suivantes.

- Approvisionnez chaque appareil individuellement à l'aide de son numéro de série Sidewalk Manufacturing (SMSN). Cette méthode est utile lorsque vous souhaitez tester le flux de travail et intégrer votre appareil plus rapidement sans avoir à télécharger un fichier CSV avec le rôle IAM approprié ou à attendre que les appareils soient prêts à être intégrés à la tâche.
- Provisionnez des appareils en masse en fournissant une URL de compartiment Amazon S3 contenant le SMSN des appareils à provisionner dans un fichier CSV. Cette méthode est particulièrement pratique lorsque vous avez un grand nombre d'appareils à intégrer. Dans ce cas, l'intégration individuelle de chaque appareil peut s'avérer fastidieuse. Au lieu de cela, il vous suffit de fournir le chemin d'accès au fichier CSV qui a été chargé dans un compartiment Amazon S3 et le rôle IAM pour accéder au fichier.

2. Obtenir la tâche d'importation et l'état d'intégration de l'appareil

Pour chaque tâche d'importation que vous créez, vous pouvez récupérer des informations sur l'état d'intégration de la tâche et sur l'état d'intégration des appareils ajoutés à la tâche. Vous pouvez

également consulter des informations d'état supplémentaires, telles que la raison pour laquelle l'intégration d'une tâche ou d'un appareil a échoué. Pour plus d'informations, veuillez consulter la rubrique

3. (Facultatif) Mettre à jour ou supprimer une tâche d'importation

Vous pouvez mettre à jour ou supprimer une tâche d'importation que vous avez déjà créée.

- Vous pouvez mettre à jour une tâche d'importation et ajouter des appareils supplémentaires à la tâche à tout moment avant le début de la tâche sur les appareils déjà ajoutés. AWS IoT Core pour Amazon Sidewalk assume le même rôle IAM que celui que vous avez utilisé lors de la création de la tâche d'importation. Lorsque vous créez la tâche, spécifiez le nouveau fichier CSV contenant les numéros de série des appareils que vous souhaitez ajouter à la tâche.

Note

Lorsque vous mettez à jour une tâche d'importation existante, vous pouvez uniquement y ajouter des appareils. AWS IoT Core pour Amazon Sidewalk effectue une opération d'union entre les appareils qui figurent déjà dans la tâche d'importation et les appareils que vous essayez d'ajouter à la tâche. Si le nouveau fichier contient les numéros de série des appareils qui existent déjà dans la tâche d'importation, ces numéros de série seront ignorés.

- Vous pouvez supprimer une tâche d'importation qui s'est déjà terminée avec succès ou une tâche d'importation qui n'a pas pu être mise à jour, par exemple lorsque les informations relatives au rôle IAM sont incorrectes ou lorsqu'un fichier de compartiment S3 n'est pas disponible lors de la création ou de la mise à jour d'une tâche.

Rubriques

- [Fourniture de dispositifs de trottoir en vrac \(p. 1435\)](#)
- [Afficher la tâche d'importation et l'état d'intégration de l'appareil \(p. 1439\)](#)

Fourniture de dispositifs de trottoir en vrac

Cette section explique comment configurer des appareils Sidewalk en masse pour AWS IoT Core pour Amazon Sidewalk en utilisant la AWS IoT console et le AWS CLI.

Fourniture de dispositifs Sidewalk en vrac (console)

Pour ajouter votre appareil Sidewalk à l'aide de la AWS IoT console, accédez à l'[onglet Sidewalk du hub Appareils](#), choisissez Provisionner des appareils en bloc, puis effectuez les étapes suivantes.

AWS IoT Core Guide du développeur

Provisionnement d'appareils Sidewalk

à l'aide de tâches d'importation

The screenshot shows the AWS IoT Core developer guide for Sidewalk provisioning. It highlights the 'How it works' section and the 'Bulk provision' table.

How it works:

- Step 1. Add your Sidewalk device**: Create a device profile and retrieve the application server public key.
- Step 2. Provision & register your Sidewalk device**: Provision your hardware as a Sidewalk endpoint by flashing the device certificates and the application server public key.
- Step 3. Connect your Sidewalk endpoint to the cloud**: Create a destination and use AWS IoT Rules to process and route data to other AWS services.

Bulk provision (0) Info

Bulk provisioning table shows the task IDs, which includes tasks that are added for individual devices, and tasks that are linked with your S3 CSV files.

Task ID	Creation date	S3 bucket	Success count	Pending count	Failed count
No bulk provisioning tasks are currently running at this time.					

1. Choisissez la méthode d'importation

Spécifiez la manière dont vous souhaitez importer les appareils à intégrer en masse AWS IoT Core pour Amazon Sidewalk.

- Pour provisionner des appareils individuels à l'aide de leur SMSN, choisissez Provisionner un appareil individuel pris en charge par l'usine.
- Pour approvisionner des appareils en masse en fournissant un fichier CSV contenant la liste des appareils et leurs SMS, choisissez Utiliser le compartiment S3.

2. Spécifier les appareils à intégrer

Selon la méthode que vous avez choisie pour intégrer vos appareils, ajoutez les informations sur les appareils et leurs numéros de série.

- Si vous avez choisi Provisionner un appareil individuel pris en charge par l'usine, spécifiez les informations suivantes :
 - Un nom pour chaque appareil à intégrer. Le nom doit être unique dans votre Compte AWS Région AWS.
 - Leur numéro de série de fabrication du trottoir (SMSN) dans le champ Entrez SMSN.
 - Destination qui décrit la règle de l'IoT pour acheminer les messages d'un appareil à un autre Service AWS.
- Si vous avez choisi Utiliser le compartiment S3 :
 - Fournissez les informations de destination du compartiment S3, qui comprennent les informations d'URL S3. Pour fournir votre fichier CSV, choisissez Browse S3, puis choisissez le fichier CSV que vous souhaitez utiliser.

AWS IoT Core pour Amazon Sidewalk renseigne automatiquement l'URL S3, qui est le chemin d'accès à votre fichier CSV dans le compartiment S3. Le format du chemin

est `s3://bucket_name/file_name`. Pour afficher le fichier dans la console [Amazon Simple Storage Service](#), choisissez Afficher.

- ii. Fournissez le rôle de provisionnement S3, qui AWS IoT Core pour Amazon Sidewalk permet d'accéder au fichier CSV du compartiment S3 en votre nom. Vous pouvez créer un rôle de service ou choisir un rôle existant.

Pour créer un nouveau rôle, vous pouvez fournir un nom de rôle ou le laisser vide pour générer automatiquement un nom aléatoire.

- iii. Fournissez une destination qui décrit la règle IoT pour acheminer les messages de l'appareil vers d'autres services AWS.

3. Démarrer la tâche d'importation

Fournissez toutes les balises facultatives sous forme de paires nom-valeur et choisissez Soumettre pour démarrer la tâche d'importation de votre appareil sans fil.

Fourniture de dispositifs de trottoir en vrac (CLI)

Pour intégrer vos appareils Sidewalk à votre compte AWS IoT Core pour Amazon Sidewalk, utilisez l'une des opérations d'API suivantes selon que vous souhaitez ajouter des appareils individuellement ou en fournissant le fichier CSV contenu dans un compartiment S3.

- Importez des appareils en masse à l'aide d'un fichier CSV S3

Pour télécharger des appareils en masse en fournissant le fichier CSV dans un compartiment S3, utilisez l'opération d'[StartWirelessDeviceImportTask](#) API ou la [start-wireless-device-import-task](#) AWS CLI commande. Lors de la création de la tâche, spécifiez le chemin d'accès au fichier CSV dans le compartiment Amazon S3 et le rôle IAM qui accorde AWS IoT Core pour Amazon Sidewalk les autorisations d'accès au fichier CSV.

Une fois que la tâche commence à s'exécuter, je AWS IoT Core pour Amazon Sidewalk vais commencer à lire le fichier CSV et comparer les numéros de série (SMSN) du fichier avec les informations correspondantes dans le journal de contrôle reçu d'Amazon Sidewalk. Lorsque les numéros de série correspondent, il commence à créer des enregistrements de périphériques sans fil correspondant à ces numéros de série.

La commande suivante présente un exemple de création d'une tâche d'importation :

```
aws iotwireless start-wireless-device-import-task \
    --cli-input-json "file://task.json"
```

L'exemple suivant affiche le contenu du fichier task.json.

Contenu de task.json

```
{
  "DestinationName": "Sidewalk_Destination",
  "Sidewalk": {
    "DeviceCreationFile": "s3://import_task_bucket/import_file1",
    "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI"
  }
}
```

L'exécution de cette commande renvoie un ID et un ARN pour la tâche d'importation.

```
{
```

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"  
    "Id": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a"  
}
```

- Approvisionnez les appareils individuellement à l'aide de leur SMSN

Pour provisionner des appareils individuellement à l'aide de leur SMSN, utilisez l'opération [StartSingleWirelessDeviceImportTask](#) API ou la [start-single-wireless-device-import-task](#) AWS CLI commande. Lors de la création de la tâche, spécifiez la destination Sidewalk et le numéro de série de l'appareil que vous souhaitez intégrer.

Lorsque le numéro de série correspond aux informations correspondantes dans le journal de contrôle reçu d'Amazon Sidewalk, la tâche s'exécute et crée l'enregistrement du périphérique sans fil.

La commande suivante présente un exemple de création d'une tâche d'importation :

```
aws iotwireless start-single-wireless-device-import-task \  
    --destination-name sidewalk_destination \  
    --sidewalk  
    '{"SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A4A"}'
```

L'exécution de cette commande renvoie un ID et un ARN pour la tâche d'importation.

```
{  
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:ImportTask/e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"  
    "Id": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"  
}
```

Mettre à jour ou supprimer des tâches d'importation

Si vous souhaitez ajouter des appareils supplémentaires à une tâche d'importation, vous pouvez mettre à jour la tâche. Vous pouvez également supprimer une tâche si vous n'en avez plus besoin ou si elle a échoué. Pour savoir quand mettre à jour ou supprimer une tâche, consultez [Comment utiliser le provisionnement groupé de Sidewalk \(p. 1434\)](#).

Warning

Les actions de suppression sont permanentes et ne peuvent pas être annulées. La suppression d'une tâche d'importation déjà terminée avec succès ne supprimera pas les terminaux qui ont déjà été intégrés à l'aide de cette tâche.

Pour mettre à jour ou supprimer des tâches d'importation :

- Utilisation de la console AWS IoT

Les étapes suivantes vous montrent comment mettre à jour ou supprimer vos tâches d'importation à l'aide de la AWS IoT console.

Pour mettre à jour une tâche d'importation :

1. Accédez au [hub des appareils Sidewalk](#) de la AWS IoT console.
2. Choisissez la tâche d'importation que vous voulez mettre à jour, puis choisissez Modifier.
3. Fournissez un autre fichier S3 contenant les numéros de série des appareils que vous souhaitez ajouter à la tâche, puis choisissez Soumettre.

Pour supprimer une tâche d'importation :

1. Accédez au [hub des appareils Sidewalk](#) de la AWS IoT console.
 2. Choisissez la tâche que vous voulez supprimer, puis choisissez Supprimer.
- À l'aide de l'APIAWS IoT Wireless ouAWS CLI

Utilisez les opérations d'APIAWS IoT sans fil ou les commandes CLI suivantes pour mettre à jour ou supprimer votre tâche d'importation.

- [UpdateWirelessDeviceImportTask](#)API ou [update-wireless-device-import-task](#)CLI

Cette opération d'API ajoute le contenu d'un fichier CSV Amazon S3 à une tâche d'importation existante. Vous ne pouvez ajouter que les numéros de série des appareils qui n'étaient pas inclus auparavant dans la tâche. Pour plus d'informations, veuillez consulter [Ajouter des appareils à la tâche d'importation \(p. 1447\)](#).

- [DeleteWirelessDeviceImportTask](#)API ou [delete-wireless-device-import-task](#)CLI

Cette opération d'API supprime la tâche d'importation qui a été marquée pour suppression à l'aide de l'ID de tâche d'importation. Pour plus d'informations, veuillez consulter [Supprimez les tâches d'importation de votreCompte AWS \(p. 1448\)](#).

Afficher la tâche d'importation et l'état d'intégration de l'appareil

Les tâches d'importation de votre appareil sans fil et les appareils Sidewalk que vous avez ajoutés à la tâche peuvent avoir l'un des messages d'état suivants. Ces messages s'affichent dans la AWS IoT console ou lorsque vous utilisez l'une des opérations ou des AWS CLI commandes de l'APIAWS IoT Wireless pour récupérer des informations sur ces tâches et leurs appareils.

Afficher les informations sur l'état d'une tâche d'importation

Après avoir créé une tâche d'importation, vous pouvez consulter la tâche d'importation que vous avez créée et l'état d'intégration des appareils ajoutés à la tâche. L'état d'intégration indique le nombre d'appareils en attente d'intégration, le nombre d'appareils qui ont été intégrés avec succès et le nombre d'appareils qui n'ont pas pu être intégrés.

Lorsqu'une tâche d'importation vient d'être créée, le nombre d'appareils en attente affiche une valeur correspondant au nombre d'appareils ajoutés. Une fois que la tâche démarre et lit le fichier CSV pour créer les enregistrements des appareils sans fil, le nombre d'appareils en attente diminue et le nombre de réussites augmente à mesure que les appareils sont correctement intégrés. Si l'un des appareils ne parvient pas à s'intégrer, le nombre d'échecs augmentera.

Pour consulter la tâche d'importation et l'état d'intégration de l'appareil, procédez comme suit :

- Utilisation de la console AWS IoT

Dans le [hub des appareils Sidewalk](#) de la AWS IoT console, vous pouvez consulter les tâches d'importation que vous avez créées et le décompte du résumé des informations d'état d'intégration de vos appareils. Si vous consultez les détails de l'une des tâches d'importation que vous avez créées, vous pouvez consulter des informations supplémentaires sur l'état d'intégration de l'appareil.

- À l'aide de l'APIAWS IoT Wireless ouAWS CLI

Pour consulter l'état d'intégration de l'appareil, utilisez l'une des opérations de l'APIAWS IoT Wireless suivantes ou la AWS CLI commande correspondante. Pour de plus amples informations et des exemples qui vous montrent comment les utiliser, consultez [AWS IoT Core pour Amazon SidewalkOpérations d'API pour le provisionnement en masse \(p. 1446\)](#).

- [ListWirelessDeviceImportTasks](#)API ou [list-wireless-device-import-tasks](#)CLI

Cette opération d'API renvoie des informations sur toutes les tâches d'importation qui ont été ajoutées à votre compte pour AWS IoT Wireless et leur statut. Il renvoie également le décompte du résumé de l'état d'intégration des appareils Sidewalk dans le cadre de ces tâches.

- [ListDevicesForWirelessDeviceImportTask](#) API ou [list-devices-for-wireless-device-import-task](#) CLI

Cette opération d'API renvoie des informations sur la tâche d'importation spécifiée et son état, ainsi que des informations sur tous les appareils Sidewalk qui ont été ajoutés à la tâche d'importation et leurs informations d'état d'intégration.

- [GetWirelessDeviceImportTask](#) API ou [get-wireless-device-import-task](#) CLI

Cette opération d'API renvoie des informations sur la tâche d'importation spécifiée et son état, ainsi qu'un décompte du résumé de l'état d'intégration des appareils Sidewalk dans cette tâche.

Importer le statut d'une une

Les tâches d'importation que vous avez créées dans votre Compte AWS peuvent avoir l'un des messages d'état suivants. Le statut indique si le traitement de votre tâche d'importation a commencé, s'est terminée ou a échoué. Vous pouvez également utiliser la AWS IoT console ou le `StatusReason` paramètre de n'importe quelle opération de l'API AWS IoT Wireless pour récupérer des informations d'état supplémentaires.

- INITIALISATION

AWS IoT Core pour Amazon Sidewalk a reçu la demande de tâche d'importation de périphériques sans fil et est en train de configurer la tâche.

- INITIALISÉ

AWS IoT Core pour Amazon Sidewalk a terminé la configuration de la tâche d'importation et attend l'arrivée du journal de contrôle pour pouvoir importer les appareils à l'aide de leurs numéros de série (SMSN) et poursuivre le traitement de la tâche.

- PENDING

La tâche d'importation attend d'être traitée dans la file d'attente. AWS IoT Core pour Amazon Sidewalk va évaluer les autres tâches qui se trouvent dans la file d'attente de traitement.

- ACHEVÉE

La tâche d'importation a été traitée et terminée.

- ÉCHEC

La tâche d'importation ou la tâche de l'appareil a échoué. Vous pouvez utiliser le `StatusReason` paramètre pour identifier la raison pour laquelle la tâche d'importation a échoué, par exemple en cas d'exception de validation.

- SUPPRESSION

La tâche d'importation a été marquée pour suppression et est en cours de suppression.

État d'intégration de l'appareil

Les appareils Sidewalk que vous avez ajoutés à votre tâche d'importation peuvent avoir l'un des messages d'état suivants. L'état indique si vos appareils sont prêts à être intégrés, s'ils ont été intégrés ou n'ont pas pu être intégrés. Vous pouvez également utiliser la AWS IoT console ou le `OnboardingStatusReason` paramètre du fonctionnement de l'API AWS IoT Wireless pour récupérer des informations d'état supplémentaires. `ListDevicesForWirelessDeviceImportTask`

- INITIALISÉ

AWS IoT Core pour Amazon Sidewalk a terminé la configuration de la tâche d'importation et attend l'arrivée du journal de contrôle pour pouvoir importer les appareils à l'aide de leurs numéros de série (SMSN) et poursuivre le traitement de la tâche.

- PENDING

La tâche d'importation est en attente dans la file d'attente pour être traitée et pour commencer à intégrer vos appareils à la tâche. AWS IoT Core pour Amazon Sidewalk évalue les autres tâches qui se trouvent dans la file d'attente de traitement.

- EMBARQUÉ

L'appareil Sidewalk a été intégré avec succès à la tâche d'importation.

- ÉCHEC

La tâche d'importation ou la tâche de l'appareil a échoué et l'appareil Sidewalk n'a pas pu être intégré à la tâche. Vous pouvez utiliser le `OnboardingStatusReason` paramètre pour récupérer des informations supplémentaires sur les raisons de l'échec de l'intégration de l'appareil.

AWS IoT Core pour Amazon SidewalkOpérations d'API

Vous pouvez effectuer les opérations d'API supplémentaires suivantes lors de l'intégration de vos terminaux Sidewalk ou lors de la création d'une tâche d'importation pour approvisionner des terminaux Sidewalk en masse.

Les sections suivantes contiennent des informations supplémentaires sur ces opérations d'API.

Rubriques

- [AWS IoT Core pour Amazon SidewalkOpérations d'API pour les profils d'appareils \(p. 1441\)](#)
- [AWS IoT Core pour Amazon SidewalkOpérations d'API pour les terminaux Sidewalk \(p. 1442\)](#)
- [AWS IoT Core pour Amazon SidewalkOpérations d'API pour les destinations des terminaux Sidewalk \(p. 1444\)](#)
- [AWS IoT Core pour Amazon SidewalkOpérations d'API pour le provisionnement en masse \(p. 1446\)](#)

AWS IoT Core pour Amazon SidewalkOpérations d'API pour les profils d'appareils

Vous pouvez effectuer les opérations d'API suivantes pour les profils d'appareils Sidewalk :

- [CreateDeviceProfile API](#) ou [create-device-profile CLI](#)
- [GetDeviceProfile API](#) ou [get-device-profile CLI](#)
- [ListDeviceProfiles API](#) ou [list-device-profiles CLI](#)
- [DeleteDeviceProfile API](#) ou [delete-device-profile CLI](#)

Les sections suivantes vous expliquent comment répertorier et supprimer des profils. Pour plus d'informations sur la création et la récupération de profils d'appareils, consultez [Étape 1 : Création d'un profil d'appareil \(p. 1414\)](#).

Répertoriez les profils d'appareils dans votreCompte AWS

Vous pouvez utiliser l'opération [ListDeviceProfiles](#)d'API pour répertorier les profils d'appareilsCompte AWS auxquels vous avez ajouté des donnéesAWS IoT Core pour Amazon Sidewalk. Ces informations peuvent vous permettre d'identifier les appareils auxquels vous souhaitez associer ce profil.

Pour filtrer la liste afin d'afficher uniquement les profils des appareils Sidewalk, définissez cette optionType surSidewalk lors de l'exécution de l'API. Voici un exemple de commande de l'interface de ligne de commande :

```
aws iotwireless list-device-profiles --wireless-device-type "Sidewalk"
```

L'exécution de cette commande renvoie la liste des profils d'appareils que vous avez ajoutés, y compris leur identifiant de profil et Amazon Resource Name (ARN). Pour récupérer des informations supplémentaires sur un profil spécifique, utilisez l'[GetDeviceProfile](#)API.

```
{  
    "DeviceProfileList": [  
        {  
            "Name": "SidewalkDeviceProfile1",  
            "Id": "12345678-a1b2-3c45-67d8-e90fa1b2c34d",  
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/12345678-  
a1b2-3c45-67d8-e90fa1b2c34d"  
        },  
        {  
            "Name": "SidewalkDeviceProfile2",  
            "Id": "a1b2c3d4-5678-90ab-cdef-12ab345c67de",  
            "Arn": "arn:aws:iotwireless:us-east-1:123456789012:DeviceProfile/  
a1b2c3d4-5678-90ab-cdef-12ab345c67de"  
        }  
    ]  
}
```

Supprimer les profils d'appareils de votreCompte AWS

Vous pouvez supprimer les profils de vos appareils à l'aide de l'opération [DeleteDeviceProfile](#)API. Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

Warning

Les actions de suppression ne peuvent pas être annulées. Le profil de l'appareil sera définitivement supprimé de votreCompte AWS.

```
aws iotwireless delete-device-profile --name "SidewalkProfile"
```

Cette commande ne produit aucune sortie. Vous pouvez utiliser l'[GetDeviceProfile](#)API ou l'opération[ListDeviceProfiles](#) API pour vérifier que le profil a été supprimé de votre compte.

AWS IoT Core pour Amazon SidewalkOpérations d'API pour les terminaux Sidewalk

Vous pouvez effectuer les opérations d'API suivantes pour vos appareils Sidewalk :

- [CreateWirelessDevice](#)API ou [create-wireless-device](#)CLI
- [GetWirelessDevice](#)API ou [get-wireless-device](#)CLI

- [ListWirelessDevices](#) API ou [list-wireless-devices](#) CLI
- [DeleteWirelessDevice](#) API ou [delete-wireless-device](#) CLI
- [UpdateWirelessDevice](#) API ou [update-wireless-device](#) CLI
- [AssociateWirelessDeviceWithThing](#) API ou [associate-wireless-device-with-thing](#) CLI
- [DisassociateWirelessDeviceFromThing](#) API ou [disassociate-wireless-device-from-thing](#) CLI

Les sections suivantes vous expliquent comment répertorier et supprimer des appareils. Pour plus d'informations sur la création de terminaux Sidewalk et la récupération des informations sur les appareils, voir [Étape 2 : Ajouter votre appareil Sidewalk \(p. 1415\)](#)

Associez les terminaux Sidewalk de votre Compte AWS entreprise à un objet IoT

Pour associer votre appareil Sidewalk à un AWS IoT objet, utilisez l'opération [AssociateWirelessDeviceWithThing](#) API.

Les éléments AWS IoT qui s'y trouvent facilitent la recherche et la gestion de vos appareils. L'association d'un objet à votre appareil permet à celui-ci d'accéder à d'autres AWS IoT Core fonctionnalités. Pour plus d'informations sur l'utilisation de cette API, consultez la documentation de référence d'API.

Ce qui suit présente un exemple d'exécution de cette commande. L'exécution de cette commande ne produit aucune sortie.

```
aws iotwireless associate-wireless-device-with-thing \
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d" \
--thing-arn "arn:aws:iot:us-east-1:123456789012:thing/MySidewalkThing"
```

Pour dissocier votre appareil Sidewalk d'un AWS IoT objet, utilisez l'opération [DisassociateWirelessDeviceFromThing](#) API, comme indiqué dans l'exemple suivant.

```
aws iotwireless disassociate-wireless-device-from-thing \
--id "12345678-a1b2-3c45-67d8-e90fa1b2c34d"
```

Répertoriez les terminaux Sidewalk dans votre Compte AWS

Pour répertorier les appareils Sidewalk dans votre Compte AWS auxquels vous avez ajouté des AWS IoT Core pour Amazon Sidewalk éléments, utilisez l'opération [ListWirelessDevices](#) API. Pour filtrer la liste afin de ne renvoyer que les appareils Sidewalk, réglez `WirelessDeviceType` le sur `Sidewalk`.

Voici un exemple d'exécution de cette commande :

```
aws iotwireless list-wireless-devices --wireless-device-type Sidewalk
```

L'exécution de cette commande renvoie la liste des appareils que vous avez ajoutés, y compris leur identifiant de profil et l'Amazon Resource Name (ARN). Pour récupérer des informations supplémentaires sur un appareil spécifique, utilisez l'opération [GetWirelessDevice](#) API.

```
{
  "WirelessDeviceList": [
    {
      "Name": "mySidewalkDevice",
      "DestinationName": "SidewalkDestination",
      "Id": "1ffd32c8-8130-4194-96df-622f072a315f",
```

```
"Type": "Sidewalk",
"Sidewalk": [
    "SidewalkId": "1234567890123456",
    "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:WirelessDevice/1ffd32c8-8130-4194-96df-622f072a315f"
]
}
```

Supprimez les appareils Sidewalk de votreCompte AWS

Pour supprimer vos appareils Sidewalk, passezWirelessDeviceID les appareils que vous souhaitez supprimer à l'opération [DeleteWirelessDevice](#)API.

Ce qui suit présente un exemple de commande :

```
aws iotwireless delete-wireless-device --id "23456789-abcd-0123-bcde-fabc012345678"
```

Cette commande ne produit aucune sortie. Vous pouvez utiliser l'GetWirelessDeviceAPI ou l'opérationListWirelessDevices API pour vérifier que l'appareil a été supprimé de votre compte.

AWS IoT Core pour Amazon SidewalkOpérations d'API pour les destinations des terminaux Sidewalk

Vous pouvez effectuer les opérations d'API suivantes pour les destinations de vos terminaux Sidewalk :

- [CreateDestination](#)API ou [create-destination](#)CLI
- [GetDestination](#)API ou [get-destination](#)CLI
- [UpdateDestination](#)API ou [update-destination](#)CLI
- [ListDestinations](#)API ou [list-destinations](#)CLI
- [DeleteDestination](#)API ou [delete-destination](#)CLI

Les sections suivantes vous expliquent comment obtenir, répertorier, mettre à jour et supprimer des destinations. Pour de plus amples informations sur la création de destinations, consultez[Ajoutez une destination pour votre terminal Sidewalk \(p. 1419\)](#).

Obtenez des informations sur votre destination

Vous pouvez utiliser l'opération [GetDestination](#)API pour obtenir des informations sur la destination pour laquelle vous avez ajouté à votre compteAWS IoT Core pour Amazon Sidewalk. Fournissez le nom de destination en entrée à l'API. L'API renverra des informations sur la destination correspondant à l'identifiant spécifié.

Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

```
aws iotwireless get-destination --name SidewalkDestination
```

L'exécution de cette commande renvoie les paramètres de votre destination.

```
{
    "Arn": "arn:aws:iotwireless:us-east-1:123456789012:Destination/IoTWirelessDestination",
    "Name": "SidewalkDestination",
    "Expression": "IoTWirelessRule",
```

```
    "ExpressionType": "RuleName",
    "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
}
```

Mettre à jour les propriétés de votre destination

Utilisez l'opération [UpdateDestination](#)d'API pour mettre à jour les propriétés de la destination pour laquelle vous avez ajouté à votre compte AWS IoT Core pour Amazon Sidewalk. Ce qui suit présente un exemple de commande de l'interface de ligne de commande qui met à jour la propriété de description :

```
aws iotwireless update-destination --name SidewalkDestination \
--description "Destination for messages processed using IoTWirelessRule"
```

Répertoriez les destinations dans votreCompte AWS

Utilisez l'opération [ListDestinations](#)d'API pour répertorier les destinationsCompte AWS auxquelles vous avez ajouté des informationsAWS IoT Core pour Amazon Sidewalk. Pour filtrer la liste afin de ne renvoyer que les destinations des terminaux Sidewalk, utilisez leWirelessDeviceType paramètre.

Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

```
aws iotwireless list-destinations --wireless-device-type "Sidewalk"
```

L'exécution de cette commande renvoie la liste des destinations que vous avez ajoutées, y compris leur Amazon Resource Name (ARN) pour. Pour obtenir des informations supplémentaires sur une destination spécifique, utilisez l'GetDestinationAPI.

```
{
  "DestinationList": [
    {
      "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/IoTWirelessDestination",
      "Name": "IoTWirelessDestination",
      "Expression": "IoTWirelessRule",
      "Description": "Destination for messages processed using IoTWirelessRule",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    },
    {
      "Arn": "arn:aws:iotwireless:us-
east-1:123456789012:Destination/IoTWirelessDestination2",
      "Name": "IoTWirelessDestination2",
      "Expression": "IoTWirelessRule2",
      "RoleArn": "arn:aws:iam::123456789012:role/IoTWirelessDestinationRole"
    }
  ]
}
```

Supprimer des destinations de votreCompte AWS

Pour supprimer votre destination, transmettez le nom de la destination à supprimer comme entrée à l'opération [DeleteDestination](#)API. Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

Warning

Les actions de suppression ne peuvent pas être annulées. La destination sera définitivement supprimée de votreCompte AWS.

```
aws iotwireless delete-destination --name "SidewalkDestination"
```

Cette commande ne produit aucune sortie. Vous pouvez utiliser l'GetDestinationAPI ou l'opérationListDestinations API pour vérifier que la destination a été supprimée de votre compte.

AWS IoT Core pour Amazon SidewalkOpérations d'API pour le provisionnement en masse

Vous pouvez effectuer les opérations d'API suivantes pour le provisionnement en masse de vos terminaux Sidewalk :

- [StartWirelessDeviceImportTask](#)API ou [start-wireless-device-import-task](#)CLI
- [StartSingleWirelessDeviceImportTask](#)API ou [start-single-wireless-device-import-task](#)CLI
- [ListWirelessDeviceImportTasks](#)API ou [list-wireless-device-import-tasks](#)CLI
- [ListDevicesForWirelessDeviceImportTask](#)API ou [list-devices-for-wireless-device-import-task](#)CLI
- [GetWirelessDeviceImportTask](#)API ou [get-wireless-device-import-task](#)CLI
- [UpdateWirelessDeviceImportTask](#)API ou [update-wireless-device-import-task](#)CLI
- [DeleteWirelessDeviceImportTask](#)API ou [delete-wireless-device-import-task](#)CLI

Les sections suivantes vous expliquent comment obtenir, répertorier, mettre à jour et supprimer des tâches d'importation. Pour de plus amples informations sur la création de tâches d'importation, consultez[AWS IoT Core pour Amazon SidewalkOpérations d'API pour le provisionnement en masse \(p. 1446\)](#).

Obtenez des informations sur votre tâche d'importation

Vous pouvez utiliser l'opération [ListDevicesForWirelessDeviceImportTask](#)d'API pour récupérer des informations sur une tâche d'importation particulière et sur l'état d'intégration des appareils concernés par cette tâche. En entrée de l'opération d'API, spécifiez l'ID de tâche d'importation que vous avez obtenu à partir des opérationsStartWirelessDeviceImportTask ou deStartSingleWirelessDeviceImportTask l'API. L'API renverra ensuite des informations sur la tâche d'importation correspondant à l'identifiant spécifié.

Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

```
aws iotwireless list-devices-for-wireless-device-import-task --id e2a5995e-743b-41f2-a1e4-3ca6a5c5249f
```

L'exécution de cette commande renvoie les informations relatives à la tâche d'importation et l'état d'intégration de l'appareil.

```
{
    "DestinationName": "SidewalkDestination",
    "ImportedWirelessDeviceList": [
        {
            "Sidewalk": {
                "OnboardingStatus": "ONBOARDED",
                "LastUpdateTime": "2023-02021T06:11:09.151Z",
                "SidewalkManufacturingSn": "82B83C8B35E856F43CE9C3D59B418CC96B996071016DB1C3BE5901F0F3071A4A"
            }
        }
    ]
}
```

```
"Sidewalk": {  
    "OnboardingStatus": "PENDING",  
    "LastUpdateTime": "2023-02-06T06:22:12.061Z",  
    "SidewalkManufacturingSn":  
        "12345ABCDE6789FABDESBDEF123456789012345FEABC0123679AFEBC01234EF"  
},  
}  
}
```

Obtenir le résumé des tâches d'importation de l'appareil

Pour obtenir le nombre d'informations récapitulatives sur l'état d'intégration des appareils que vous avez ajoutés à une tâche d'importation particulière, utilisez l'opération [GetWirelessDeviceImportTask](#) API. Ce qui suit présente un exemple de commande de l'interface de ligne de commande.

```
aws iotwireless get-wireless-device-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

Le code suivant présente un exemple de réponse de la commande.

```
{  
    "NumberOfFailedImportedDevices": 2,  
    "NumberOfOnboardedImportedDevices": 4,  
    "NumberOfPendingImportedDevices": 1  
}
```

Ajouter des appareils à la tâche d'importation

Utilisez l'opération [UpdateWirelessDeviceImportTask](#) d'API pour ajouter des appareils à une tâche d'importation existante que vous avez ajoutée. Vous pouvez utiliser cette opération d'API pour ajouter les numéros de série (SMSN) des appareils qui n'étaient pas inclus auparavant dans la tâche que vous avez créée à l'aide de l'opération [StartWirelessDeviceImportTask](#) API.

Pour ajouter des appareils à la tâche d'importation, dans le cadre de la demande d'API, spécifiez un nouveau fichier CSV dans un compartiment Amazon S3 contenant les numéros de série des appareils à ajouter. La demande ne sera acceptée que si le processus d'intégration n'a pas encore commencé pour les appareils qui font actuellement l'objet de la tâche d'importation. Si le processus d'intégration a déjà commencé, la demande [UpdateWirelessDeviceImportTask](#) d'API échouera.

Si vous souhaitez toujours ajouter des appareils à la tâche d'importation, vous pouvez effectuer l'opération [UpdateWirelessDeviceImportTask](#) API une deuxième fois. Avant d'effectuer cette opération d'API, la première demande [UpdateWirelessDeviceImportTask](#) API doit avoir terminé le traitement du fichier CSV dans le compartiment S3.

Note

Lorsque vous effectuez une demande d'[ListImportedWirelessDeviceTasks](#) API, l'URL S3 du nouveau fichier CSV spécifié à l'aide de l'opération [d'UpdateWirelessDeviceImportTask](#) API n'est actuellement pas renvoyée. Au lieu de cela, l'opération d'API renvoie l'URL S3 de la demande envoyée initialement à l'aide de la demande [StartWirelessDeviceImportTask](#) d'API.

Ce qui suit présente un exemple de commande de l'interface de ligne de commande.

```
aws iotwireless update-wireless-device-import task \  
    --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f" \  
    --CSV-File "s3://my-bucket/devices.csv"
```

```
--sidewalk '{"FileForCreateDevices": "s3://import_task_bucket/import_file3"}'
```

Répertoriez les tâches d'importation dans votreCompte AWS

Utilisez l'`ListWirelessDeviceImportTasks` API ou la commande `list-imported-wireless-device-tasks` CLI pour répertorier les tâches d'importation dans votreCompte AWS. Ce qui suit présente un exemple de commande de l'interface de ligne de commande.

```
aws iotwireless list-wireless-device-import-tasks
```

L'exécution de cette commande renvoie une liste des tâches d'importation que vous avez créées. La liste inclut leurs fichiers CSV Amazon S3 et le rôle IAM spécifié, l'ID de la tâche d'importation et des informations récapitulatives sur l'état d'intégration de l'appareil.

```
{
  "ImportWirelessDeviceTaskList": [
    {
      "FileForCreateDevices": "s3://import_task_bucket/import_file1",
      "ImportTaskId": "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f",
      "NumberOfFailedImportedDevices": 1,
      "NumberOfOnboardedImportedDevices": 3,
      "NumberOfPendingImportedDevices": 2,
      "Role": "arn:aws:iam::123456789012:role/service-role/ACF1zBEI",
      "TimeStamp": "1012202218:23:55"
    },
    {
      "FileForCreateDevices": "s3://import_task_bucket/import_file2",
      "ImportTaskId": "a1b234c5-67ef-21a2-a1b2-3cd4e5f6789a",
      "NumberOfFailedImportedDevices": 2,
      "NumberOfOnboardedImportedDevices": 4,
      "NumberOfPendingImportedDevices": 1,
      "Role": "arn:aws:iam::123456789012:role/service-role/CDEFaBC1",
      "TimeStamp": "1201202210:12:20"
    }
  ]
}
```

Supprimez les tâches d'importation de votreCompte AWS

Pour supprimer une tâche d'importation, transmettez l'ID de la tâche d'importation à l'opération `d>DeleteWirelessDeviceImportTask` API ou à la commande `delete-wireless-device-import-task` CLI.

Warning

Les actions de suppression ne peuvent pas être annulées. La tâche d'importation sera définitivement supprimée de votreCompte AWS.

Lorsque vous exécutez la demande `DeleteWirelessDeviceImportTask` d'API, un processus en arrière-plan commence à supprimer la tâche d'importation. Lorsque la demande est en cours, les numéros de série (SMSN) des appareils concernés par les tâches d'importation sont en cours de suppression. Ce n'est qu'une fois la suppression terminée que vous pourrez voir ces informations à l'aide des opérations de `GetImportedWirelessDeviceTasks` API `ListImportedWirelessDeviceTasks` ou.

Si une tâche d'importation contient toujours des appareils qui attendent d'être intégrés, la demande `DeleteWirelessDeviceImportTask` API ne sera traitée que lorsque tous les appareils concernés par la tâche d'importation auront été intégrés ou n'auront pas pu être intégrés. Une tâche d'importation expire au bout de 90 jours et, une fois expirée, elle peut être supprimée de votre compte. Toutefois, les appareils qui ont été intégrés avec succès à l'aide de la tâche d'importation ne seront pas supprimés.

Note

Si vous tentez de créer une autre tâche d'importation incluant le numéro de série d'un appareil en attente de suppression à l'aide de la demande `DeleteWirelessDeviceImportTaskAPI`, l'opération `StartWirelessDeviceImportTaskAPI` renverra une erreur.

Ce qui suit présente un exemple de commande de l'interface de ligne de commande :

```
aws iotwireless delete-import-task --Id "e2a5995e-743b-41f2-a1e4-3ca6a5c5249f"
```

Cette commande ne produit aucune sortie. Une fois la tâche supprimée, pour vérifier que la tâche d'importation a été supprimée de votre compte, vous pouvez utiliser l'opération `GetWirelessDeviceImportTask API` ou l'opération `ListWirelessDeviceImportTasks API`.

Surveillance et événements pour AWS IoT Core pour Amazon Sidewalk

Vous pouvez configurer des événements et utiliser CloudWatch les journaux pour surveiller vos appareils Sidewalk et résoudre tout problème.

Événements pour les appareils Sidewalk

AWS IoT Core pour Amazon Sidewalk Utilisez-le pour publier des messages vous informant d'événements tels que la mise en service ou l'enregistrement des appareils Sidewalk de votre compte. Les événements sont publiés via MQTT avec une charge utile JSON dont le contenu dépend du type d'événement.

Types d'événements pour les appareils Sidewalk

Les ressources sur les événements pour Sidewalk incluent :

- Événements relatifs à l'état d'enregistrement des appareils

Ces événements publient des notifications en cas de modification de l'état d'enregistrement de l'appareil, par exemple lorsqu'un appareil Sidewalk a été approvisionné ou enregistré.

- Événements de proximité

Ces événements publient des notifications lorsqu'une balise est reçue de votre appareil Sidewalk. Des balises sont envoyées à intervalles réguliers lorsque votre appareil Sidewalk s'approche d'Amazon Sidewalk.

- Statut de distribution des messages

Ces événements publient des notifications concernant l'état des messages échangés entre AWS IoT Core pour Amazon Sidewalk et l'appareil Sidewalk. Par exemple, il publiera un événement pour indiquer quand un message n'a pas pu être livré même si l'`SendDataToWirelessDeviceAPI` renvoie un identifiant de message.

Activer les événements pour les appareils Sidewalk

Pour recevoir des événements, votre appareil doit appliquer une politique appropriée lui permettant de se connecter à la passerelle de l'AWS IoT appareil et de s'abonner aux rubriques d'événements MQTT, comme indiqué dans l'exemple suivant.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe",  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:region:account:$aws/iotwireless/events/join/*",  
                "arn:aws:iotwireless:region:account:$aws/iotwireless/events/connection_status/  
*",  
                "arn:aws:iotwireless:region:account:$aws/iotwireless/events/  
device_registration_state/*",  
                "arn:aws:iotwireless:region:account:$aws/iotwireless/events/proximity/*",  
                "arn:aws:iotwireless:region:account:$aws/iotwireless/events/  
message_delivery_status/*"  
            ]  
        }]  
    }]
```

Vous pouvez activer les événements à l'aide de AWS CLI l'AWS IoT Core pour Amazon SidewalkAPI ou du AWS Management Console. Pour configurer des événements pour tous les appareils Sidewalk, utilisez l'UpdateEventConfigurationByResourceTypesAPI ou la commandeupdate-event-configuration-by-resource-types CLI.

```
aws iotwireless update-event-configuration-by-resource-types \  
    --device-registration-state Sidewalk={WirelessDeviceEventTopic="Enabled"}
```

Pour configurer des événements pour des appareils individuels, utilisez l'UpdateResourceEventConfigurationAPI ou la commandeupdate-resource-event-configuration CLI.

```
aws iotwireless update-resource-event-configuration --identifier-type WirelessDeviceID \  
    --identifier "1ffd32c8-8130-4194-96df-622f072a315f" \  
    --message-delivery-status Sidewalk={WirelessDeviceIdEventTopic="Enabled"}
```

Pour plus d'informations sur la configuration des événements, consultez Kits et événements [Sidewalk](#), [consultez Kits](#) Sidewalk

Surveillance des dispositifs de trottoir

Vous pouvez surveiller vos appareils et applications Sidewalk qui s'exécutent en temps réel à l'aide d'Amazon CloudWatch. CloudWatch Utilisez-le pour collecter et suivre des statistiques, consulter les journaux contenant des informations sur l'état de vos appareils Sidewalk, filtrer les journaux pour afficher uniquement les erreurs ou obtenir des informations utiles.

Lorsque vous utilisez Amazon CloudWatch, configurez d'abord un rôle de journalisation, puis utilisez les métriques de journal ou CloudWatch Insights pour surveiller vos appareils. Toute erreur qui se produit sera enregistrée et envoyée à votre groupe de CloudWatch journaux. Vous pouvez ensuite en savoir plus sur l'erreur et résoudre les problèmes pour les résoudre.

Les étapes suivantes vous montrent comment enregistrer et surveiller vos appareils Sidewalk et surveiller vos appareils Sidewalk et surveiller vos appareils Sidewalk et surveiller vos

1. Créez un rôle de journalisation pour enregistrer vos appareils Sidewalk, comme illustré dans l'exemple de politique suivant. Pour plus d'informations, voir [Créer un rôle et une politique de journalisation pour AWS IoT Wireless](#).

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

2. Pour obtenir des journaux moins détaillés, par exemple pour afficher uniquement les informations d'erreur, configurez la journalisation dans les journaux. Pour plus d'informations, voir [Configurer la journalisation des ressources AWS IoT sans fil](#).
3. Surveillez vos appareils Sidewalk en consultant les journaux dans la console CloudWatch Logs, en créant des filtres simples et en consultant les entrées des journaux dans les groupes Logs. Pour plus d'informations, voir [Afficher les entrées du journal CloudWatch AWS IoT Wireless](#).
4. Créez des expressions de filtre avancées en fonction des informations de journalisation que vous souhaitez obtenir, puis accédez à CloudWatch Insights pour filtrer les requêtes de journal et obtenir des informations supplémentaires. Pour plus d'informations, consultez [View CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless](#).

Pour obtenir des instructions détaillées, consultez la section [Surveillance et journalisation pour les réseaux AWS IoT sans fil](#).

Surveillance et journalisation pour AWS IoT Wireless l'utilisation d'Amazon CloudWatch

Vous pouvez surveiller vos AWS IoT Wireless ressources et vos applications qui s'exécutent en temps réel à l'aide d'Amazon CloudWatch. Vous pouvez surveiller l'état de vos appareils LoRa WAN et Sidewalk que vous avez intégrés à l'aide d'Amazon CloudWatch.

- Pour plus d'informations sur l'intégration des appareils LoRa WAN, consultez [Connexion de passerelles et d'appareils à AWS IoT Core for LoRaWAN \(p. 1276\)](#).
- Pour en savoir plus sur l'intégration des appareils Amazon Sidewalk à AWS IoT Core, consultez [Démarrer avec AWS IoT Core pour Amazon Sidewalk \(p. 1404\)](#).

CloudWatch Utilisez-le pour recueillir et suivre les métriques, qui sont des variables que vous pouvez mesurer pour vos ressources et applications. Pour plus d'informations sur les avantages de l'utilisation de la surveillance, consultez [Surveillance des AWS IoT \(p. 466\)](#).

Si vous souhaitez obtenir davantage d'informations de journal en temps réel à partir de vos périphériques LoRa WAN, utilisez l'analyseur de réseau. Pour plus d'informations, veuillez consulter [Surveillance de votre parc de ressources sans fil en temps réel à l'aide d'un analyseur de réseau \(p. 1374\)](#).

Comment surveiller vos ressources sans fil

Pour enregistrer et surveiller vos ressources sans fil, effectuez les opérations suivantes.

1. Créez un rôle de journalisation pour enregistrer vos AWS IoT Wireless ressources, comme décrit dans [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#).
2. Les messages de journal de la console CloudWatch Logs ont un niveau de journalisation par défaut de ERROR, qui est moins détaillé et ne contient que des informations d'erreur. Si vous souhaitez afficher des messages plus détaillés, nous vous recommandons d'utiliser d'abord l'interface de ligne de commande pour configurer la journalisation, comme décrit dans [Configurer la journalisation des AWS IoT Wireless ressources \(p. 1455\)](#).
3. Vous pouvez ensuite surveiller vos ressources en consultant les entrées du journal dans la console CloudWatch Logs. Pour plus d'informations, veuillez consulter [Afficher les entrées du CloudWatch AWS IoT Wireless journal \(p. 1464\)](#).
4. Vous pouvez créer des expressions de filtre à l'aide de groupes de journaux, mais nous vous recommandons de créer d'abord des filtres simples et d'afficher les entrées de journal dans les groupes de journaux, puis d'accéder à CloudWatch Insights pour créer des requêtes afin de filtrer les entrées du journal en fonction de la ressource ou de l'événement que vous surveillez. Pour plus d'informations, veuillez consulter [Utilisez CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless \(p. 1470\)](#).

Les rubriques suivantes expliquent comment configurer la journalisation AWS IoT Wireless et la collecte des métriques à partir de CloudWatch. Outre les appareils LoRa WAN, 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 l'ajout de ces appareils, consultez [AWS IoT Core pour Amazon Sidewalk \(p. 1400\)](#).

Rubriques

- [Configurer la journalisation pour AWS IoT Wireless \(p. 1453\)](#)
- [SurveillerAWS IoT Wireless à l'aide de CloudWatch journaux \(p. 1463\)](#)

Configurer la journalisation pour AWS IoT Wireless

Avant de pouvoir surveiller et enregistrerAWS IoT l'activité, activez d'abord la journalisation desAWS IoT Wireless ressources à l'aide de l'interface de ligne de commande ou de l'API.

Lorsque vous réfléchissez à la manière de configurer votreAWS IoT Wireless journalisation, la configuration de journalisation par défaut détermine la manière dontAWS IoT l'activité sera enregistrée, sauf indication contraire de votre part. Pour commencer, vous souhaiterez peut-être obtenir des journaux détaillés avec un niveau de journalisation par défaut deINFO.

Après avoir examiné les journaux initiaux, vous pouvez remplacer le niveau de journalisation par défautERROR, qui est moins détaillé, et définir un niveau de journalisation plus détaillé et spécifique à la ressource pour les ressources susceptibles de nécessiter plus d'attention. Les niveaux de journal peuvent être modifiés quand vous le souhaitez.

Les rubriques suivantes montrent comment configurer la journalisation desAWS IoT Wireless ressources.

Rubriques

- [Création d'un rôle et d'une politique de journalisation pourAWS IoT Wireless \(p. 1453\)](#)
- [Configurer la journalisation desAWS IoT Wireless ressources \(p. 1455\)](#)

Création d'un rôle et d'une politique de journalisation pourAWS IoT Wireless

L'exemple suivant illustre la création d'un rôle de journalisation pour lesAWS IoT Wireless ressources uniquement. Si vous souhaitez également créer un rôle de journalisation pourAWS IoT Core, consultez[Création d'un rôle de journalisation \(p. 467\)](#).

Créez un rôle de journalisation pourAWS IoT Wireless

Avant de pouvoir activer la journalisation, vous devez créer un rôle IAM et une politique qui vousAWS autorise à surveillerAWS IoT Wireless l'activité en votre nom.

Créez un rôle IAM pour la journalisation

Pour créer un rôle de journalisation pourAWS IoT Wireless, ouvrez le [hub Rôles de la console IAM](#) et choisissez Créer un rôle.

1. Sous Sélectionner le type d'entité de confiance, choisissez Un autreAWS compte.
2. Dans Identifiant du compte, saisissez votre identifiant deAWS compte, puis choisissez Suivant : Autorisations.
3. Dans la zone de recherche, saisissez **AWSIoTWirelessLogging**.
4. Cochez la case en regard de la politique nommée AWSIoTWirelessLogging, puis choisissez Suivant : Balises.
5. Choisissez Next: Review (Suivant : Vérification).
6. Dans Nom du rôle, entrez**IoTWirelessLogsRole**, puis choisissez Créez un rôle.

Modifier la relation de confiance du rôle IAM

Dans le message de confirmation qui s'affiche après avoir exécuté l'étape précédente, choisissez le nom du rôle que vous avez créé, IoTWirelessLogsRole. Vous allez ensuite modifier le rôle pour ajouter la relation de confiance suivante.

1. Dans la section Résumé du rôle IoTWirelessLogsRole, choisissez l'onglet Relations de confiance, puis choisissez Modifier la relation de confiance.
2. Dans le document de politique, modifiez laPrincipal propriété pour qu'elle ressemble à cet exemple.

```
"Principal": {  
    "Service": "iotwireless.amazonaws.com"  
},
```

Une fois laPrincipal propriété modifiée, le document de politique complet doit ressembler à 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, choisissez Mettre à jour la politique de confiance.

Politique de journalisation pour AWS IoT Wireless

Le document de politique suivant fournit la politique de rôle et la politique de confiance qui permettent AWS IoT Wireless de soumettre des entrées de journal CloudWatch en votre nom.

Note

Ce document de politique AWS géré a été automatiquement créé pour vous lorsque vous avez créé le rôle de journalisation, IoTWirelessLogsRole.

Politique de rôle

Ce qui suit montre le document de politique des rôles.

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

}

Politique de confiance visant à enregistrer uniquement les AWS IoT Wireless activités

Ce qui suit montre la politique de confiance pour la journalisation des AWS IoT Wireless activités uniquement.

```
{  
    "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 enregistrer AWS IoT Core l'activité, les documents de politique vous permettent de consigner les deux activités. Pour plus d'informations sur la création d'un rôle de journalisation pour AWS IoT Core, consultez [Création d'un rôle de journalisation \(p. 467\)](#).

Étapes suivantes

Vous avez appris à créer un rôle de journalisation pour enregistrer vos AWS IoT Wireless ressources. Par défaut, les journaux ont un niveau de journalisation de. Par conséquent, si vous ne souhaitez voir que les informations d'erreur, accédez [Afficher les entrées du CloudWatch AWS IoT Wireless journal \(p. 1464\)](#) à la section Surveiller vos ressources sans fil en consultant les entrées du journal.

Si vous souhaitez obtenir plus d'informations dans les entrées du journal, vous pouvez configurer le niveau de journalisation par défaut pour vos ressources ou pour différents types d'événements, par exemple en définissant le niveau de journalisation sur INFO. Pour plus d'informations sur la configuration de la journalisation de vos ressources, consultez [Configurer la journalisation des AWS IoT Wireless ressources \(p. 1455\)](#).

Configurer la journalisation des AWS IoT Wireless ressources

Pour configurer la journalisation AWS IoT Wireless des ressources, vous pouvez utiliser l'API ou la CLI. Lorsque vous commencez à surveiller AWS IoT Wireless les ressources, vous pouvez utiliser la configuration par défaut. Pour ce faire, vous pouvez ignorer cette rubrique et passer [Surveiller AWS IoT Wireless à l'aide de CloudWatch journaux \(p. 1463\)](#) à la surveillance de vos journaux.

Une fois que vous avez commencé à surveiller les journaux, vous pouvez utiliser l'interface de ligne de commande pour modifier les niveaux de journalisation et opter pour une option plus détaillée, telle que la fourniture d'ERR0R information INFO et l'activation de la journalisation pour davantage de ressources.

AWS IoT Wireless ressources et niveaux de journalisation

Avant d'utiliser l'API ou l'interface de ligne de commande, consultez le tableau suivant pour en savoir plus sur les différents niveaux de journalisation et les ressources pour lesquelles vous pouvez configurer la journalisation. Le tableau présente les paramètres que vous voyez dans les CloudWatch journaux.

lorsque vous surveillez les ressources. La façon dont vous configurez la journalisation pour vos ressources déterminera les journaux que vous verrez dans la console.

Pour plus d'informations sur l'apparence d'un exemple de CloudWatch journal et sur la manière dont vous pouvez utiliser ces paramètres pour enregistrer des informations utiles sur les AWS IoT Wireless ressources, consultez [Afficher les entrées du CloudWatch AWS IoT Wireless journal \(p. 1464\)](#).

Niveaux de journalisation et ressources

Nom	Valeurs possibles	Description
logLevel	INFO, ERROR ou DISABLED	<ul style="list-style-type: none"> • ERROR: affiche toute erreur entraînant l'échec d'une opération. Les journaux ne contiennent que ERROR des informations. • INFO: fournit des informations de haut niveau sur le flux de tâches. Les journaux incluent INFO des ERROR informations. • DISABLED: Désactiver toute la journalisation.
resource	WirelessGateway ou WirelessDevice	Le type de ressource, qui peut être WirelessGateway ou WirelessDevice.
wirelessGatewayType	LoRaWAN	Le type de passerelle sans fil, quand resource c'est le cas WirelessGateway, qui est toujours LoRa WAN.
wirelessDeviceType	LoRaWAN ou Sidewalk	Le type de périphérique sans fil, quand resource c'est le cas WirelessDevice, qui peut être LoRaWAN ou Sidewalk.
wirelessGatewayId		L'identifiant de la passerelle sans fil, quand resource est WirelessGateway.
wirelessDeviceId		L'identifiant du périphérique sans fil, quand resource est WirelessDevice.
event	JoinRejoin, Registration et Certificate	<p>Le type d'événement à enregistrer auquel le destinataire appartient, fait que la ressource que vous enregistrez est un périphérique sans fil ou une passerelle sans fil.</p> <p>Pour plus d'informations, veuillez consulter Afficher les entrées du CloudWatch AWS IoT Wireless journal (p. 1464).</p>

AWS IoT Wireless API de journalisation

Vous pouvez utiliser les actions d'API suivantes pour configurer la journalisation des ressources. Le tableau présente également un exemple de politique IAM que vous devez créer pour utiliser les actions d'API. La section suivante décrit l'utilisation des API pour configurer les niveaux de journalisation de vos ressources.

Journalisation des actions d'API

Nom d'API	Description	Exemple de politique IAM
GetLogLevelsByResourceType	Renvoie les niveaux de journalisation par défaut actuels, ou les niveaux de journalisation par type de ressource, qui peuvent inclure des options de	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": "iotwireless:DescribeLogLevels" }] }</pre>

Nom d'API	Description	Exemple de politique IAM
	journalisation pour les périphériques sans fil ou les passerelles sans fil.	<pre> "Effect": "Allow", "Action": ["iotwireless:GetLogLevelsByResourceTypes"], "Resource": ["*"] } </pre>
<u>GetResourceLogLevel</u>	Renvoie le remplacement au niveau du journal pour un identifiant de ressource et un type de ressource donnés. La ressource peut être un dispositif sans fil ou une passerelle sans fil.	<pre> { "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:GetResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ...] }] } </pre>

Nom d'API	Description	Exemple de politique IAM
<u>PutResourceLogLevel</u>	<p>Définit le remplacement au niveau du journal pour un identifiant de ressource et un type de ressource donnés. La ressource peut être une passerelle sans fil ou un dispositif sans fil.</p> <p>Note</p> <p>Cette API est limitée à 200 remplacements au niveau du journal par compte.</p>	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:PutResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a", ...] }] }</pre>
<u>ResetAllResourceLogLevels</u>	<p>Supprime les remplacements au niveau du journal pour toutes les ressources, y compris les passerelles sans fil et les périphériques sans fil.</p> <p>Note</p> <p>Cette API n'affecte pas les niveaux de journalisation définis à l'aide de l'UpdateLogLevelsByResourceType API.</p>	<pre>{ "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/*"] }] }</pre>

Nom d'API	Description	Exemple de politique IAM
<u>ResetResourceLogLevel</u>	Supprime le remplacement au niveau du journal pour un identifiant de ressource et un type de ressource donnés. La ressource peut être une passerelle sans fil ou un dispositif sans fil.	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:ResetResourceLogLevel"], "Resource": ["arn:aws:iotwireless:us-east-1:123456789012:WirelessDevice/012bc5ab12-cd3a-d00e-1f0e20c1204a",] }] }</pre>
<u>UpdateLogLevelsByResourceType</u>	<p>Définissez le niveau de journalisation par défaut, ou les niveaux de journalisation par type de ressource. Vous pouvez utiliser cette API pour les options de journalisation pour les appareils sans fil ou les passerelles sans fil, et contrôler les messages de journal qui y seront affichés CloudWatch.</p> <p>Note</p> <p>Les événements sont facultatifs et le type d'événement est lié au type de ressource. Pour plus d'informations, veuillez consulter Événements et types de ressources (p. 1465).</p>	<pre>{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Action": ["iotwireless:UpdateLogLevelsByResourceType"], "Resource": ["*"] }] }</pre>

Configurer les niveaux de journalisation des ressources à l'aide de l'interface de ligne de commande

Cette section explique comment configurer les niveaux de journalisation des AWS IoT Wireless ressources à l'aide de l'API ou AWS CLI.

Avant d'utiliser l'interface de ligne de commande :

- Assurez-vous d'avoir créé la politique 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, consultez [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#).

Pourquoi utiliser la AWS CLI

Par défaut, si vous créez le rôle IAMIoTWirelessLogsRole, comme décrit dans [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#), vous verrez CloudWatch des journaux AWS Management Console dont le niveau de journalisation par défaut est de ERROR. Pour modifier le niveau de journalisation par défaut pour toutes vos ressources ou pour des ressources spécifiques, utilisez l'API de AWS IoT Wireless journalisation ou l'interface de ligne de commande.

Comment utiliser la AWS CLI

Les actions d'API peuvent être classées dans les types suivants selon que vous souhaitez configurer des niveaux de journalisation pour toutes les ressources ou pour des ressources spécifiques :

- L'API `GetLogLevelByResourceTypes` agit et `UpdateLogLevelByResourceTypes` peut récupérer et mettre à jour les niveaux de journalisation de toutes les ressources de votre compte qui appartiennent à un type spécifique, telles qu'une passerelle sans fil, un périphérique LoRa WAN ou Sidewalk.
- L'API `GetResourceLogLevel` et `PutResourceLogLevel` peut récupérer `GetResourceLogLevel` et `PutResourceLogLevel`, mettre à jour et réinitialiser les niveaux de journalisation des ressources individuelles que vous spécifiez à l'aide d'un identifiant de ressource.
- L'action de l'API `ResetAllResourceLogLevels` réinitialise le remplacement au niveau du journal null pour toutes les ressources pour lesquelles vous avez spécifié un remplacement au niveau du journal à l'aide de l'API `PutResourceLogLevel`.

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.

- Par défaut, le niveau de journalisation de toutes les ressources est 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 la [update-log-levels-by-resource-types](#) commande. L'exemple suivant montre comment créer un fichier JSON et `Input.json` le fournir en tant qu'entrée à la commande CLI. Vous pouvez utiliser cette commande pour désactiver la journalisation de manière sélective ou pour remplacer le niveau de journalisation par défaut pour des types de ressources et d'événements spécifiques.

```
{  
    "DefaultLogLevel": "INFO",  
    "WirelessDeviceLogOptions":  
    [  
        {  
            "Type": "Sidewalk",  
            "LogLevel": "INFO",  
            "Events":  
            [  
                {  
                    "Event": "Registration",  
                    "LogLevel": "WARN"  
                }  
            ]  
        }  
    ]  
}
```

```
        "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

La liste des options de journalisation pour un périphérique sans fil. Chaque option de journal inclut le type de périphérique sans fil (Sidewalk ou LoRa WAN) et une liste d'options de journal des événements du périphérique sans fil. Chaque option de journal des événements du périphérique sans fil peut éventuellement inclure le type d'événement et son niveau de journalisation.

WirelessGatewayLogOptions

La liste des options de journalisation pour une passerelle sans fil. Chaque option de journal inclut le type de passerelle sans fil (LoRaWAN) et une liste d'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 journalisation.

DefaultLogLevel

Niveau de journalisation à utiliser pour toutes vos ressources. Les valeurs valides sont ERROR, INFO et DISABLED. La valeur par défaut est INFO.

LogLevel

Le niveau de journalisation que vous souhaitez utiliser pour des types de ressources et des événements individuels. Ces niveaux de journalisation remplacent le niveau de journalisation

par défaut, tel que le niveau de journalisation **INFO** pour la passerelle LoRa WAN, les niveaux de journalisation **DISABLED** et **ERROR** les deux types d'événements.

Exécutez la commande suivante pour fournir le `Input.json` fichier en entrée de la commande. Cette commande ne produit aucune sortie.

```
aws iotwireless update-log-levels-by-resource-types \
--cli-input-json file://input.json
```

Si vous souhaitez supprimer les options de journalisation pour les périphériques sans fil et les passerelles sans fil, exécutez la commande suivante.

```
{
  "DefaultLogLevel": "DISABLED",
  "WirelessDeviceLogOptions": [],
  "WirelessGatewayLogOptions": []
}
```

- La commande `update-log-levels-by-resource-types` ne renvoie aucune sortie. Utilisez la [get-log-levels-by-resource-types](#) commande pour récupérer les informations de journalisation spécifiques à la ressource. La commande renvoie le niveau de journalisation par défaut, ainsi que les options de journalisation du périphérique sans fil et de la passerelle sans fil.

Note

La `get-log-levels-by-resource-types` commande ne peut pas récupérer directement les niveaux de journalisation dans la CloudWatch console. Vous pouvez utiliser la `get-log-levels-by-resource-types` commande pour obtenir les dernières informations de niveau de journalisation que vous avez spécifiées pour vos ressources à l'aide de la `update-log-levels-by-resource-types` commande.

```
aws iotwireless get-log-levels-by-resource-types
```

Lorsque vous exécutez la commande suivante, elle renvoie les dernières informations de journalisation que vous avez spécifiées `update-log-levels-by-resource-types`. Par exemple, si vous supprimez les options de journalisation des périphériques sans fil, l'exécution de `get-log-levels-by-resource-types` renverra cette valeur sous la forme 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 journalisation des passerelles sans fil individuelles ou des ressources des appareils sans fil, utilisez les commandes CLI suivantes :

- [put-resource-log-level](#)
- [get-resource-log-level](#)
- [reset-resource-log-level](#)

À titre d'exemple indiquant quand utiliser ces CLI, supposons que votre compte comporte un grand nombre de périphériques ou de passerelles sans fil connectés. Si vous souhaitez résoudre les erreurs qui ne concernent que certains de vos appareils sans fil, vous pouvez désactiver la journalisation pour tous les appareils sans fil en définissant leDefaultLogLevel et en utilisant leput-resource-log-level pour définir leLogLevel to uniquementERROR pour les appareils de votre compte.DISABLED

```
aws iotwireless put-resource-log-level \
    --resource-identifier
    --resource-type WirelessDevice
    --log-level ERROR
```

Dans cet exemple, la commande définit le niveau de journalisationERROR uniquement pour la ressource de périphérique sans fil spécifiée et les journaux de toutes les autres ressources sont désactivés. Cette commande ne produit aucune sortie. Pour récupérer ces informations et vérifier que les niveaux de journalisation ont été définis, utilisez laget-resource-log-level commande.

4. À l'étape précédente, après avoir débogué le problème et résolu l'erreur, vous pouvez exécuter lareset-resource-log-level commande pour réinitialiser le niveau de journalisation de cette ressource ànull. Si vous avez utilisé laput -resource-log-level commande pour définir la dérogation au niveau du journal pour plusieurs périphériques sans fil ou ressources de passerelle, par exemple pour résoudre des erreurs sur plusieurs appareils, vous pouvez rétablir les remplacements au niveau du journalnull pour toutes ces ressources à l'aide de la [reset-all-resource-log-levels](#)commande.

```
aws iotwireless reset-all-resource-log-levels
```

Cette commande ne produit aucune sortie. Pour récupérer les informations de journalisation des ressources, exécutez laget-resource-log-level commande.

Étapes suivantes

Vous avez appris à créer le rôle de journalisation et à utiliser l'AWS IoT WirelessAPI pour configurer la journalisation de vosAWS IoT Wireless ressources. Ensuite, pour en savoir plus sur la surveillance de vos entrées de journal, rendez-vous sur[SurveillerAWS IoT Wireless à l'aide de CloudWatch journaux \(p. 1463\)](#).

SurveillerAWS IoT Wireless à l'aide de CloudWatch journaux

AWS IoT Wirelesscontient plus de 50 entrées de CloudWatch journal activées par défaut. Chaque entrée de journal décrit le type d'événement, le niveau de journalisation et le type de ressource. Pour plus d'informations, veuillez consulter [AWS IoT Wirelessressources et niveaux de journalisation \(p. 1455\)](#).

Comment contrôler vosAWS IoT Wireless ressources

Lorsque la journalisation est activée AWS IoT Wireless, AWS IoT Wireless envoie des événements de progression concernant chaque message au fur et à mesure de son transfert depuis AWS IoT et vers vos appareils. Par défaut, les entrées du AWS IoT Wireless journal présentent un niveau d'erreur par défaut. Lorsque vous activez la journalisation comme décrit dans [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#), la CloudWatch console affiche des messages dont le niveau de journalisation par défaut est de ERROR. En utilisant ce niveau de journalisation, 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 ceux dont le niveau de journalisation est de INFO, ou si vous souhaitez désactiver les journaux pour certains de vos appareils et afficher des messages de journal uniquement pour certains de vos appareils, vous pouvez utiliser l'API de AWS IoT Wireless journalisation. Pour plus d'informations, veuillez consulter [Configurer les niveaux de journalisation des ressources à l'aide de l'interface de ligne de commande \(p. 1459\)](#).

Vous pouvez également créer des expressions de filtre pour n'afficher que les messages requis.

Avant de pouvoir afficher AWS IoT Wireless les journaux dans la console

Pour que le groupe de journaux /aws/iotwireless apparaisse dans la CloudWatch console, vous devez avoir effectué les opérations suivantes.

- Connexion activée AWS IoT Wireless. Pour plus d'informations sur l'activation de la connexion AWS IoT Wireless, consultez [Configurer la journalisation pour AWS IoT Wireless \(p. 1453\)](#).
- A écrit certaines entrées de journal en effectuant AWS IoT Wireless des opérations.

Pour créer et utiliser des expressions de filtre plus efficacement, nous vous recommandons d'essayer d'utiliser les CloudWatch informations décrites dans les rubriques suivantes. Nous vous recommandons également de suivre les sujets dans l'ordre dans lequel ils sont présentés ici. Cela vous aidera à utiliser d'abord les groupes de CloudWatch journaux pour en savoir plus sur les différents types de ressources, leurs types d'événements et les niveaux de journalisation que vous pouvez utiliser pour afficher les entrées du journal dans la console. Vous pouvez ensuite apprendre à créer des expressions de filtre à l'aide d'CloudWatch Insights pour obtenir des informations plus utiles à partir de vos ressources.

Rubriques

- [Afficher les entrées du CloudWatch AWS IoT Wireless journal \(p. 1464\)](#)
- [Utilisez CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless \(p. 1470\)](#)

Afficher les entrées du CloudWatch AWS IoT Wireless journal

Après avoir configuré la journalisation AWS IoT Wireless comme décrit dans [Création d'un rôle et d'une politique de journalisation pour AWS IoT Wireless \(p. 1453\)](#) et écrit certaines entrées de journal, vous pouvez consulter les entrées du journal dans la CloudWatch console en effectuant les étapes suivantes.

Affichage AWS IoT des journaux dans la console CloudWatch Log groups

Dans la [CloudWatch console](#), CloudWatch les journaux apparaissent dans un groupe de journaux nommé /aws/iotwireless. Pour plus d'informations sur les CloudWatch journaux, consultez la section [CloudWatch Journaux](#).

Pour consulter vos AWS IoT journaux dans la CloudWatch console

Accédez à la [CloudWatch console](#) et choisissez Log groups dans le volet de navigation.

1. Dans la zone de texte Filtrer, entrez/**aws/iotwireless**, puis choisissez le groupe de **/aws/iotwireless** journaux.
2. Pour voir la liste complète des AWS IoT Wireless journaux générés pour votre compte, choisissez Tout rechercher. Pour consulter un flux de journal individuel, cliquez sur l'icône d'expansion.
3. Pour filtrer les flux de journaux, vous pouvez également saisir une requête dans la zone de texte Filtrer les événements. Voici quelques requêtes à essayer :

• `{ $.logLevel = "ERROR" }`

Utilisez ce filtre pour rechercher tous les journaux dont le niveau de ERROR journalisation est égal à. Vous pouvez étendre les flux d'erreurs individuels pour lire les messages d'erreur, ce qui vous aidera à les résoudre.

• `{ $.resource = "WirelessGateway" }`

Trouvez tous les journaux de la `WirelessGateway` ressource, quel que soit le niveau de journalisation.

• `{ $.event = "CUPS_Request" && $.logLevel = "ERROR" }`

Recherchez tous les journaux dont le type d'événement `CUPS_Request` et le niveau de journalisation sont `ERROR`.

Événements et types de ressources

Le tableau suivant indique 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 journalisation par défaut pour les ressources et les types d'événements ou remplacer le niveau de journalisation par défaut en spécifiant un niveau de journalisation pour chacun d'entre eux.

Types d'événements en fonction des ressources utilisées

Ressource	Type de ressource	Type d'événement	
Passerelle sans fil	LoRaWAN	<ul style="list-style-type: none">• Demande CUPS_Requête• Certificat	
Appareil WiWiWireless	LoRaWAN	<ul style="list-style-type: none">• Joindre• Rejoindre• Uplink_Data• Données_descendante	
Appareil WiWiWireless	Trottoir	<ul style="list-style-type: none">• Inscription• Uplink_Data• Données_descendante	

La rubrique suivante contient des informations supplémentaires sur ces types d'événements et les entrées de journal pour les passerelles sans fil et les périphériques sans fil.

Rubriques

- [Entrées de journal pour les passerelles sans fil et les ressources des appareils sans fil \(p. 1466\)](#)

Entrées de journal pour les passerelles sans fil et les ressources des appareils sans fil

Une fois que vous avez activé la journalisation, vous pouvez consulter les entrées du journal pour vos passerelles et appareils sans fil. La section suivante décrit les différents types d'entrées de journal en fonction de vos types de ressources et d'événements.

Entrées du journal de passerelle sans fil

Cette section présente certains des exemples d'entrées de journal relatives aux ressources de votre passerelle sans fil que vous verrez dans la [CloudWatch console](#). Ces messages de journal peuvent avoir un type d'événementCUPS_Request égal ouCertificate égal à et peuvent être configurés pour afficher un niveau de journalisation égal àINFOERROR, ouDISABLED au niveau des ressources ou au niveau de l'événement. Si vous souhaitez afficher uniquement les informations d'erreur, définissez le niveau de journalisation surERROR. Le message contenu dans l'entrée duERROR journal contiendra des informations sur la raison de l'échec.

Les entrées de journal de votre ressource de passerelle sans fil peuvent être classées en fonction des types d'événements suivants :

- Demande CUPS_Requête

La LoRa Basics Station exécutée sur votre passerelle envoie régulièrement 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 journalisation surINFO lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, alors dans les journaux :

- Si l'événement est réussi, vous verrez des messages de journal contenant unlogLevel deINFO. Les messages incluront des détails sur la réponse CUPS envoyée à votre passerelle et les détails de la passerelle. Vous trouverez ci-dessous un exemple de cette entrée de journal. Pour plus d'informations sur les champslogLevel et les autres champs de l'entrée de journal, consultez[AWS IoT Wireless ressources et niveaux de journalisation \(p. 1455\)](#).

```
{
  "timestamp": "2021-05-13T16:56:08.853Z",
  "resource": "WirelessGateway",
  "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
  "wirelessGatewayType": "LoRaWAN",
  "gatewayEui": "feffff0000000e2",
  "event": "CUPS_Request",
  "logLevel": "INFO",
  "message": "Sending CUPS response of total length 3213 to GatewayEui: feffff0000000e2 with TC Credentials,"
}
```

- En cas d'erreur, vous verrez les entrées du journal comportant unlogLevel ofERROR, et les messages contiendront des détails sur l'erreur. Une erreur peut notamment survenir lors de l'CUPS_Request événement : CUPS CRC manquant, inadéquation entre l'URI TC de la passerelle et l'enregistrement de la passerelle sans filAWS IoT Wireless IoTWirelessGatewayCertManagerRole, absence ou impossibilité d'obtenir un 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 vous assurer que vous avez saisi le bon CRC CUPS.

```
{
  "timestamp": "2021-05-13T16:56:08.853Z",
  "resource": "WirelessGateway",
  "wirelessGatewayId": "5da85cc8-3361-4c79-8be3-3360fb87abda",
  "wirelessGatewayType": "LoRaWAN",
```

```
"gatewayEui": "feffff0000000e2",
"event": "CUPS_Request",
"logLevel": "ERROR",
"message": "The CUPS CRC is missing from the request. Check your gateway setup and
enter the CUPS CRC,"
}
```

- Certificat

Ces entrées de journal vous aideront à vérifier si votre passerelle sans fil présentait le bon certificat pour authentifier la connexion AWS IoT. Pour ce type d'événement, si vous définissez le niveau de journalisation sur INFO lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, alors dans les journaux :

- Si l'événement est réussi, vous verrez des messages de journal contenant un logLevel de INFO. Les messages incluront des détails sur l'identifiant du certificat et l'identifiant de la passerelle sans fil. Vous trouverez ci-dessous un exemple de cette entrée de journal. Pour plus d'informations sur les champs `logLevel` et les autres champs de l'entrée de journal, consultez [AWS IoT Wireless ressources et niveaux de journalisation \(p. 1455\)](#).

```
{
  "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)"}
```

- En cas d'erreur, vous verrez les entrées du journal comportant un logLevel de ERROR, et les messages contiendront des détails sur l'erreur. Par exemple, une erreur peut survenir lors de l'Certificate événement : un ID de certificat, un identifiant de passerelle sans fil non valide ou une incompatibilité entre l'identifiant de la passerelle sans fil et l'identifiant du certificat. L'exemple suivant montre un identifiant de passerelle sans fil non valide ERROR en raison d'un identifiant de passerelle sans fil. Pour résoudre l'erreur, vérifiez les identificateurs 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 du journal de périphérique sans fil

Cette section présente certains des exemples d'entrées de journal relatives aux ressources de votre appareil sans fil que vous verrez dans la [CloudWatch console](#). Le type d'événement associé à ces messages de journal varie selon que vous utilisez un périphérique LoRa WAN ou Sidewalk. Chaque type de ressource ou d'événement de périphérique sans fil peut être configuré pour afficher un niveau de journalisation de INFO, ERROR, ou DISABLED.

Note

Votre demande ne doit pas contenir à la fois des métadonnées LoRa WAN et des métadonnées sans fil Sidewalk. Pour éviter une entrée deERROR journal dans ce scénario, spécifiez les données sans fil LoRa WAN ou Sidewalk.

LoRaEntrées du journal des périphériques WAN

Les entrées du journal de votre appareil sans fil LoRa WAN peuvent être classées en fonction des types d'événements suivants :

- **Join et Rejoin**

Lorsque vous ajoutez un périphérique LoRa WAN et que vous le connectezAWS IoT Wireless, avant que votre appareil puisse envoyer des données de liaison montante, vous devez terminer un processus appeléactivation oujoin procedure. Pour plus d'informations, veuillez consulter [Ajoutez votre appareil sans fil àAWS IoT Core for LoRaWAN \(p. 1289\)](#).

Pour ce type d'événement, si vous définissez le niveau de journalisation surINFO lors de la configuration de l'interface de ligne de commande pour votre ressource de passerelle sans fil, alors dans les journaux :

- Si l'événement est réussi, vous verrez des messages de journal contenant unLogLevel deINFO. Les messages incluront des détails sur l'état de votre demande d'adhésion ou de réintégration. Vous trouverez ci-dessous un exemple de cette entrée de journal. Pour plus d'informations sur les champslogLevel et les autres champs de l'entrée de journal, consultez[AWS IoT Wirelessressources et niveaux de journalisation \(p. 1455\)](#).

```
{  
    "timestamp": "2021-05-13T16:56:08.853Z",  
    "resource": "WirelessDevice",  
    "wirelessDeviceType": "LoRaWAN",  
    "wirelessDeviceId": "5da85cc8-3361-4c79-8be3-3360fb87abda",  
    "devEui": "feffff00000000e2",  
    "event": "Rejoin",  
    "logLevel": "INFO",  
    "message": "Rejoin succeeded"  
}
```

- En cas d'erreur, vous verrez les entrées du journal comportant unLogLevel ofERROR, et les messages contiendront des détails sur l'erreur. Une erreur peut survenir pour lesRejoin événementsJoin et, par exemple, un paramètre de région LoRa WAN non valide ou une vérification du code d'intégrité des messages (MIC) non valide. L'exemple suivant montre une erreur de jointure due à une vérification du MIC. Pour résoudre l'erreur, vérifiez si vous avez saisi 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

Le type d'événementUplink_Data est utilisé pour les messages générésAWS IoT Wireless lorsque la charge utile est envoyée du périphérique sans fil àAWS IoT. Le type d'événementDownlink_Data

est utilisé pour les messages liés à des messages de liaison descendante AWS IoT envoyés depuis le périphérique sans fil.

Note

Les événements Uplink_Data et Downlink_Data s'appliquent à la fois aux appareils LoRa WAN et Sidewalk.

Pour ce type d'événement, si vous définissez le niveau de journalisation sur INFO lors de la configuration de l'interface de ligne de commande pour vos appareils sans fil, vous verrez dans les journaux :

- Si l'événement est réussi, vous verrez des messages de journal contenant un logLevel de INFO. Les messages comprendront des détails sur l'état du message de liaison montante ou descendante qui a été envoyé et sur l'identifiant du périphérique sans fil. Voici un exemple de cette entrée de journal pour un appareil Sidewalk. Pour plus d'informations sur les champs LogLevel et les autres champs de l'entrée de journal, consultez [AWS IoT Wireless ressources et niveaux de journalisation \(p. 1455\)](#).

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

- En cas d'erreur, vous verrez les entrées du journal comportant un LogLevel of ERROR, et les messages contiendront des détails sur l'erreur, ce qui vous aidera à la résoudre. Une erreur peut survenir lors de l'Registration événement : problèmes d'authentification, demandes non valides ou trop nombreuses, impossibilité de chiffrer ou de déchiffrer la charge utile, ou impossibilité 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 du journal de l'appareil de trottoir

Les entrées du 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 auprès desquels vous vous inscrivez AWS IoT Wireless. Pour ce type d'événement, si vous définissez le niveau de journalisation sur INFO lors de la configuration de l'interface de ligne de commande pour la ressource de votre appareil sans fil, vous verrez apparaître dans les journaux des messages de journal

contenant un logLevel de INFO et ERROR. Les messages incluront des détails sur la progression de l'enregistrement du début à la fin. Les messages de journal contiennent des informations sur la résolution des problèmes liés à l'enregistrement de votre appareil.

Vous trouverez ci-dessous un exemple de message de journal dont le niveau de journalisation est de INFO. Pour plus d'informations sur les champs `logLevel` et les autres champs de l'entrée de journal, consultez [AWS IoT Wireless ressources et niveaux de journalisation \(p. 1455\)](#).

```
{  
    "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

Les types d'événements Uplink_Data et Downlink_Data pour les périphériques Sidewalk sont similaires aux types d'événements correspondants pour les appareils LoRa WAN. Pour plus d'informations, reportez-vous aux sections Uplink_Data et Downlink_Data décrites précédemment pour les entrées du journal des périphériques LoRa WAN.

Étapes suivantes

Vous avez appris à consulter les entrées de journal de vos ressources et les différentes entrées de journal que vous pouvez consulter dans la CloudWatch console après avoir activé la journalisation AWS IoT Wireless. Bien que vous puissiez créer des flux de filtres à l'aide de groupes de journaux, nous vous recommandons d'utiliser CloudWatch Insights pour créer et utiliser des flux de filtres. Pour plus d'informations, veuillez consulter [Utilisez CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless \(p. 1470\)](#).

Utilisez CloudWatch Insights pour filtrer les journaux pour AWS IoT Wireless

Bien que vous puissiez utiliser CloudWatch les journaux pour créer des expressions de filtre, nous vous recommandons d'utiliser CloudWatch des informations pour créer et utiliser plus efficacement des expressions de filtre en fonction de votre application.

Nous vous recommandons d'utiliser d'abord les groupes de CloudWatch journaux pour en savoir plus sur les différents types de ressources, leurs types d'événements et les niveaux de journalisation que vous pouvez utiliser pour afficher les entrées du journal dans la console. Vous pouvez ensuite utiliser les exemples de certaines expressions de filtre de cette page comme référence pour créer vos propres filtres pour vos AWS IoT Wireless ressources.

Affichage AWS IoT des journaux dans la console CloudWatch Logs Insights

Dans la [CloudWatch console](#), CloudWatch les journaux apparaissent dans un groupe de journaux nommé /aws/iotwireless. Pour plus d'informations sur les CloudWatch journaux, consultez la section [CloudWatch Journaux](#).

Pour consulter vos AWS IoT journaux dans la CloudWatch console

Accédez à la [CloudWatch console](#) et choisissez Logs Insights dans le volet de navigation.

1. Dans la zone de texte Filtrer, saisissez/**aws/iotwireless**, puis choisissez/aws/iotwireless Logs Insights.
2. Pour voir la liste complète des groupes de journaux, choisissez Sélectionner le ou les groupes de journaux. Pour consulter les groupes de journaux pour AWS IoT Wireless, sélectionnez/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 mieux comprendre les indicateurs de vos ressources.

Créez des requêtes utiles pour filtrer et obtenir des informations sur AWS IoT Wireless

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, tels qu'une passerelle LoRa WAN ou un périphérique Sidewalk. Par exemple, pour filtrer les journaux afin d'afficher uniquement les messages des appareils Sidewalk, vous pouvez saisir la requête suivante et choisir Exécuter la requête. Pour enregistrer cette requête, choisissez Save (Enregistrer).

```
fields @message
| filter @message like /Sidewalk/
```

Une fois la requête exécutée, vous verrez les résultats dans l'onglet Journaux, qui affiche les horodatages des journaux relatifs aux appareils Sidewalk de votre compte. Vous verrez également un graphique à barres, qui indiquera l'heure à laquelle les événements se sont produits, s'ils se sont déjà produits en lien avec votre appareil Sidewalk. L'exemple suivant montre comment vous développez l'un des résultats dans l'onglet Journaux. Sinon, si vous souhaitez résoudre les erreurs liées aux appareils Sidewalk, vous pouvez ajouter un autre filtre qui définit le niveau de journalisationERROR et affiche 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": "feffff000000011a", "event": "Downlink_Data", "logLevel": "INFO", "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda", "message": "Successfully sent downlink message. Amazon SidewalkId = 20000000006, Sequence number = 0" } @timestamp 1623894967640 devEui feffff000000011a 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 souhaitez savoir quand votre message de liaison descendante a été envoyé depuis votre périphérique sans fil LoRa WAN, vous pouvez saisir la requête suivante et choisir Exécuter la requête. Pour enregistrer cette requête, choisissez Save (Enregistrer).

```
filter @message like /Downlink message sent/
```

Une fois la requête exécutée, vous verrez les résultats dans l'onglet Journaux, qui indique l'heure à laquelle le message de liaison descendante a été correctement envoyé à votre appareil sans fil. Vous verrez également un graphique à barres indiquant l'heure à laquelle un message de liaison descendante a été envoyé, si des messages de liaison descendante ont déjà été envoyés à votre appareil sans fil. L'exemple suivant montre comment vous développez l'un des résultats dans l'onglet Journaux. Sinon, si aucun message de lien descendant n'a été envoyé, vous pouvez modifier la requête pour afficher uniquement les résultats correspondant aux moments où le message n'a pas été envoyé afin de résoudre le problème.

Field	Value
@ingestionTime	1623884043676
@log	954314929104:/aws/iotwireless
@logStream	WirelessDevice-
Downlink_Data-42d0e6d09ba4d7015f4e9756fc616d401cd85fe3ac19854d9fb866153c872	
@message	{ "timestamp": "2021-06-16T22:54:00.770493863Z", "resource": "WirelessDevice", "wirelessDeviceId": "3b058d05-4e84-4e1a-b026-4932bddf978d", "wirelessDeviceType": "LoRaWAN", "devEui": "feffff000000011a", "event": "Downlink_Data", "logLevel": "INFO", "messageId": "7e752a10-28f5-45a5-923f-6fa7133fedda", "message": "Downlink message sent. MessageId: 7e752a10-28f5-45a5-923f-6fa7133fedda" }
@timestamp	1623884040858
devEui	feffff000000011a
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 du journal. Vous pouvez combiner certains des filtres décrits précédemment et concevoir vos propres filtres en fonction de la ressource que vous surveillez. Pour plus d'informations sur l'utilisation d' CloudWatch Insights, consultez la section [Analyse 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 bouton Historique dans la console CloudWatch Logs Insights, vous pouvez consulter les requêtes précédemment exécutées et les réexécuter si nécessaire, ou les modifier davantage en créant des requêtes supplémentaires.

Notifications d'événements pour AWS IoT Wireless

AWS IoT Wireless peut publier des messages pour vous informer des événements concernant les appareils LoRa WAN et Sidewalk auxquels vous êtes connecté à AWS IoT Core. Par exemple, vous pouvez être informé d'événements tels que la mise en service ou l'enregistrement des appareils Sidewalk de votre compte.

Comment vos ressources peuvent être informées des événements

Des notifications d'événements sont publiées lorsque certains événements se produisent. Par exemple, des événements sont générés lorsque votre appareil Sidewalk est approvisionné. Chaque événement entraîne l'envoi d'une seule notification d'événement. Les notifications d'événements sont publiées via MQTT avec une charge utile JSON. Le contenu de la charge utile dépend du type d'événement.

Note

Les notifications d'événements sont publiées au moins une fois. Il est possible qu'ils soient publiés plus d'une fois. L'ordre des notifications d'événements n'est pas garanti.

Événements et types de ressources

Le tableau suivant indique les différents types d'événements pour lesquels vous recevrez des notifications. Les types d'événements varient selon que le type de ressource est un appareil sans fil, une passerelle sans fil ou un compte Sidewalk. Vous pouvez également activer les événements pour vos ressources au niveau des ressources, qui s'appliquent à toutes les ressources d'un type particulier, ou à certaines ressources, comme décrit dans la section suivante. Pour plus d'informations sur les différents types d'événements, consultez [Notifications d'événements pour les ressources LoRa WAN \(p. 1481\)](#) et [Notifications d'événements pour les ressources Sidewalk \(p. 1486\)](#).

Types d'événements en fonction des ressources

Ressource	Type de ressource	Type d'événement	
Appareil sans fil	LoRaWAN	Joindre	
	Trottoir	<ul style="list-style-type: none">• État d'enregistrement de périphérique• Proximité• Statut de distribution des messages	
Passerelle sans fil	LoRaWAN	État de la connexion	
Compte Sidewalk	Trottoir	<ul style="list-style-type: none">• État d'enregistrement de périphérique• Proximité• Statut de distribution des messages	

Politique de réception de notifications d'événements sans fil

Pour recevoir des notifications d'événements, votre appareil doit appliquer une politique appropriée lui permettant de se connecter à la passerelle de l'AWS IoT et de s'abonner aux sujets d'événements MQTT. Vous devez aussi vous abonner aux filtres de rubriques appropriés.

Voici un exemple de la politique requise pour recevoir des notifications pour les différents événements sans fil.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Subscribe",  
                "iot:Receive"  
            ],  
            "Resource": [  
                "arn:aws:iotwireless:$region:$account:$aws/iotwireless/events/join/*",  
                "arn:aws:iotwireless:$region:$account:$aws/iotwireless/events/connection_status/  
*",  
                "arn:aws:iotwireless:$region:$account:$aws/iotwireless/events/  
device_registration_state/*",  
                "arn:aws:iotwireless:$region:$account:$aws/iotwireless/events/proximity/*",  
                "arn:aws:iotwireless:$region:$account:$aws/iotwireless/events/  
message_delivery_status/*"  
            ]  
        }]  
    }  
}
```

Format des rubriques MQTT pour les événements sans fil

Pour vous envoyer des notifications d'événements concernant vos ressources sans fil, AWS IoT utilise les rubriques réservées MQTT qui commencent par le signe dollar (\$). Vous pouvez publier et vous abonner à ces sujets réservés. Cependant, vous ne pouvez pas créer de nouveaux sujets qui commencent par le signe du dollar.

Note

Les rubriques MQTT vous sont spécifiques au Compte AWS et utilisent le format `arn:aws:iotwireless:$aws-region:$AWS-account-ID:topic/Topic`. Pour plus d'informations, veuillez consulter [Rubriques MQTT \(p. 115\)](#).

Les rubriques MQTT réservées aux appareils sans fil utilisent le format suivant :

- Rubriques au niveau des ressources

Ces rubriques s'appliquent à toutes les ressources d'un type particulier auquel vous êtes connecté au Compte AWS IoT Wireless.

`$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/resources`

- Sujets au niveau de l'identifiant

Ces rubriques s'appliquent à la sélection de ressources d'un type particulier dans votre Compte AWS navigateur auxquelles vous êtes connecté au Compte AWS IoT Wireless, en fonction de l'identifiant de la ressource.

`$aws/iotwireless/events/{eventName}/{eventType}/{resourceType}/
{resourceIdentifierType}/{resourceID}/{id}`

Pour plus d'informations sur les rubriques au niveau des ressources et des identifiants, consultez[Configurations d'événements \(p. 1478\)](#).

Le tableau suivant présente des exemples de rubriques MQTT pour les différents événements :

Événements et sujets relatifs au MQTT

Événement	Rubrique MQTT	Remarques
État d'enregistrement de l'appareil Sidewalk	<ul style="list-style-type: none"> Rubrique au niveau des ressources <code>\$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/wireless_devices</code> Rubrique au niveau de l'identifiant <code>\$aws/iotwireless/events/device_registration_state/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}</code> 	<ul style="list-style-type: none"> {eventType} peut être registered ou provisioned {resourceType} peut être sidewalk_accounts ou wireless_devices {resourceID} est le amazon_id pour sidewalk_accounts et le wireless_device_id pour wireless_devices
Proximité des trottoirs	<ul style="list-style-type: none"> Rubrique au niveau des ressources <code>\$aws/iotwireless/events/proximity/{eventType}/sidewalk/wireless_devices</code> Rubrique au niveau de l'identifiant <code>\$aws/iotwireless/events/proximity/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}</code> 	<ul style="list-style-type: none"> {eventType} peut être beacon_discovered ou beacon_lost {resourceType} peut être sidewalk_accounts ou wireless_devices {resourceID} est le amazon_id pour sidewalk_accounts et le wireless_device_id pour wireless_devices
Statut de distribution des messages sur Sidewalk	<ul style="list-style-type: none"> Rubrique au niveau des ressources <code>\$aws/iotwireless/events/message_delivery_status/{resourceType}/{resourceID}/{id}</code> 	<ul style="list-style-type: none"> {eventType} peut être success ou error {resourceType} peut être sidewalk_accounts ou wireless_devices {resourceID} est le amazon_id pour sidewalk_accounts

Événement	Rubrique MQTT	Remarques
	<ul style="list-style-type: none"> {eventType}/sidewalk/wireless_devices Rubrique au niveau de l'identifiant <ul style="list-style-type: none"> \$aws/iotwireless/events/message_delivery_status/{eventType}/sidewalk/{resourceType}/{resourceID}/{id} 	et le wireless_device_id pour wireless_devices
LoRaConnexion à un réseau WAN	<ul style="list-style-type: none"> Rubrique au niveau des ressources <ul style="list-style-type: none"> \$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices Rubrique au niveau de l'identifiant <ul style="list-style-type: none"> \$aws/iotwireless/events/join/{eventType}/lorawan/wireless_devices/{resourceID}/{id} 	<ul style="list-style-type: none"> {eventType} peut être join_req_0_received/join_req_2_received ou join_accepted {resourceID} peut être wireless_device_id ou dev_eui
LoRaÉtat de connexion à la passerelle WAN	<ul style="list-style-type: none"> Rubrique au niveau des ressources <ul style="list-style-type: none"> \$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways Rubrique au niveau de l'identifiant <ul style="list-style-type: none"> \$aws/iotwireless/events/join/{eventType}/lorawan/wireless_gateways/{resourceID}/{id} 	<ul style="list-style-type: none"> {eventType} peut être connected ou disconnected {resourceID} peut être wireless_gateway_id ou gateway_eui

Pour plus d'informations sur les différents événements, consultez[Notifications d'événements pour les ressources LoRa WAN \(p. 1481\)](#) et[Notifications d'événements pour les ressources Sidewalk \(p. 1486\)](#).

Si vous êtes abonné à ces sujets, vous serez averti lorsqu'un message est publié sur l'un des sujets de notification d'événements. Pour plus d'informations, veuillez consulter[Rubriques réservées \(p. 117\)](#).

Tarification des événements sans fil

Pour plus d'informations sur les tarifs d'abonnement à des événements et de réception de notifications, consultez la section [AWS IoT Core Tarifs](#).

Activer les événements pour les ressources sans fil

Avant que les abonnés aux sujets réservés puissent recevoir des messages, vous devez activer les notifications d'événements. Pour ce faire, vous pouvez utiliser le AWS Management Console, ou l'AWS IoT Wireless API ou AWS CLI.

Configurations d'événements

Vous pouvez configurer des événements pour envoyer des notifications à toutes les ressources appartenant à un type particulier ou à des ressources sans fil individuelles. Le type de ressource peut être une passerelle sans fil, un compte partenaire Sidewalk ou un appareil sans fil, qui peut être un périphérique LoRa WAN ou Sidewalk. Pour plus d'informations sur les types d'événements que vous pouvez activer pour vos appareils sans fil, consultez [Types d'événements pour les ressources LoRa WAN \(p. 1481\)](#) et [Types d'événements pour les ressources Sidewalk \(p. 1486\)](#).

Toutes les ressources

Vous pouvez activer des événements de telle sorte que toutes les ressources de votre Compte AWS qui appartiennent à un type de ressource particulier reçoivent des notifications. Par exemple, vous pouvez activer un événement qui vous informe des modifications de l'état de connexion pour toutes les passerelles LoRa WAN auxquelles vous êtes connecté AWS IoT Core for LoRaWAN. La surveillance de ces événements vous aidera à être averti en cas de déconnexion de certaines passerelles LoRa WAN de votre parc de ressources ou de perte d'une balise associée à un certain nombre d'appareils Sidewalk de votre parc de ressources Compte AWS.

Ressources individuelles

Vous pouvez également ajouter des ressources LoRa WAN et Sidewalk individuelles à la configuration de votre événement et activer les notifications correspondantes. Cela vous aidera à surveiller les ressources individuelles d'un type particulier. Par exemple, vous pouvez ajouter certains appareils LoRa WAN et Sidewalk à votre configuration et recevoir des notifications concernant les événements relatifs à l'état de connexion ou d'enregistrement des appareils pour ces ressources.

Prérequis

Votre ressource LoRa WAN ou Sidewalk doit disposer d'une politique appropriée lui permettant de recevoir des notifications d'événements. Pour plus d'informations, veuillez consulter [Politique de réception de notifications d'événements sans fil \(p. 1475\)](#).

Activez les notifications à l'aide du AWS Management Console

Pour activer les messages d'événements depuis la console, accédez à l'onglet [Paramètres](#) de la AWS IoT console, puis accédez à la section des notifications d'événements LoRa WAN et Sidewalk.

Vous pouvez activer les notifications pour toutes les ressources de votre Compte AWS qui appartiennent à un type de ressource particulier et les surveiller.

Pour activer les notifications pour toutes les ressources

1. Dans la section Notification des événementsLoRa WAN et Sidewalk, accédez à l'onglet Toutes les ressources, choisissez Action, puis choisissez Gérer les événements.
2. Activez les événements que vous souhaitez surveiller, puis choisissez Mettre à jour les événements. Si vous ne souhaitez plus surveiller certains événements, choisissez Action, puis choisissez Gérer les événements, puis désactivez ces événements.

Vous pouvez également activer les notifications pour les ressources individuelles de votreCompte AWS compte qui appartient à un type de ressource particulier et les surveiller.

Pour activer les notifications pour des ressources individuelles

1. Dans la section Notification des événementsLoRa WAN et Sidewalk, choisissez Action, puis choisissez Ajouter des ressources.
2. Sélectionnez les ressources et les événements pour lesquels vous souhaitez recevoir des notifications :
 - a. Choisissez si vous souhaitez surveiller les événements relatifs à vos ressourcesLoRa WAN ou à vos ressources Sidewalk.
 - b. En fonction du type de ressource, vous pouvez choisir les événements que vous souhaitez activer pour les ressources. Vous pouvez ensuite vous inscrire à ces événements et recevoir des notifications. Si vous choisissez :
 - LoRaRessources WAN : vous pouvez activer les événements de participation pour vos appareils LoRa WAN ou les événements d'état de connexion pour vos passerelles LoRa WAN.
 - Ressources Sidewalk : vous pouvez activer l'enregistrement des appareils, l'état ou les événements de proximité, ou les deux, pour vos comptes partenaires Sidewalk et vos appareils Sidewalk.

Note

La configuration de l'événement d'état de distribution de message Sidewalk n'est pas disponible dans la console. Il ne peut être activé qu'à l'aide de l'AWS IoT WirelessAPI ou duAWS CLI.

3. En fonction du type de ressource et des événements que vous avez choisis, sélectionnez les périphériques ou passerelles sans fil que vous souhaitez surveiller. Vous pouvez sélectionner jusqu'à 250 ressources pour toutes les ressources combinées.
4. Choisissez Soumettre pour ajouter vos ressources.

Les ressources que vous ajoutez apparaîtront avec leurs rubriques MQTT dans l'onglet correspondant à votre type de ressource dans la section des notifications d'événementsLoRa WAN et Sidewalk de la console.

- LoRaLes événements de connexion au réseau WAN et les événements pour vos appareils Sidewalk apparaîtront dans la section Appareils sans fil de la console.
- Les événements relatifs à l'état de connexion de vos passerelles LoRa WAN apparaîtront dans la section Passerelles sans fil.
- L'état d'enregistrement de l'appareil et les événements de proximité associés à vos comptes Sidewalk apparaîtront dans l'onglet Comptes Sidewalk.

Abonnez-vous à des sujets à l'aide du client MQTT

Selon que vous avez activé les événements pour toutes les ressources ou pour des types de ressources individuels, les événements que vous avez activés apparaîtront dans la console avec leurs rubriques MQTT dans l'onglet Toutes les ressources ou dans l'onglet correspondant au type de ressource spécifié.

- Si vous choisissez l'une des rubriques MQTT, vous pouvez accéder au client MQTT pour vous abonner à ces rubriques et recevoir des messages.
- Si vous avez ajouté plusieurs événements, vous pouvez vous abonner à plusieurs sujets d'événements et recevoir des notifications les concernant. Pour vous abonner à plusieurs sujets, choisissez vos sujets, puis choisissez Action, puis choisissez S'abonner.

Activez les notifications à l'aide duAWS CLI

Vous pouvez configurer des événements et ajouter des ressources à votre configuration à l'aide de l'AWS IoT WirelessAPI ou duAWS CLI.

Activer les notifications pour toutes les ressources

Vous pouvez activer les notifications pour toutes les ressources de votreCompte AWS compte qui appartiennent à un type de ressource particulier et les surveiller à l'aide de l'[UpdateEventConfigurationByResourceTypes](#)API ou de la commande [update-event-configuration-by-resource-types](#)CLI. Par exemple :

```
aws iotwireless update-event-configuration-by-resource-types \
--cli-input-json input.json
```

Contenu du fichier input.json

```
{
    "DeviceRegistrationState": {
        "Sidewalk": {
            "AmazonIdEventTopic": "Enabled"
        }
    },
    "ConnectionStatus": {
        "LoRaWAN": {
            "WirelessGatewayEventTopic": "Enabled"
        }
    }
}
```

Note

L'échappement de tous les guillemets ("") est effectué avec des barres obliques inverses (\).

Vous pouvez obtenir la configuration actuelle de l'événement en appelant l'[GetEventConfigurationByResourceTypes](#)API ou en utilisant la commande [get-event-configuration-by-resource-types](#)CLI. Par exemple :

```
aws iotwireless get-event-configuration-by-resource-types
```

Activer les notifications pour des ressources individuelles

Pour ajouter des ressources individuelles à la configuration de vos événements et contrôler quels événements sont publiés à l'aide de l'API ou de la CLI, appelez l'[UpdateResourceEventConfiguration](#)API ou utilisez la commande [update-resource-event-configuration](#)CLI. Par exemple :

```
aws iotwireless update-resource-event-configuration \
```

```
--identifier 1ffd32c8-8130-4194-96df-622f072a315f \
--identifier-type WirelessDeviceId \
--cli-input-json input.json
```

Contenu du fichier input.json

```
{
  "Join": {
    "LoRaWAN": {
      "DevEuiEventTopic": "Disabled"
    },
    "WirelessDeviceIdEventTopic": "Enabled"
  }
}
```

Note

L'échappement de tous les guillemets ("") est effectué avec des barres obliques inverses (\).

Vous pouvez obtenir la configuration actuelle de l'événement en appelant l'[GetResourceEventConfiguration](#) API ou en utilisant la commande [get-resource-event-configuration](#) CLI. Par exemple :

```
aws iotwireless get-resource-event-configuration \
--identifier-type WirelessDeviceId \
--identifier 1ffd32c8-8130-4194-96df-622f072a315f
```

Lister les configurations d'événements

Vous pouvez également utiliser l'AWS IoT Wireless API ou le AWS CLI pour répertorier les configurations d'événements dans lesquelles au moins un sujet d'événement a été activé. Pour répertorier les configurations, utilisez l'opération [ListEventConfigurations](#) API ou la commande [list-event-configurations](#) CLI. Par exemple :

```
aws iotwireless list-event-configurations --resource-type WirelessDevice
```

Notifications d'événements pour les ressources LoRa WAN

Vous pouvez utiliser les opérations de AWS IoT Wireless l'API AWS Management Console or pour vous informer des événements concernant vos périphériques et passerelles LoRa WAN. Pour plus d'informations sur les notifications d'événements et sur la manière de les activer, consultez [Notifications d'événements pour AWS IoT Wireless \(p. 1474\)](#) et [Activer les événements pour les ressources sans fil \(p. 1478\)](#).

Types d'événements pour les ressources LoRa WAN

Les événements que vous pouvez activer pour vos ressources LoRa WAN incluent :

- Participez à des événements qui vous informent de la participation à des événements pour votre appareil LoRa WAN. Vous recevez des notifications lorsqu'un appareil se connecte à AWS IoT Core for LoRaWAN, ou lorsqu'une demande de reconnexion de type 0 ou de type 2 est reçue.

- Événements d'état de connexion qui vous avertissent lorsque l'état de connexion de votre passerelle LoRa WAN passe à Connecté ou Déconnecté.

Les sections suivantes contiennent des informations supplémentaires sur les événements relatifs à vos ressources LoRa WAN :

Rubriques

- [LoRaParticiper à des événements WAN \(p. 1482\)](#)
- [Événements d'état de connexion \(p. 1484\)](#)

LoRaParticiper à des événements WAN

AWS IoT Core for LoRaWAN peut publier des messages pour vous informer des événements de participation aux appareils LoRa WAN auxquels vous êtes connecté AWS IoT. Les événements de participation vous avertissent lorsqu'une demande d'adhésion ou de réintégration de type 0 ou de type 2 est reçue et que l'appareil s'y est inscrit AWS IoT Core for LoRaWAN.

Comment fonctionne la participation à des événements

Lorsque vous intégrez vos appareils LoRa WAN à AWS IoT Core for LoRaWAN, AWS IoT Core for LoRaWAN exécute une procédure de jointure pour votre appareil avec AWS IoT Core for LoRaWAN. Votre appareil est alors activé pour être utilisé et peut envoyer un message de liaison montante pour indiquer qu'il est disponible. Une fois l'appareil connecté, des messages de liaison montante et descendante peuvent être échangés entre votre appareil et AWS IoT Core for LoRaWAN. Pour plus d'informations sur l'intégration à votre appareil, consultez [Intégrez vos appareils à AWS IoT Core for LoRaWAN \(p. 1288\)](#).

Vous pouvez activer les événements pour vous avertir lorsque votre appareil se connecte à AWS IoT Core for LoRaWAN. Vous serez également averti si l'événement de participation échoue, lorsqu'une demande de réintégration de type 0 ou de type 2 est reçue et lorsqu'elle est acceptée.

Activer la participation à des événements via le LoRa WAN

Avant que les abonnés au LoRa WAN rejoignent les sujets réservés puissent recevoir des messages, vous devez activer les notifications d'événements pour eux à partir de l'API ou de la AWS Management Console CLI ou à l'aide de l'API ou de la CLI. Vous pouvez activer ces événements pour toutes les ressources LoRa WAN de vos ressources Compte AWS ou pour certaines de vos ressources. Pour plus d'informations sur l'activation de ces événements, consultez [Activer les événements pour les ressources sans fil \(p. 1478\)](#).

Format des rubriques MQTT pour les événements LoRa WAN

Les rubriques MQTT réservées aux périphériques LoRa WAN utilisent le format suivant. Si vous êtes abonné à ces rubriques, tous les appareils LoRa WAN enregistrés sur votre compte Compte AWS peuvent recevoir la notification :

- Rubriques au niveau des ressources

\$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices

- Rubriques d'identification

\$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_devices/{resourceID}/{id}

Où :

```
{eventName}

{eventName} doit être join.

{eventType}

{eventType} peut être :
• join_req_received
• rejoin_req_0_received
• rejoin_req_2_received
• join_accepted

{ID de ressource}

{ResourceID} peut être dev_eui ou wireless_device_id.
```

Par exemple, vous pouvez vous abonner aux rubriques suivantes pour recevoir une notification d'événement lorsque vous avez AWS IoT Core for LoRaWAN accepté une demande d'adhésion provenant de vos appareils.

```
$aws/iotwireless/events/join/join_accepted lorawan/wireless_devices/
wireless_device_id/{id}
```

Vous pouvez également utiliser le caractère+ générique pour vous abonner à plusieurs sujets en même temps. Le caractère+ générique correspond à n'importe quelle chaîne du niveau contenant le caractère, comme dans la rubrique suivante :

```
$aws/iotwireless/events/join/join_req_received lorawan/wireless_devices/
wireless_device_id/+
```

Note

Vous ne pouvez pas utiliser le caractère générique# pour vous abonner aux sujets réservés. Pour plus d'informations sur les filtres de rubrique, consultez [Filtres de rubrique \(p. 116\)](#).

Pour plus d'informations sur l'utilisation du+ caractère générique lorsque vous vous abonnez à des rubriques, consultez [Filtres de rubrique \(p. 116\)](#).

Charge utile des messages pour l'événement de LoRa participation au réseau WAN

Ce qui suit montre la charge utile du message pour l'événement de jointure LoRa WAN.

```
{
// General fields
"eventId": "string",
"eventType": "join_req_received|rejoin_req_0_received|rejoin_req_2_received|
join_accepted",
"WirelessDeviceId": "string",
"timestamp": "timestamp",

// Event-specific fields
"LoRaWAN": {
    "DevEui": "string",

    // The fields below are optional indicating that it can be a null value.
    "DevAddr": "string",
    "JoinEui": "string",
```

```
        "AppEui": "string",  
    }  
}
```

La charge utile contient les attributs suivants :

eventId

Un identifiant d'événement unique généré par AWS IoT Core for LoRaWAN (chaîne).

eventType

Type d'événement qui s'est produit. Il peut avoir l'une des valeurs suivantes :

- `join_req_received`: Ce champ affichera les paramètres de l'EUIJoinEui ou `AppEui`
- `rejoin_req_0_received`
- `rejoin_req_2_received`
- `join_accepted`: Ce champ affichera le `NetId` et `DevAddr`.

wirelessDeviceId

L'ID du périphérique LoRa WAN.

timestamp

Horodatage Unix du moment à laquelle l'événement s'est produit.

DevEui

L'identifiant unique de l'appareil figurant sur l'étiquette de l'appareil ou sur la documentation de l'appareil.

DevAddr et EUI (facultatif)

Ces champs sont l'adresse facultative de l'appareil et les paramètres `EUIJoinEUI` ou `AppEUI`.

Événements d'état de connexion

AWS IoT Core for LoRaWAN peut publier des messages pour vous informer des événements relatifs à l'état de connexion des passerelles LoRa WAN auxquelles vous êtes connecté AWS IoT. Les événements d'état de connexion vous avertissent lorsque l'état de connexion d'une passerelle LoRa WAN passe à Connecté ou Déconnecté.

Fonctionnement des événements relatifs à l'état de connexion

Une fois que vous avez intégré votre passerelle à AWS IoT Core for LoRaWAN, vous pouvez connecter votre passerelle à AWS IoT Core for LoRaWAN et vérifier son état de connexion. Cet événement vous avertit lorsque l'état de connexion de votre passerelle passe à Connecté ou Déconnecté. Pour plus d'informations sur l'intégration et la connexion de votre passerelle à AWS IoT Core for LoRaWAN, consultez [Intégrez vos passerelles vers AWS IoT Core for LoRaWAN \(p. 1279\)](#) et [Connectez votre passerelle LoRa WAN et vérifiez son état de connexion \(p. 1287\)](#).

Format des rubriques MQTT pour les passerelles LoRa WAN

Les rubriques MQTT réservées aux passerelles LoRa WAN utilisent le format suivant. Si vous êtes abonné à ces rubriques, toutes les passerelles LoRa WAN enregistrées sur votre compte AWS peuvent recevoir la notification :

- Pour les rubriques au niveau des ressources :

\$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways

- Pour les rubriques relatives aux identificateurs :

\$aws/iotwireless/events/{eventName}/{eventType}/lorawan/wireless_gateways/{resourceID}/{id}

Où :

{eventName}

{eventName} doit être connection_status.

{eventType}

{eventType} peut être connected ou disconnected.

{ID de ressource}

{ResourceID} peut être gateway_eui ou wireless_gateway_id.

Par exemple, vous pouvez vous abonner aux rubriques suivantes pour recevoir une notification d'événement lorsque toutes vos passerelles sont connectées à AWS IoT Core for LoRaWAN :

\$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/wireless_gateway_id/{id}

Vous pouvez également utiliser le caractère+ générique pour vous abonner à plusieurs sujets en même temps. Le caractère+ générique correspond à n'importe quelle chaîne du niveau contenant le caractère, comme dans la rubrique suivante :

\$aws/iotwireless/events/connection_status/connected/lorawan/wireless_gateways/wireless_gateway_id/+

Note

Vous ne pouvez pas utiliser le caractère générique# pour vous abonner aux sujets réservés. Pour plus d'informations sur les filtres de rubrique, consultez [Filtres de rubrique \(p. 116\)](#).

Pour plus d'informations sur l'utilisation du+ caractère générique lorsque vous vous abonnez à des rubriques, consultez [Filtres de rubrique \(p. 116\)](#).

Charge utile des messages pour les événements relatifs à l'état de la connexion

Ce qui suit montre la charge utile du message pour l'événement d'état de la connexion.

```
{  
    // General fields  
    "eventId": "string",  
    "eventType": "connected|disconnected",  
    "WirelessGatewayId": "string",  
    "timestamp": "timestamp",  
  
    // Event-specific fields  
    "LoRaWAN": {  
        "GatewayEui": "string"  
    }  
}
```

La charge utile contient les attributs suivants :

eventId

Un identifiant d'événement unique généré par AWS IoT Core for LoRaWAN (chaîne).

eventType

Type d'événement qui s'est produit. Peut être `connected` ou `disconnected`.

wirelessGatewayId

L'ID de la passerelle LoRa WAN.

timestamp

Horodatage Unix du moment à laquelle l'événement s'est produit.

GatewayEui

Identifiant unique de la passerelle figurant sur l'étiquette de la passerelle ou sur la documentation de la passerelle.

Notifications d'événements pour les ressources Sidewalk

Vous pouvez utiliser les opérations de AWS IoT Wireless l'API AWS Management Console or pour vous informer des événements concernant vos appareils Sidewalk et vos comptes partenaires. Pour plus d'informations sur les notifications d'événements et sur la manière de les activer, consultez [Notifications d'événements pour AWS IoT Wireless \(p. 1474\)](#) et [Activer les événements pour les ressources sans fil \(p. 1478\)](#).

Types d'événements pour les ressources Sidewalk

Les événements que vous pouvez activer pour vos ressources Sidewalk incluent :

- Événements relatifs à l'appareil qui vous informent des modifications de l'état de votre appareil Sidewalk, par exemple lorsque l'appareil a été enregistré et qu'il est prêt à être utilisé.
- Événements de proximité qui vous avertissent AWS IoT Wireless lorsque vous recevez une notification d'Amazon Sidewalk indiquant qu'une balise a été découverte ou perdue.

Les sections suivantes contiennent plus d'informations sur les événements pour vos ressources Sidewalk :

Rubriques

- [Événements relatifs à l'état d'enregistrement des appareils \(p. 1486\)](#)
- [Événements de proximité \(p. 1489\)](#)
- [Événements d'état de distribution de message \(p. 1491\)](#)

Événements relatifs à l'état d'enregistrement des appareils

Les événements relatifs à l'état d'enregistrement de l'appareil publient des notifications d'événements en cas de modification de l'état d'enregistrement de l'appareil, par exemple lorsqu'un appareil Sidewalk a été

approvisionné ou enregistré. Les événements vous fournissent des informations sur les différents états que traverse l'appareil entre le moment où il est approvisionné et celui où il a été enregistré.

Comment fonctionnent les événements relatifs à l'état d'enregistrement de l'appareil

Lorsque vous intégrez votre appareil Sidewalk à Amazon Sidewalk et AWS IoT Wireless que AWS IoT Wireless vous effectuez une `create` opération et ajoutez votre appareil Sidewalk à votre Compte AWS. Votre appareil entre alors dans l'état provisionné, puis `eventType` devient `provisioned`. Pour plus d'informations sur l'intégration à votre appareil, consultez [Démarrer avec AWS IoT Core pour Amazon Sidewalk \(p. 1404\)](#).

Une fois l'appareil connecté `provisioned`, Amazon Sidewalk effectue une `register` opération pour enregistrer votre appareil Sidewalk auprès de AWS IoT Wireless. Le processus d'enregistrement commence, où le cryptage et les clés de session sont configurés avec AWS IoT. Lorsque l'appareil est enregistré `registered`, `eventType` il devient et votre appareil est prêt à être utilisé.

Une fois l'appareil connecté `registered`, Sidewalk peut envoyer une demande à `register` votre appareil. AWS IoT Wireless répond ensuite à la demande et rétablit l'état de l'appareil à `provisioned`. Pour plus d'informations sur l'état des appareils, consultez [DeviceState](#).

Activer les notifications pour les événements relatifs à l'état d'enregistrement de l'appareil

Avant que les abonnés aux rubriques réservées à l'état d'enregistrement de l'appareil puissent recevoir des messages, vous devez activer les notifications d'événements pour eux à partir de l'API ou de la AWS Management Console CLI ou à l'aide de l'API ou de la CLI. Vous pouvez activer ces événements pour toutes les ressources Sidewalk de vos ressources Compte AWS ou pour certaines d'entre elles. Pour plus d'informations sur l'activation de ces événements, consultez [Activer les événements pour les ressources sans fil \(p. 1478\)](#).

Format des rubriques MQTT pour les événements relatifs à l'état de l'enregistrement des appareils

Pour vous informer des événements relatifs à l'état d'enregistrement de l'appareil, vous pouvez vous abonner aux rubriques réservées du MQTT qui commencent par le signe du dollar (\$). Pour plus d'informations, veuillez consulter [Rubriques MQTT \(p. 115\)](#).

Les rubriques MQTT réservées aux événements d'état de l'enregistrement des appareils Sidewalk utilisent le format suivant :

- Pour les rubriques au niveau des ressources :

\$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices

- Pour les rubriques relatives aux identificateurs :

\$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}

Où :

{eventName}

{eventName} doit être `device_registration_state`.

```
{eventType}  
    {eventType} peut être provisioned ou registered.  
{resourceType}  
    {resourceType} peut être sidewalk_accounts ou wireless_devices.  
{ID de ressource}  
    {ResourceID} correspond à {resourceType} wireless_device_id  
    des sidewalk_accounts et à {resourceType} de wireless_devices.
```

Vous pouvez également utiliser le caractère+ générique pour vous abonner à plusieurs sujets en même temps. Le caractère+ générique correspond à n'importe quelle chaîne du niveau contenant le caractère. Par exemple, si vous souhaitez être informé de tous les types d'événements possibles (provisioned|registered) et pour tous les appareils enregistrés sous un identifiant Amazon particulier, vous pouvez utiliser le filtre thématique suivant :

```
$aws/iotwireless/events/device_registration_state/+/sidewalk/sidewalk_accounts/  
amazon_id/+
```

Note

Vous ne pouvez pas utiliser le caractère générique# pour vous abonner aux sujets réservés. Pour plus d'informations sur les filtres de rubrique, consultez[Filtres de rubrique \(p. 116\)](#).

Charge utile des messages pour les événements relatifs à l'état de l'enregistrement de l'appareil

Une fois que vous avez activé les notifications relatives aux événements relatifs à l'état d'enregistrement des appareils, les notifications d'événements sont publiées via MQTT avec une charge utile JSON. Ces événements contiennent l'exemple de charge utile suivant :

```
{  
    "eventId": "string",  
    "eventType": "provisioned|registered",  
    "WirelessDeviceId": "string",  
    "timestamp": "timestamp",  
  
    // Event-specific fields  
    "operation": "create|deregister|register",  
    "Sidewalk": {  
        "AmazonId": "string",  
        "SidewalkManufacturingSn": "string"  
    }  
}
```

La charge utile contient les attributs suivants :

eventId

Un ID d'événement unique (chaîne).

eventType

Type d'événement qui s'est produit. Peut être provisioned ou registered.

wirelessDeviceId

Identifiant du périphérique sans fil.

timestamp

Horodatage Unix du moment à laquelle l'événement s'est produit.
fonctionnement

L'opération qui a déclenché l'événement. Les valeurs valides sont `create`, `register` et `deregister`.

trottoir

ID Amazon Sidewalk ou `SidewalkManufacturingSn` pour lequel vous souhaitez recevoir des notifications d'événements.

Événements de proximité

Les événements de proximité publient des notifications d'événements lorsqu'ils AWS IoT reçoivent une balise provenant de l'appareil Sidewalk. Lorsque votre appareil Sidewalk approche d'Amazon Sidewalk, les balises envoyées depuis votre appareil sont filtrées par Amazon Sidewalk à intervalles réguliers et reçues par AWS IoT Wireless. AWS IoT Wireless vous informe ensuite de ces événements lorsqu'une balise est reçue.

Comment fonctionnent les événements de proximité

Les événements de proximité vous AWS IoT avertissent lorsque vous recevez une balise. Vos appareils Sidewalk peuvent émettre des balises à tout moment. Lorsque votre appareil se trouve à proximité d'Amazon Sidewalk, Sidewalk reçoit les balises et les transmet à AWS IoT Wireless à intervalles réguliers. Amazon Sidewalk a configuré cet intervalle de temps à 10 minutes. Lorsque AWS IoT Wireless vous recevrez la balise de Sidewalk, vous serez informé de l'événement.

Les événements de proximité vous avertissent lorsqu'une balise est découverte ou lorsqu'une balise est perdue. Vous pouvez configurer les intervalles auxquels vous êtes averti de l'événement de proximité.

Activer les notifications pour les événements de proximité

Avant que les abonnés aux sujets réservés de Sidewalk Proximity puissent recevoir des messages, vous devez activer les notifications d'événements pour eux à partir de AWS Management Console, ou à l'aide de l'API ou de la CLI. Vous pouvez activer ces événements pour toutes les ressources Sidewalk de vos ressources Compte AWS ou pour certaines d'entre elles. Pour plus d'informations sur l'activation de ces événements, consultez [Activer les événements pour les ressources sans fil \(p. 1478\)](#).

Format des sujets MQTT pour les événements de proximité

Pour vous informer des événements de proximité, vous pouvez vous abonner aux sujets réservés du MQTT commençant par le signe du dollar (\$). Pour plus d'informations, veuillez consulter [Rubriques MQTT \(p. 115\)](#).

Les rubriques MQTT réservées aux événements de proximité de Sidewalk utilisent le format suivant :

- Pour les rubriques au niveau des ressources :

`$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices`

- Pour les rubriques relatives aux identificateurs :

`$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}`

Où :

```
{eventName}  
    {eventName} doit être proximity.  
{eventType}  
    {eventType} peut être beacon_discovered ou beacon_lost.  
{resourceType}  
    {resourceType} peut être sidewalk_accounts ou wireless_devices.  
{ID de ressource}  
    {ResourceID} correspond à {resourceType} wireless_device_id  
    des sidewalk_accounts et à {resourceType} wireless_devices.
```

Vous pouvez également utiliser le caractère+ générique pour vous abonner à plusieurs sujets en même temps. Le caractère+ générique correspond à n'importe quelle chaîne du niveau contenant le caractère. Par exemple, si vous souhaitez être informé de tous les types d'événements possibles (beacon_discovered et beacon_lost) et pour tous les appareils enregistrés sous un identifiant Amazon particulier, vous pouvez utiliser le filtre thématique suivant :

```
$aws/iotwireless/events/proximity/+ /sidewalk/sidewalk_accounts/amazon_id/+
```

Note

Vous ne pouvez pas utiliser le caractère générique# pour vous abonner aux sujets réservés. Pour plus d'informations sur les filtres de rubrique, consultez [Filtres de rubrique \(p. 116\)](#).

Charge utile de messages pour les événements de proximité

Une fois que vous avez activé les notifications pour les événements de proximité, les messages d'événements sont publiés via MQTT avec une charge utile JSON. Ces événements contiennent l'exemple de charge utile suivant :

```
{  
    "eventId": "string",  
    "eventType": "beacon_discovered|beacon_lost",  
    "WirelessDeviceId": "string",  
    "timestamp": "1234567890123",  
  
    // Event-specific fields  
    "Sidewalk": {  
        "AmazonId": "string",  
        "SidewalkManufacturingSn": "string"  
    }  
}
```

La charge utile contient les attributs suivants :

eventId

Un identifiant d'événement unique, qui est une chaîne.

eventType

Type d'événement qui s'est produit. Peut être beacon_discovered ou beacon_lost.

wirelessDeviceId

Identifiant du périphérique sans fil.

timestamp

Horodatage Unix du moment à laquelle l'événement s'est produit.

trottoir

ID Amazon Sidewalk ou SidewalkManufacturingSn pour lequel vous souhaitez recevoir des notifications d'événements.

Événements d'état de distribution de message

Les événements d'état de livraison des messages publient des notifications d'événements concernant l'état des messages échangés entre vos appareils Sidewalk et AWS IoT Wireless. Les notifications d'événements sont publiées à la fois pour les messages de liaison descendante AWS IoT Wireless envoyés depuis l'appareil Sidewalk et pour les messages de liaison montante envoyés depuis votre appareil vers AWS IoT Wireless.

Fonctionnement des événements d'état de distribution de message

Une fois que vous avez intégré AWS IoT Wireless et connecté votre appareil Sidewalk à votre appareil, des messages peuvent être échangés entre votre appareil et AWS IoT Wireless. Les événements publient des notifications concernant l'état de remise des messages qui indiquent si ces messages ont été correctement remis à votre appareil ou à AWS IoT Wireless.

Par exemple, si un message de liaison montante est reçu du terminal avec un indicateur d'accusé de réception (ACK), une notification est publiée indiquant que le message a été livré avec succès. Lorsque vous envoyez des messages de liaison descendante depuis AWS IoT Wireless l'appareil Sidewalk, l'SendDataToWirelessDevice API renvoie un messageMessageId de liaison descendante même si des paquets ont été perdus ou si le message n'a pas été livré. Dans ce cas, les événements d'état de remise du message renvoient une erreur indiquant que le message n'a pas pu être remis au terminal.

Activer les notifications pour les événements relatifs à l'état de livraison des messages

Avant que les abonnés aux sujets réservés sur le statut de livraison des messages Sidewalk puissent recevoir des messages, vous devez activer les notifications d'événements pour eux à l'aide de l'AWS IoT Wireless API ou du AWS CLI. Vous pouvez activer ces événements pour toutes les ressources Sidewalk de vos ressources Compte AWS ou pour certaines d'entre elles.

Note

La configuration de l'événement d'état de distribution de message Sidewalk n'est pas disponible dans la console.

Pour plus d'informations sur l'activation de ces événements, consultez [Activez les notifications à l'aide du AWS CLI \(p. 1480\)](#).

Format des rubriques MQTT pour les événements relatifs à l'état de livraison des messages

Pour recevoir des notifications concernant les événements relatifs à l'état de livraison des messages, vous pouvez vous abonner aux rubriques réservées du MQTT qui commencent par le signe du dollar (\$). Pour plus d'informations, veuillez consulter [Rubriques MQT \(p. 115\)](#).

Les rubriques MQTT réservées aux événements de proximité de Sidewalk utilisent le format suivant :

- Pour les rubriques au niveau des ressources :

\$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/wireless_devices

- Pour les rubriques relatives aux identificateurs :

\$aws/iotwireless/events/{eventName}/{eventType}/sidewalk/{resourceType}/{resourceID}/{id}

Où :

{eventName}

{eventName} doit être message_delivery_status.

{eventType}

{eventType} peut être success ou error.

{resourceType}

{resourceType} peut être sidewalk_accounts ou wireless_devices.

{ID de ressource}

{ResourceID} correspond à amazon_id à {resourceType} wireless_device_id des sidewalk_accounts et à {resourceType} de wireless_devices.

Vous pouvez également utiliser le caractère+ générique pour vous abonner à plusieurs sujets en même temps. Le caractère+ générique correspond à n'importe quelle chaîne du niveau contenant le caractère. Par exemple, si vous souhaitez être informé de tous les types d'événements possibles (success,error) et pour tous les appareils enregistrés sous un identifiant Amazon particulier, vous pouvez utiliser le filtre thématique suivant :

\$aws/iotwireless/events/message_delivery_status/+(sidewalk/sidewalk_accounts/amazon_id/+)

Note

Vous ne pouvez pas utiliser le caractère générique# pour vous abonner aux sujets réservés. Pour plus d'informations sur les filtres de rubrique, consultez [Filtres de rubrique \(p. 116\)](#).

Charge utile des messages pour les événements relatifs à l'état de livraison des messages

Une fois que vous avez activé les notifications pour les événements d'état de remise des messages, les messages d'événements sont publiés via MQTT avec une charge utile JSON. Ces événements contiennent l'exemple de charge utile suivant, selon qu'il s'agit d'un événement réussi, indiquant que le périphérique a bien reçu le message, ou d'une erreur.

Événements de réussite

Ce qui suit montre le format de la charge utile lorsque l'événement est réussi.

```
{  
    "eventId": "string",  
    "eventType": "success",  
    "WirelessDeviceId": "string",  
    "timestamp": "timestamp",  
    "Sidewalk": {  
        "Seq": "Integer",  
    }  
}
```

```
        "MsgType": "CUSTOM_COMMAND_ID_RESP",
        "CmdExStatus": "COMMAND_EXEC_STATUS_SUCCESS"
    }
```

La charge utile contient les attributs suivants :

eventId

Un identifiant d'événement unique, qui est une chaîne.

eventType

Type d'événement qui s'est produit. Peut être `success` ou `error`. Dans ce cas, le champ `eventType` est `error`.

WirelessDeviceId

Identifiant du périphérique sans fil.

timestamp

Horodatage Unix du moment à laquelle l'événement s'est produit.

trottoir

Le wrapper Sidewalk qui contient le code d'état des messages de réussite, le numéro de séquence du message et le type de message.

Événements d'erreur

Ce qui suit montre le format de la charge utile lorsque l'événement indique qu'une erreur s'est produite.

```
{
    "eventId": "string",
    "eventType": "error" ,
    "WirelessDeviceId": "string",
    "timestamp": "timestamp",
    "Sidewalk": {
        "Seq": "Integer",
        "Status": "DeviceNotReachable" | "RADIO_TX_ERROR" | "MEMORY_ERROR"
    }
}
```

La charge utile contient des attributs similaires à ceux que `eventType` se présentent sous la forme d'`unsuccess`. Voici quelques différences ou attributs supplémentaires :

eventType

Type d'événement qui s'est produit. Dans ce cas, il s'agit d'un `error`.

trottoir

Le wrapper Sidewalk qui contient le numéro de séquence et le code d'état indiquant pourquoi le message de liaison descendante n'a pas été envoyé correctement.

AWS IoTSDK pour appareils, kits SDK mobiles et client deAWS IoT l'appareil

Cette page résume les SDK pourAWS IoT appareils, les bibliothèques open source, les guides du développeur, les exemples d'applications et les guides de portage pour vous aider à créer des solutions IoT innovantes avecAWS IoT les plateformes matérielles de votre choix.

Ces SDK sont destinés à être utilisés sur votre appareil IoT. Si vous développez une application IoT destinée à être utilisée sur un appareil mobile, consultez le[Kits SDK AWS Mobile \(p. 1496\)](#). Si vous développez une application IoT ou un programme côté serveur, consultez le[Kits de développement logiciel \(SDK\) AWS \(p. 80\)](#).

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 SDK vous aident à connecter vos appareils IoT àAWS IoT l'aide des protocoles MQTT et WSS.

C++

AWS IoTSDK pour appareils C++

Le SDKAWS IoT C++ Device permet aux développeurs de créer des applications connectées à l'aideAWS desAWS IoT API. 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 en savoir plus, consultez les ressources suivantes :

- [AWS IoTDevice SDK C++ v2 activé GitHub](#)
- [AWS IoTDevice SDK C++ v2 Readme](#)
- [AWS IoTExemples de SDK C++ v2 pour appareils](#)
- [AWS IoTDocumentation de l'API C++ v2 du SDK de l'appareil](#)

Python

AWS IoTSDK de périphérique pour Python

LeAWS IoT Device SDK pour Python permet aux développeurs d'écrire des scripts Python afin d'utiliser leurs appareils pour accéder à laAWS IoT plate-forme via MQTT ou MQTT via le WebSocket protocole. En connectant leurs appareils àAWS IoT, les utilisateurs peuvent travailler en toute sécurité avec le courtier de messages, les règlesAWS IoT et les ombres fournis par et avec d'autresAWS services tels queAWS Lambda Kinesis et Amazon S3, etc.

- [AWS IoTSDK de l'appareil pour Python v2 sur GitHub](#)
- [Kit SDK des appareils AWS IoT pour Python v2 - Readme](#)

- [AWS IoTExemples de SDK de périphérique pour Python v2](#)
- [AWS IoTDocumentation du SDK de l'appareil pour Python v2](#)

JavaScript

AWS IoTSDK de l'appareil pour JavaScript

Le package aws-iot-device-sdk.js permet aux développeurs d'écrire des JavaScript applications qui accèdent à AWS IoT via MQTT ou MQTT via le WebSocket protocole. Il peut être utilisé dans des environnements Node.js et des applications de navigateur. Pour en savoir plus, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil pour JavaScript v2 sur GitHub](#)
- [AWS IoTSDK de l'appareil pour JavaScript v2 Readme](#)
- [AWS IoTSDK de périphérique pour les échantillons JavaScript v2](#)
- [AWS IoTDocumentation du SDK de l'appareil pour l'API JavaScript v2](#)

Java

AWS IoTDevice SDK for Java

Le AWS IoT Device SDK for Java permet aux développeurs Java d'accéder à la AWS IoT plateforme via MQTT ou MQTT via le WebSocket protocole. 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 déserialiser des documents JSON. Pour en savoir plus, consultez les ressources suivantes :

- [AWS IoTLe SDK de l'appareil pour Java v2 est activé GitHub](#)
- [AWS IoT Kit SDK des périphériques pour Java v2 – Readme](#)
- [AWS IoTExemples de kit SDK for Java v2](#)
- [AWS IoTDocumentation sur l'API Java v2 :](#)

Kit SDK des appareils AWS IoT pour Embedded C

Note

Ce SDK est destiné à être utilisé par des développeurs de logiciels embarqués expérimentés.

Le Kit SDK des appareils AWS IoT pour Embedded C (C-SDK) est une collection de fichiers sources C sous la licence open source du MIT qui peuvent être utilisés dans des applications intégrées pour connecter en toute sécurité des appareils IoT à AWS IoT Core. Il inclut un client MQTT, un analyseur JSON et AWS IoT Device Shadow, des AWS IoT tâches, un provisionnement de AWS IoT flotte et AWS IoT Device Defender des bibliothèques. Ce SDK est distribué sous forme de code source et peut être intégré au microprogramme du client avec le code de l'application, d'autres bibliothèques et le système d'exploitation (OS) de votre choix.

Le Kit SDK des appareils AWS IoT pour 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 en savoir plus, consultez les ressources suivantes :

- [AWS IoTSDK de l'appareil pour C embarqué sur GitHub](#)
- [Kit SDK des appareils AWS IoT pour Embedded C – Readme](#)
- [AWS IoTSDK de périphérique pour les échantillons C intégrés](#)

Versions antérieures des SDK pourAWS IoT appareils

Il s'agit de versions antérieures des SDK pourAWS IoT appareils qui ont été remplacées par les versions les plus récentes répertoriées ci-dessus. Ces SDK reçoivent uniquement des mises à jour de maintenance et de sécurité. Ils ne seront pas mis à jour pour inclure de nouvelles fonctionnalités et ne doivent pas être utilisés sur de nouveaux projets.

- [AWS IoTSDK de périphérique C++ activé GitHub](#)
- [Kit SDK des appareils C++ AWS IoT](#)
- [AWS IoTSDK de l'appareil pour Python v1 sur GitHub](#)
- [Kit SDK des appareils AWS IoT pour Python v1 - Readme](#)
- [AWS IoTSDK de l'appareil pour Java sur GitHub](#)
- [Kit SDK des périphériques AWS IoT pour Java Readme](#)
- [AWS IoTSDK de l'appareil pour JavaScript un GitHub](#)
- [AWS IoTSDK de l'appareil pour JavaScript Readme](#)
- [Kit SDK Arduino Yún activé GitHub](#)
- [Kit SDK Arduino Yún – Readme](#)

Kits SDK AWS Mobile

Les SDKAWS mobiles fournissent aux développeurs d'applications mobiles un support spécifique à la plate-forme pour les API desAWS IoT services, la communication entre appareils IoT à l'aide de MQTT et les API d'autresAWS services.

Android

AWS Mobile SDK for Android

LeAWS Mobile SDK for Android contient une bibliothèque, des exemples et de la documentation permettant aux développeurs de créer des applications mobiles connectées à l'aide deAWS. Ce SDK inclut également la prise en charge des communications entre appareils MQTT et l'appel des API desAWS IoT services. Pour en savoir plus, consultez les ressources suivantes :

- [AWS Mobile SDK for Androidsur GitHub](#)
- [AWS Mobile SDK for AndroidLisez-moi](#)
- [Exemples AWS Mobile SDK for Android](#)
- [Référence d'API AWS Mobile SDK for Android](#)
- [AWSIoTClient Documentation de référence pour les classes](#)

iOS

AWS Mobile SDK for iOS

AWS Mobile SDK for iOSII s'agit d'un kit de développement logiciel open source, distribué sous une licence Apache Open Source. AWS Mobile SDK for iOSfournit une bibliothèque, des exemples de code et de la documentation pour aider les développeurs à créer des applications mobiles connectées

à l'aide deAWS. Ce SDK inclut également la prise en charge des communications entre appareils MQTT et l'appel des API desAWS IoT Core services. Pour en savoir plus, consultez les ressources suivantes :

- [AWS Mobile SDK for iOS sur GitHub](#)
- [AWS Mobile SDK for iOS Lisez-moi](#)
- [Exemples AWS Mobile SDK for iOS](#)
- [AWSIoT Documents de référence pour les cours dans leAWS Mobile SDK for iOS](#)

AWS IoTAppareil client

LeAWS IoT Device Client fournit un code qui permet à votre appareil de se connecter à votre appareilAWS IoT, d'effectuer des tâches de provisionnement de flotte, de respecter les politiques de sécurité des appareils, de se connecter à l'aide d'un tunnel sécurisé et de traiter des tâches sur votre appareil. Vous pouvez installer ce logiciel sur votre appareil pour gérer ces tâches de routine afin de vous concentrer sur votre solution spécifique.

Note

LeAWS IoT Device Client fonctionne avec des appareils IoT à microprocesseurs dotés de processeurs x86_64 ou ARM et de systèmes d'exploitation Linux courants.

C++

AWS IoTAppareil client

Pour de plus amples informations sur leAWS IoT Device Client en C++, veuillez consulter :

- [AWS IoTDevice Client en code source C++ sur GitHub](#)
- [AWS IoTDevice Client en C++ Readme](#)

Résolution des problèmes de AWS IoT

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Les informations suivantes peuvent vous aider à résoudre les problèmes courants dans AWS IoT.

Tâches

- [Diagnostic des problèmes de connectivité \(p. 1498\)](#)
- [Diagnostic des problèmes de règles \(p. 1501\)](#)
- [Diagnostic des problèmes de shadows \(p. 1502\)](#)
- [Diagnostic des problèmes liés aux actions de flux d'entrée Salesforce IoT \(p. 1503\)](#)
- [Guide de dépannage de flotte \(p. 1504\)](#)
- [Dépannage « Limite de flux dépassée pourAWScompte » \(p. 1506\)](#)
- [Guide de dépannage AWS IoT Device Defender \(p. 1506\)](#)
- [AWS IoTGuide de dépannage Device Advisor \(p. 1510\)](#)
- [Résolution des déconnexions de flotte d'appareils \(p. 1512\)](#)
- [Erreurs AWS IoT \(p. 1512\)](#)

Diagnostic des problèmes de connectivité

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Une connexion réussie àAWS IoT nécessite :

- Une connexion valide
- Un certificat valide et actif
- Une politique qui autorise la connexion et le fonctionnement souhaités

Connexion

Comment puis-je trouver le bon point de terminaison ?

- 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 endpoint Address renvoyé par le [describe-endpoint](#) ou [describe-domain-configuration](#) commandes. Il s'agit de la même adresse que le point de terminaison à l'étape précédente.

Comment résoudre un problème de connectivité qui persiste ?

Vous pouvez utiliser AWS Device Advisor pour vous aider à résoudre les problèmes. Les tests préédéfinis de Device Advisor vous aident à valider le logiciel de votre appareil par rapport aux meilleures pratiques d'utilisation de [TLS](#), [MQTT](#), [AWS IoT Device Shadow](#), et [AWS IoT Tâches](#).

Voici un lien vers le [Device Advisor](#) contenu.

Authentification

Les appareils doivent être [authentifié](#) (p. 320) pour connecter à AWS IoT Points de terminaison . Pour les appareils qui utilisent [Certificats client X.509](#) (p. 320) pour l'authentification, les certificats doivent être enregistrés auprès de AWS IoT et soyez 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 sur [Author du serveur dans AWS IoT Core](#) puis suivez les liens pour télécharger le certificat CA approprié.

Ce qui est vérifié lorsqu'un appareil se connecte à AWS IoT?

Lorsqu'un appareil tente de se connecter à AWS IoT :

1. AWS IoT vérifie la validité d'un certificat et d'une valeur SNI (Server Name Indication).
2. AWS IoT vérifie que le certificat utilisé est enregistré auprès du AWS IoT Compte et qu'il a été activé.
3. Lorsqu'un appareil tente d'effectuer une action dans AWS IoT, par exemple pour s'abonner ou publier un message, la stratégie attachée au certificat qu'il a 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 puis-je vérifier le statut d'un certificat ?

- Lister les certificats

Si vous ne connaissez pas l'ID du certificat, vous pouvez consulter l'état de tous vos certificats en utilisant la commande [aws 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"
```

- Consultez le certificat dans le AWS IoT Console

Dans [AWS IoT console](#), dans le menu de gauche, choisissez Secure, puis Certificats.

Choisissez le certificat que vous utilisez pour vous connecter dans la liste pour ouvrir sa page de détails.

Sur la page détaillée du certificat, vous pouvez voir son état actuel.

Le statut du certificat peut être modifié en utilisant les Actions dans le coin supérieur droit de la page de détails.

Autorisation

AWS IoT utilise des ressources [Stratégies AWS IoT Core \(p. 357\)](#) pour autoriser ces ressources à effectuer [actions \(p. 358\)](#). Pour qu'une action soit autorisée, les ressources doivent être associées à un document de stratégie qui accorde l'autorisation d'effectuer 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 les actions [\(p. 358\)](#) que vous essayez de jouer.

Assurez-vous que la stratégie ci-jointe autorise les ressources [\(p. 360\)](#) qui tentent d'effectuer les actions autorisées.

J'ai une AUTHORIZATION_FAILURE entrée dans 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 les actions [\(p. 358\)](#) que vous essayez de jouer.

Assurez-vous que la stratégie ci-jointe autorise les ressources [\(p. 360\)](#) qui tentent d'effectuer les actions autorisées.

Comment puis-je vérifier ce que la politique autorise ?

Dans [AWS IoT console](#), dans le menu de gauche, choisissez Secure, puis Certificats.

Choisissez le certificat que vous utilisez pour vous connecter dans la liste pour ouvrir sa page de détails.

Sur la page détaillée du certificat, vous pouvez voir son état actuel.

Dans le menu de gauche de la page détaillée du certificat, choisissez Stratégies pour voir les stratégies attachées au certificat.

Choisissez la stratégie souhaitée pour voir sa page de détails.

Sur la page de détails de la politique, consultez la Document de stratégie pour voir ce qu'il autorise.

Choisissez Modifier le document de stratégie pour apporter des modifications au document de stratégie.

Sécurité et identité

Lorsque vous fournissez les certificats de serveur pour AWS IoT configuration de domaine personnalisée, les certificats ont un maximum de quatre noms de domaine.

Pour de plus amples informations, veuillez consulter [Points de terminaison et quotas AWS IoT Core](#).

Diagnostic des problèmes de règles

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Cette section décrit certains des éléments à vérifier lorsque vous rencontrez un problème avec une règle.

Configuration CloudWatch Journaux pour dépannage

La meilleure façon de résoudre les problèmes que vous rencontrez avec les règles consiste à utiliser CloudWatch Bûches. Lorsque vous activez CloudWatch Journaux pour AWS 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 [Surveiller AWS IoT à l'aide CloudWatch des journaux \(p. 490\)](#).

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 AssumeRole sur la ressource. Voici un exemple de journal généré par [la journalisation affinée \(p. 472\)](#) :

```
{  
    "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 \(p. 470\)](#) :

```
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 des journaux AWS IoT dans la console CloudWatch” \(p. 490\)](#).

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 attendues :

- Consultez les journaux pour détecter les messages d'erreur.
- Vérifiez que votre syntaxe SQL correspond au document JSON contenu dans le message.

Passez en revue les noms d'objet et de propriété utilisés dans la requête avec ceux utilisés dans le document JSON de la charge utile du message de la rubrique. Pour plus d'informations sur la mise en forme JSON dans les requêtes SQL, consultez [Extensions JSON \(p. 680\)](#).

- Vérifiez si les noms d'objets ou de propriétés JSON incluent des caractères réservés ou numériques.

Pour plus d'informations sur les caractères réservés dans les références d'objets JSON dans les requêtes SQL, consultez [Extensions JSON \(p. 680\)](#).

Diagnostic des problèmes de shadows

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Diagnostic de shadows

Problème	Consignes pour la résolution des problèmes
Un document shadow d'appareil est rejeté avec <code>Invalid JSON document</code> .	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. 729) .
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 <code>desired</code> et <code>reported</code> 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

Diagnostic des problèmes liés aux actions de flux d'entrée Salesforce IoT

Aidez-nous à améliorer ce sujet

Dites-nous ce qui pourrait l'améliorer

Trace d'exécution

Comment consulter la trace d'exécution d'une action Salesforce ?

Consultez la section [Surveiller AWS IoT à l'aide CloudWatch des journaux \(p. 490\)](#). 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 CloudWatch Bûches. 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 CloudWatch Bûches. 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, contactez AWS IoT Support.

Guide de dépannage de flotte

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Dépannage des requêtes d'agrégation pour le service d'indexation de parc

Si vous rencontrez des erreurs de non-correspondance de type, vous pouvez utiliser CloudWatch Journaux pour résoudre le problème. CloudWatch Les journaux doivent être activés avant que les journaux ne soient écrits par le service d'indexation de flotte. Pour plus d'informations, consultez [Surveiller AWS IoT à l'aide CloudWatch des journaux \(p. 490\)](#).

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 à CloudWatch Consigne quand 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 plus longtemps que time-to-live (TTL). 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 métriques de flotte

Impossible de créer une métrique de flotte

La rétrogradation des sources de données en mettant à jour la configuration d'indexation de flotte n'est pas prise

Si vous essayez de créer une métrique de flotte avec des sources de données rétrogradées (par exemple, auparavant, les sources de données étaient des données de registre, des données parallèles et des données de connectivité des appareils, et maintenant les sources de données sont des données de registre et des données fantômes et sans données de connectivité des appareils), vous verrez des erreurs et vous ne pourrez pas créer une métrique de flotte.

La modification des champs personnalisés utilisés par les métriques de flotte existantes n'est pas prise en charge.

Impossible de voir les points de données dans CloudWatch

Si vous êtes en mesure de créer une métrique de flotte mais que vous ne pouvez pas voir les points de données dans CloudWatch, il est probable que vous n'ayez rien qui réponde aux critères de chaîne de requête.

Consultez cet exemple de commande pour la création d'une métrique de parc :

```
aws iot create-fleet-metric --metric-name "example_FM" --query-string  
"thingName:TempSensor* AND attributes.temperature>80" --period 60 --aggregation-field  
"attributes.temperature" --aggregation-type name=Statistics,values=count
```

Si aucun élément ne répond aux critères de chaîne de requête--query-string
"thingName:TempSensor* AND attributes.temperature>80" :

- avecvalues=count, vous serez en mesure de créer une métrique de flotte et des points de données seront affichés dans CloudWatch. Les points de données de la valeurcountest toujours 0.
- avecvaluesautres quecount, vous serez en mesure de créer une métrique de flotte, mais vous ne verrez pas la métrique de flotte dans CloudWatch et il n'y aura aucun point de données à afficher dans CloudWatch.

Dépannage « Limite de flux dépassée pourAWScompte »

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

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 la commandeAWS CLIou kit SDK :

```
aws iot delete-stream -stream-id value
```

Pour plus de détails, consultez [delete-stream](#).

Note

Vous pouvez utiliser le pluginlist-streamspour rechercher les ID de flux.

Guide de dépannage AWS IoT Device Defender

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Général

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 surAWS IoTappareils connectés ou passerelles d'appareils. Les appareils doivent fournir un identifiant client ou un nom d'objet cohérent.

Audit

Q : J'ai activé un contrôle qui 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 (par exemple, des certificats, des objets ou des

stratégies), les résultats du contrôle peuvent nécessiter un certain temps après que vous avez activé celui-ci.

Détection

Q : Comment puis-je connaître les seuils à définir dans unAWS IoT Device Defendercomportement de 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 la manière dont 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 future 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 : J'envoie 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 par l'abonnement aux rubriques MQTT suivantes :

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected
```

\$aws/things/THING_NAME/defender/metrics/FORMAT/accepted

THING_NAMEest le nom de l'objet signalant la métrique etFORMATest « 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 `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 dansAWS 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 alors que 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 \(p. 1120\)](#). Vérifiez que les appareils non enregistrés utilisent un identifiant de client ou un nom d'objet cohérent. Si le nom d'objet contient des caractères de contrôle ou est composé de plus de 128 octets de caractères codés UTF-8, les messages notifiés par les appareils sont rejettés.

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 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: Voir [AWS 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 permet AWS IoT Device Defender pour 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. Par exemple :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "sns:Publish",  
            "Resource": "<sns-topic-arn>"  
        }  
    ]  
}
```

- Si la rubrique SNS utilisée pour la publication des alertes est une rubrique cryptée, avec l'autorisation de publier sur la rubrique SNS, AWS IoT doit se voir accorder deux autres autorisations. Par exemple :

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sns:Publish",  
                "kms:Decrypt",  
                "kms:GenerateDataKey"  
            ],  
            "Resource": "<sns-topic-arn>"  
        }  
    ]  
}
```

Q : Soumission de mon rapport de mesures avec un type de mesure personnalisé number échoue avec le message d'erreur Malformed metrics report. Que se passe-t-il ?

A: Le type `number` ne prend qu'une seule valeur de mesure en entrée, mais lors de la soumission de la valeur des métriques dans le champ `DeviceMetrics`, il doit être transmis sous forme de tableau avec une seule valeur. Assurez-vous que vous soumettez la valeur de la métrique sous forme de tableau.

Charge utile de l'erreur :

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}),"custom_metrics":{"my_custom_metric":{"number":0}}}}
```

Message d'erreur:

```
{"thingName":"myThing","status":"REJECTED","statusDetails":{"ErrorCode":"InvalidPayload","ErrorMessage":"Malformed metrics report"},"timestamp":1635802047699}
```

Charge utile sans erreur :

```
{"header":{"report_id":12334567,"version":"1.0"},"metrics":{"network_stats":{"bytes_in":30680,"bytes_out":10652,"packets_in":113,"packets_out":118}),"custom_metrics":{"my_custom_metric":[{"number":0}]}}}
```

Réponse :

```
{"thingName":"myThing","12334567":1635800375,"status":"ACCEPTED","timestamp":1635801636023}
```

AWS IoTGuide de dépannage Device Advisor

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

Général

Q : Puis-je exécuter plusieurs suites de tests en parallèle ?

A: Oui. Device Advisor prend désormais en charge l'exécution de plusieurs suites de tests sur différents appareils utilisant un point de terminaison au niveau de l'appareil. Si vous utilisez le point de terminaison au niveau du compte, vous pouvez exécuter une suite à la fois, car un point de terminaison Device Advisor est disponible par compte. Pour plus d'informations, consultez la section [.Configurer votre appareil](#).

Q : J'ai vu sur mon appareil que la connexion TLS avait été refusée par Device Advisor. Est-ce que c'est attendu ?

A: Oui. Device Advisor refuse la connexion TLS avant et après chaque test. Nous recommandons aux utilisateurs de mettre en œuvre un mécanisme de nouvelle tentative d'appareil pour bénéficier d'une expérience de test entièrement automatisée avec Device Advisor. Si vous exécutez une suite de tests avec plus d'un scénario de test, par exemple TLS connect, MQTT connect et MQTT publish, nous vous recommandons de créer un mécanisme pour votre appareil. Le mécanisme peut essayer de se connecter à notre point de terminaison de test toutes les 5 secondes pendant une minute à deux. De cette manière, vous pouvez exécuter plusieurs scénarios 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 le AWS IoT Management Console et filtrez la source d'événement pour qu'elle soit `iotdeviceadvisor.amazonaws.com`.

Q : Comment puis-je voir les journaux de Device Advisor CloudWatch?

A: Les journaux générés lors de l'exécution d'une suite de tests sont téléchargés sur CloudWatch si vous ajoutez la stratégie requise (par exemple, CloudWatchFullAccess) à votre rôle de service (voir [Configuration \(p. 1168\)](#)). S'il existe au moins un scénario de test dans la suite de tests, un groupe de journaux « aws/iot/deviceadvisor/\$testSuiteId » est créé avec deux flux de journaux. L'un des flux est le « \$testRunId » et inclut des journaux des actions entreprises avant et après l'exécution des scénarios de test dans votre suite de tests, telles que les étapes de configuration et de nettoyage. L'autre flux de journaux est « \$suiteRunId-\$testRunId », qui est spécifique à l'exécution d'une suite de tests. Événements envoyés depuis des appareils et AWS IoT Core sera connecté à ce flux de journaux.

Q : Quel est l'objectif du rôle d'autorisation de l'appareil ?

A: Device Advisor se situe entre votre appareil de test et AWS IoT Core pour simuler des scénarios de test. Il accepte les connexions et les messages de vos appareils de test et les transmet à AWS IoT Core en assumant le rôle d'autorisation de votre appareil et en établissant une connexion en votre nom. Il est important de s'assurer que les autorisations de rôle d'appareil sont les mêmes que celles du certificat que vous utilisez pour exécuter les tests. AWS IoT Les stratégies de certificat ne sont pas appliquées lorsque Device Advisor initie une connexion à AWS IoT Core en votre nom en utilisant le rôle d'autorisation de l'appareil. Toutefois, les autorisations du rôle d'autorisation de l'appareil 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 : Pourquoi est-ce que je vois des résultats incohérents ?

A: L'une des principales causes des résultats incohérents est la mise en place d'un `testEXECUTION_TIMEOUT` à une valeur trop faible. Pour plus d'informations sur les recommandations et par défaut `EXECUTION_TIMEOUT` valeurs, consultez [Cas de test Device Advisor](#).

Q : Quel protocole MQTT est pris en charge par Device Advisor ?

A: Device Advisor prend en charge MQTT version 3.1.1 avec les certificats clients X509.

Q : Que se passe-t-il si mon scénario de test échoue avec un message d'expiration du délai d'exécution alors que j'essaie de connecter mon appareil au point de terminaison du test ?

A: Validez toutes les étapes de [Créez un rôle IAM à utiliser comme rôle d'appareil](#). Si le test échoue toujours, il se peut que l'appareil n'envoie pas l'extension SNI (Server Name Indication) correcte, ce qui est nécessaire au bon fonctionnement de Device Advisor. La valeur SNI correcte est l'adresse du point de terminaison renvoyée lorsque vous suivez la [Section Configurer votre appareil](#). AWS IoT Exige également que les appareils envoient l'extension SNI (Server Name Indication) au protocole TLS (Transport Layer Security). Pour de plus amples informations, veuillez consulter [Sécurité du transport dans AWS IoT](#).

Q : Ma connexion MQTT échoue avec un message »libaws-c-mqtt :

`AWS_ERROR_MQTT_UNEXPECTED_HANGUP` » erreur (ou) la connexion MQTT de mon appareil est automatiquement déconnectée du point de terminaison Device Advisor. Comment résoudre cette erreur ?

A: Ce code d'erreur particulier et les déconnexions inattendues peuvent être causés par de nombreux facteurs différents, mais sont probablement liés au [rôle de l'appareil](#) fixé à l'appareil. Les points de contrôle ci-dessous (par ordre de priorité) résoudront ce problème.

- Le rôle d'appareil associé à l'appareil doit disposer des autorisations IAM minimales requises pour exécuter les tests. Device Advisor utilisera le rôle d'appareil associé pour effectuer AWS IoT Actions MQTT pour le compte de l'appareil de test. Si les autorisations requises sont absentes, le `AWS_ERROR_MQTT_UNEXPECTED_HANGUP` erreur s'affichera ou des déconnexions inattendues se produiront lorsque l'appareil tentera de se connecter au point de terminaison Device Advisor. Par exemple, si vous avez choisi d'exécuter la publication MQTT, les actions Connect et Publish doivent être incluses dans le rôle avec la valeur correspondante ClientId et Topic (vous pouvez fournir plusieurs valeurs en utilisant des virgules pour séparer les valeurs, et vous pouvez fournir

des valeurs de préfixe à l'aide d'un caractère générique (*). Par exemple : Pour fournir des autorisations de publication sur n'importe quel sujet commençant par `TestTopic`, vous pouvez fournir »`TestTopic*`» comme valeur de la ressource. Voici quelques [exemples de stratégies](#).

- Incompatibilité entre les valeurs définies dans le rôle d'appareil pour vos types de ressources et les valeurs réelles utilisées dans le code. Par exemple : Une non-correspondance dans ClientId défini dans le rôle et le ClientId utilisé dans le code de votre appareil. Des valeurs comme ClientId, Topic et TopicFilter doit être identique dans le rôle et le code de l'appareil.
- Le certificat d'appareil attaché à votre appareil doit être actif et avoir une [politique](#) attaché à celui-ci avec le [autorisations d'action pour ressources](#). Notez que la stratégie de certificat d'appareil accorde ou refuse l'accès à AWS IoT Resources et AWS IoT Core opérations de plan de données. Device Advisor exige que vous ayez un certificat d'appareil actif attaché à votre appareil qui accorde les autorisations d'action utilisées lors d'un scénario de test.

Résolution des déconnexions de flotte d'appareils

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

AWS IoT les déconnexions du parc d'appareils 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 de AWS IoT service ou limite de limitation.

Pour diagnostiquer la raison de la déconnexion

Vous pouvez consulter la [AWS IoT Logs V2](#) groupe de journaux dans [CloudWatch](#) pour identifier la raison de la déconnexion dans `ledisconnectReason` de l'entrée de journal.

Vous pouvez également utiliser AWS IoT de [événements du cycle de vie](#) pour identifier la raison de la déconnexion. Si vous êtes abonné à [événement déconnexion du cycle de vie](#) (`$aws/events/presence/disconnected/`), vous recevez une notification de AWS IoT lorsque la déconnexion se produit. Vous pouvez identifier la raison de la déconnexion dans `ledisconnectReason` champ de la notification.

Pour de plus amples informations, veuillez consulter [CloudWatch AWS IoT Entrées du journal](#) et [Événements du cycle de vie](#).

Pour résoudre les problèmes de déconnexion dus à AWS IoT maintenance de service

Déconnexions causées par AWS IoT sont enregistrées en tant que `SERVER_INITIATED_DISCONNECT` dans AWS IoT de l'événement du cycle de vie CloudWatch. Pour gérer ces déconnexions, ajustez votre configuration côté client pour vous assurer que vos appareils peuvent être automatiquement reconnectés au AWS IoT platform.

Pour résoudre les problèmes de déconnexion dus à une limite de limitation

Les déconnexions causées par une limite de limitation sont enregistrées comme `THROTTLED` dans AWS IoT de l'événement du cycle de vie CloudWatch. Pour gérer ces déconnexions, vous pouvez demander [augmentation de la limite de message broker](#) à mesure que le nombre d'appareils augmente.

Pour de plus amples informations, veuillez consulter [AWS IoT Agent de messages de base](#).

Erreurs AWS IoT

Aidez-nous à améliorer ce sujet

[Dites-nous ce qui pourrait l'améliorer](#)

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.
500	Erreur interne.
503	Service non disponible.

AWS IoTQuotas

Vous pouvez trouver des informations sur AWS IoTQuotas dans AWS Référence générale.

- Pour AWS IoT Core informations sur les quotas, voir [AWS IoT Core Points de terminaison et quotas](#).
- Pour AWS IoT Device Management informations sur les quotas, voir [AWS IoT Device Management Points de terminaison et quotas](#).
- Pour AWS IoT Device Defender informations sur les quotas, voir [AWS IoT Device Defender Points de terminaison et quotas](#).

Tarification d'AWS IoT Core

Vous trouverez des informations sur la AWS IoT Core tarification sur la page AWS Marketing et dans le [calculateur de AWS prix](#).

- Pour vérifier les informations AWS IoT Core de tarification, consultez la section [AWS IoT CoreTarification](#).
- Pour estimer le coût de votre solution d'architecte, consultez le [calculateur de AWS prix](#).

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.