aws

Guide du développeur

AWS IoT FleetWise

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

AWS IoT FleetWise: Guide du développeur

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

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

Table of Contents

Qu'est-ce que c'est AWS IoT FleetWise ? 1
Avantages 2
Cas d’utilisation 3
Vous êtes nouveau dans le domaine de AWS l'IoT FleetWise ? 4
Accès à AWS l'IoT FleetWise 4
Tarification de AWS l'IoT FleetWise 4
Services connexes 4
Concepts clés 5
Concepts clés 5
Caractéristiques de l' AWS IoT FleetWise 10
AWS Régions prises en charge 11
Configurer l' AWS IoT FleetWise 14
Configurez votre Compte AWS 14
Inscrivez-vous pour un Compte AWS 14
Création d’un utilisateur doté d’un accès administratif 15
Démarrage dans la console 16
Configurez vos paramètres 17
Configurer les paramètres (console) 17
Configurer les paramètres (AWS CLI) 18
Utilisation IPv6 avec l' AWS IoT FleetWise 19
IPv6 prérequis pour les points de terminaison du plan de contrôle 19
IPv6 prise en charge des AWS PrivateLink terminaux 19
Tester la compatibilité des IPv6 adresses 20
Utilisation des IPv6 adresses dans les politiques IAM 20
Utilisation des points de terminaison Dual-Stack 21
Mise en route 24
Introduction 24
Prérequis 25
Étape 1 : configurer le logiciel Edge Agent pour AWS l'IoT FleetWise 26
Étape 2 : Création d'un modèle de véhicule 27
Étape 3 : créer un manifeste du décodeur 29
Étape 4 : Configuration d'un manifeste de décodeur 30
Étape 5 : Création d'un véhicule 31
Étape 6 : créer une campagne 32

Étape 7 : nettoyer 34
Étapes suivantes 34
Ingérer des données 35
Modèles de véhicules 39
Catalogues de signaux 42
Configuration des signaux 45
Création d'un catalogue de signaux 51
Importer un catalogue de signaux 56
Mettre à jour un catalogue de signaux 66
Supprimer un catalogue de signaux 70
Obtenir des informations sur le catalogue de signaux 71
Modèles de véhicules 72
Création d'un modèle de véhicule 73
Mettre à jour un modèle de véhicule 80
Supprimer un modèle de véhicule 82
Obtenir des informations sur le modèle du véhicule 84
Manifestes du décodeur 85
Configuration des interfaces et des signaux 88
Création d'un manifeste de décodeur 91
Mettre à jour le manifeste d'un décodeur 101
Supprimer un manifeste de décodeur 104
Obtenir les informations du manifeste du décodeur 106
Gérer les véhicules 108
Véhicules de ravitaillement 109
Authentifier les véhicules 110
Autoriser les véhicules 112
Rubriques réservées 113
Création d'un véhicule 119
Création d'un véhicule (console) 119
Création d'un véhicule (AWS CLI) 121
Créez plusieurs véhicules 124
Mettre à jour un véhicule 126
Mettre à jour plusieurs véhicules 129
Supprimer un véhicule 130
Supprimer un véhicule (console) 131
Supprimer un véhicule (AWS CLI) 131

Obtenir des informations sur le véhicule 132
Gérez les flottes 134
Création d'une flotte 135
Associer un véhicule à une flotte 136
Dissocier un véhicule d'une flotte 137
Mettre à jour une flotte 138
Supprimer une flotte 139
Vérifier la suppression de la flotte 139
Obtenez des informations sur le parc 140
Gérez les données à l'aide de campagnes 143
Créer une campagne 149
Création d'une campagne (console) 150
Création d'une campagne (AWS CLI) 159
Expressions logiques pour les FleetWise campagnes AWS IoT 164
Mettre à jour une campagne 166
Supprimer une campagne 167
Supprimer une campagne (console) 167
Supprimer une campagne (AWS CLI) 168
Vérifier la suppression de la campagne 168
Obtenir des informations sur les campagnes 168
Stockage et transfert 169
Création de partitions de données 170
Télécharger les données de campagne 173
Téléchargez des données à l'aide de AWS IoT Jobs 174
Collectez les données des codes de diagnostic des problèmes 176
Mots clés du code de diagnostic des problèmes 178
Créez une campagne de collecte de données pour les codes d'erreur diagnostiques 180
Cas d'utilisation du code de diagnostic 183
Visualisez les données du véhicule 186
Traitement des données du véhicule envoyées à un sujet MQTT 187
Traitez les données du véhicule dans Timestream 188
Visualisez les données du véhicule stockées dans Timestream 189
Traitez les données du véhicule dans Amazon S3 189
Format d'objet Amazon S3 191
Analyser les données du véhicule stockées dans Amazon S3 192
Commandes 194

Concepts de commandes 195
Concepts clés des commandes 196
État de l'exécution de la commande 199
Véhicules et commandes 205
Présentation du flux de travail 205
Flux de travail du véhicule 207
Flux de travail des commandes 209
(Facultatif) Notifications de commandes 211
Création et gestion de commandes 212
Création d'une ressource de commande 213
Récupérer des informations sur une commande 215
Répertorier les commandes de votre compte 216
Mettre à jour ou rendre obsolète une ressource de commande 217
Supprimer une ressource de commande 219
Démarrez et surveillez les exécutions de commandes 220
Mettre à jour le résultat d'exécution des commandes 224
Obtenir l'exécution d'une commande 226
Répertorier les exécutions de commandes dans votre compte 227
Supprimer l'exécution d'une commande 229
Exemple : utilisation de commandes 230
Vue d'ensemble de l'exemple du mode de direction du véhicule 231
Prérequis 231
Politique IAM pour l'utilisation des commandes à distance 231
Exécuter AWS IoT des commandes (AWS CLI) 234
Nettoyage 239
Scénarios d'utilisation des commandes 241
Création d'une commande sans paramètres 242
Création d'une commande avec des valeurs par défaut pour les paramètres 243
Création d'une commande avec des valeurs de paramètres 244
Utilisation de commandes avec des modèles d'état 245
Dernier état connu 248
Création d'un modèle d'état 249
Associer un modèle FleetWise d'état AWS IoT à un véhicule 251
Mettre à jour un modèle d'état 252
Supprimer un modèle d'état 253
Obtenir des informations sur le modèle d'état 254

Opérations sur les modèles d'état 255
Activer et désactiver la collecte de données d'état 256
Récupère un instantané de l'état du véhicule 261
Traitez les dernières données connues du véhicule à l'aide de la messagerie MQTT 264
Configuration de la collecte de données indépendante du réseau 268
Introduction 268
Configuration de l’environnement 268
Modèles de données 269
mises à jour du catalogue de signaux 269
Modèle de véhicule et décodeur 271
Envoyer une commande 274
AWS CLI et SDKs 276
Résolution des problèmes 277
Problèmes liés au manifeste du décodeur 277
Problèmes liés aux agents Edge 281
Problème : le logiciel Edge Agent ne démarre pas. 281
Problème : [ERREUR] [Io TFleet WiseEngine : :connect] : [Impossible d'initialiser la bibliothèque de persistance] 283
Problème : Le logiciel Edge Agent ne collecte pas les diagnostics embarqués (OBD) II PIDs et les codes de diagnostic (DTCs). 283
Problème : Le FleetWise logiciel Edge Agent for AWS IoT ne collecte pas de données sur le réseau ou n'est pas en mesure d'appliquer les règles d'inspection des données. 283
Problème : [ERROR] [AwsIotConnectivityModule: :connect] : [Échec de la connexion avec erreur] ou [WARN] [AwsIotChannel: :send] : [Aucune connexion MQTT active.] 285
Stocker et transférer les numéros 285
Problème : réception d'un message `AccessDeniedException` avec toutes les autorisations IAM requises 285
Problème : Les données téléchargées vers AWS IoT Jobs ne tiennent pas compte des `endTime` 285
Problème : le téléchargement des données vers AWS IoT Jobs a un statut `REJECTED` d'exécution. 286
Sécurité 287
Protection des données 288
Le chiffrement au repos dans AWS l'IoT FleetWise 289
Chiffrement en transit 289
Chiffrement des données dans AWS l'IoT FleetWise 290

Contrôle de l’accès 303
AWS IoT FleetWise Autoriser l'envoi et la réception de données sur un sujet MQTT 303
Accorder l' AWS IoT FleetWise accès à une destination Amazon S3 306
Autoriser AWS IoT FleetWise l'accès à une destination Amazon Timestream 309
Accorder AWS IoT Device Management l'autorisation de générer la charge utile pour les commandes avec AWS IoT FleetWise 313
Gestion de l’identité et des accès 317
Public ciblé 318
Authentification par des identités 318
Gestion des accès à l’aide de politiques 322
Comment AWS l'IoT FleetWise fonctionne avec l'IAM 325
Exemples de politiques basées sur l’identité 335
Résolution des problèmes 339
Référence des autorisations d'API 341
Validation de conformité 350
Résilience 352
Sécurité de l’infrastructure 352
Connexion à l' AWS IoT FleetWise via un point de terminaison VPC d'interface 353
Analyse de la configuration et des vulnérabilités 357
Bonnes pratiques de sécurité 357
Accorder le moins d’autorisations possibles 357
Ne journalisez pas les informations sensibles 358
AWS CloudTrail À utiliser pour afficher l'historique des appels d'API 358
Veiller à la synchronisation de l'horloge de votre appareil 358
Surveillance de AWS l'IoT FleetWise 359
Surveillance avec CloudWatch 359
Moniteur avec CloudWatch journaux 364
Afficher les FleetWise journaux AWS IoT dans la CloudWatch console 365
Configuration de la journalisation 371
CloudTrail journaux 374
AWS FleetWise Informations sur l'IoT dans CloudTrail 374
Comprendre les entrées du fichier journal 376
Historique de la documentation 377
..... ccclxxx

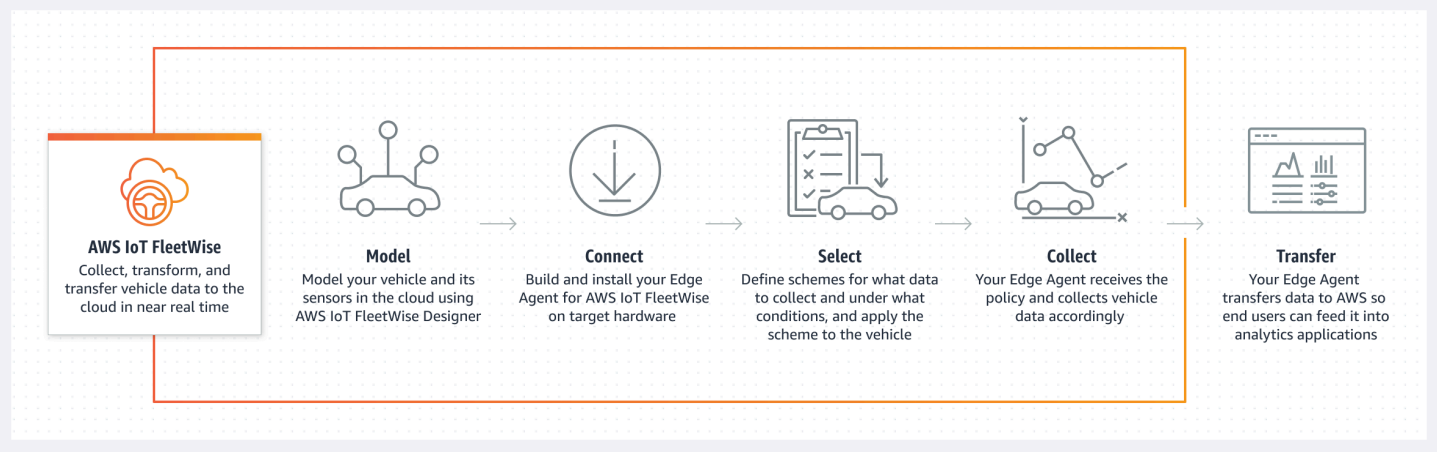
Qu'est-ce que AWS l'IoT FleetWise ?

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

AWS L'IoT FleetWise est un service géré que vous pouvez utiliser pour collecter des données sur les véhicules et les organiser dans le cloud. Vous pouvez utiliser les données collectées pour améliorer la qualité, les performances et l'autonomie du véhicule. Avec AWS l'IoT FleetWise, vous pouvez collecter et organiser les données provenant de véhicules utilisant différents protocoles et formats de données. AWS L'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 campagnes de collecte de données pour contrôler les données du véhicule à collecter et le moment de les transférer vers le cloud.

Lorsque les données du véhicule sont dans le cloud, vous pouvez les utiliser pour des applications qui analysent l'état du parc de véhicules. Ces données peuvent vous aider à identifier les problèmes de maintenance potentiels, à rendre les systèmes multimédia embarqués plus intelligents et à améliorer les technologies avancées telles que la conduite autonome et les systèmes d'assistance au conducteur grâce à l'analyse et à l'apprentissage automatique (ML).

Le schéma suivant montre l'architecture de base de l' AWS IoT FleetWise.



Rubriques

- Avantages
- Cas d’utilisation
- Vous êtes nouveau dans le domaine de AWS l'IoT FleetWise ?
- Accès à AWS l'IoT FleetWise
- Tarification de AWS l'IoT FleetWise
- Services connexes
- Concepts et fonctionnalités clés de l' AWS IoT FleetWise
- AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise

Avantages

Les principaux avantages de l' AWS IoT FleetWise sont les suivants :

Collectez les données du véhicule de manière plus intelligente

> Améliorez la pertinence des données grâce à une collecte de données intelligente qui envoie uniquement les données dont vous avez besoin vers le cloud à des fins d'analyse.

Analysez facilement les données standardisées de l'ensemble du parc

> Analysez les données standardisées d'un parc de véhicules sans avoir à développer un système de collecte ou d'enregistrement de données personnalisé.

Synchronisation automatique des données dans le cloud

> Bénéficiez d'une vue unifiée des données collectées à la fois par des capteurs standard (données de télémétrie) et des systèmes de vision (données provenant de caméras, de radars et de lidars), et synchronisez-les automatiquement dans le cloud. AWS L'IoT FleetWise permet de synchroniser automatiquement les données du système de vision structurées et non structurées, les métadonnées et les données des capteurs standard dans le cloud. Cela rationalise le processus pour obtenir une vue d'ensemble des événements et obtenir des informations.

Stockez les données à la périphérie et transférez-les dans des conditions optimales

> Réduisez les coûts de transmission en stockant temporairement les données sur les véhicules. Vous pouvez transférer des données sélectionnées vers le cloud dans des conditions optimales spécifiées, par exemple lorsque les véhicules se connectent au Wi-Fi.

> **Note**
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Cas d’utilisation

Les scénarios dans lesquels vous pouvez utiliser l' AWS IoT FleetWise sont les suivants :

Entraînez des modèles AI/ML

Améliorez en permanence les modèles d'apprentissage automatique utilisés pour les systèmes d'assistance au conducteur autonomes et avancés en collectant des données provenant de véhicules de production.

Améliorez l'expérience client numérique

Utilisez les données des systèmes d'infodivertissement pour rendre le contenu audiovisuel embarqué dans le véhicule et les informations intégrées aux applications plus pertinents.

Maintenir la santé du parc de véhicules

Utilisez les informations issues des données du parc pour surveiller l'état de santé et le niveau de charge des batteries des véhicules électriques, gérer les programmes de maintenance, analyser la consommation de carburant, etc.

Création et gestion de commandes

Utilisez des commandes pour exécuter des commandes sur un véhicule depuis le cloud. Vous pouvez envoyer des commandes à distance à un véhicule, et en quelques secondes, le véhicule exécutera la commande. Par exemple, vous pouvez configurer des commandes pour verrouiller la portière d'un véhicule ou régler la température.

Création et gestion de modèles d'états

Les modèles d'état fournissent un mécanisme permettant aux propriétaires de véhicules de suivre l'état de leur véhicule. L'agent AWS IoT FleetWise Edge qui s'exécute sur le véhicule collecte et envoie des mises à jour des signaux au cloud.

Vous êtes nouveau dans le domaine de AWS l'IoT FleetWise ?

Si vous débutez dans le AWS domaine de l'IoT FleetWise, nous vous recommandons de commencer par lire les sections suivantes :

- Concepts et fonctionnalités clés de l' AWS IoT FleetWise
- Configurer l' AWS IoT FleetWise
- Tutoriel : Initiez-vous à AWS l'IoT FleetWise
- Ingérez les FleetWise données de AWS l'IoT dans le cloud

Accès à AWS l'IoT FleetWise

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour accéder à AWS l'IoT FleetWise.

Tarification de AWS l'IoT FleetWise

Les véhicules envoient des données au cloud par le biais de messages MQTT. Vous payez à la fin de chaque mois pour les véhicules que vous avez créés dans le cadre de AWS l'IoT FleetWise. Vous payez également pour les messages que vous collectez dans les véhicules. Pour obtenir des informations actualisées sur les tarifs, consultez la page de FleetWise tarification de l'AWS IoT. Pour en savoir plus sur le protocole de messagerie MQTT, consultez MQTT dans le guide du AWS IoT Core développeur.

Services connexes

AWS L'IoT FleetWise s'intègre aux AWS services suivants pour améliorer la disponibilité et l'évolutivité de vos solutions cloud.

- AWS IoT Core— Enregistrez et contrôlez AWS IoT les appareils qui téléchargent les données du véhicule vers AWS l'IoT FleetWise et envoient des commandes à distance à un véhicule. Pour plus d'informations, veuillez consulter la rubrique Présentation de AWS IoT dans le Guide du développeur AWS IoT .
- Amazon Timestream — Utilisez une base de données de séries chronologiques pour stocker et analyser les données de votre véhicule. Pour plus d'informations, consultez la section Qu'est-ce qu'Amazon Timestream dans le guide du développeur Amazon Timestream.

- Amazon S3 — Utilisez un service de stockage d'objets pour stocker et gérer les données de votre véhicule. Pour plus d'informations, consultez la section Qu'est-ce qu'Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Concepts et fonctionnalités clés de l' AWS IoT FleetWise

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les sections suivantes fournissent une vue d'ensemble des composants des FleetWise services AWS IoT et de la manière dont ils interagissent.

Après avoir lu cette introduction, consultez la Configurer l' AWS IoT FleetWise section pour savoir comment configurer AWS l'IoT FleetWise.

Rubriques

- Concepts clés
- Caractéristiques de l' AWS IoT FleetWise

Concepts clés

AWS L'IoT FleetWise fournit un cadre de modélisation des véhicules qui vous permet de modéliser votre véhicule ainsi que ses capteurs et actionneurs dans le cloud. Pour permettre une communication sécurisée entre votre véhicule et le cloud, l' AWS IoT fournit FleetWise également une implémentation de référence pour vous aider à développer le logiciel Edge Agent que vous pouvez installer dans votre véhicule. Vous pouvez définir des schémas de collecte de données dans le cloud et les déployer sur votre véhicule. Le logiciel Edge Agent qui s'exécute dans votre véhicule utilise des schémas de collecte de données pour contrôler les données à collecter et le moment de les transférer vers le cloud.

Les concepts de base de l' AWS IoT sont les suivants FleetWise.

Signal

Les signaux sont des structures fondamentales que vous définissez pour contenir les données du véhicule et ses métadonnées. Un signal peut être un attribut, une branche, un capteur ou un actionneur. Par exemple, vous pouvez créer un capteur pour recevoir les valeurs de température du véhicule et pour stocker ses métadonnées, notamment le nom du capteur, le type de données et une unité. Pour de plus amples informations, veuillez consulter Gérez les catalogues FleetWise de signaux AWS IoT.

Attribut

Les attributs représentent des informations statiques qui ne changent généralement pas, telles que le fabricant et la date de fabrication.

Branche

Les branches représentent des signaux dans une structure imbriquée. Les branches illustrent les hiérarchies de signaux. Par exemple, la `Vehicle` branche possède une branche enfant,`Powertrain`. La `Powertrain` branche possède une branche enfant,`combustionEngine`. Pour localiser la `combustionEngine` branche, utilisez l'`Vehicle.Powertrain.combustionEngine`expression.

Sensor

Les données des capteurs signalent l'état actuel du véhicule et changent au fil du temps, à mesure que l'état du véhicule change, comme le niveau du liquide, les températures, les vibrations ou la tension.

Actuator

Les données de l'actionneur indiquent l'état d'un appareil du véhicule, tel que les moteurs, les appareils de chauffage et les serrures de porte. La modification de l'état d'un dispositif du véhicule peut mettre à jour les données de l'actionneur. Par exemple, vous pouvez définir un actionneur pour représenter le réchauffeur. L'actionneur reçoit de nouvelles données lorsque vous allumez ou éteignez le chauffage.

Structure personnalisée

Une structure personnalisée (également appelée structure) représente une structure de données complexe ou d'ordre supérieur. Il facilite la liaison logique ou le regroupement de données provenant de la même source. Une structure est utilisée lorsque des données sont lues ou écrites dans le cadre d'une opération atomique, par exemple pour représenter un type de données complexe ou une forme d'ordre supérieur.

Un signal de type structure est défini dans le catalogue de signaux en utilisant une référence à un type de données de structure au lieu d'un type de données primitif. Les structures peuvent être utilisées pour tous les types de signaux, y compris les capteurs, les attributs, les actionneurs et les types de données des systèmes de vision. Si un signal de type structure est envoyé ou reçu, l' AWS IoT FleetWise s'attend à ce que tous les éléments inclus aient des valeurs valides. Tous les éléments sont donc obligatoires. Par exemple, si une structure contient les éléments Vehicle.Camera.Image.Height, Vehicle.Camera.Image.Width et Vehicle.Camera.Image.Data, on s'attend à ce que le signal envoyé contienne des valeurs pour tous ces éléments.

> **Note**
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Propriété personnalisée

Une propriété personnalisée représente un membre de la structure de données complexe. Le type de données de la propriété peut être primitif ou d'une autre structure.

Lorsque vous représentez une forme d'ordre supérieur à l'aide d'une structure et d'une propriété personnalisée, la forme d'ordre supérieur prévue est toujours définie et visualisée sous la forme d'une arborescence. La propriété personnalisée est utilisée pour définir tous les nœuds foliaires tandis que la structure est utilisée pour définir tous les nœuds non foliaires.

Catalogue de signaux

Un catalogue de signaux contient un ensemble de signaux. Les signaux d'un catalogue de signaux peuvent être utilisés pour modéliser des véhicules utilisant différents protocoles et formats de données. Par exemple, deux voitures sont fabriquées par des constructeurs automobiles différents : l'une utilise le protocole Control Area Network (bus CAN) ; l'autre utilise le protocole OBD (On-Board Diagnostics). Vous pouvez définir un capteur dans le catalogue de signaux pour recevoir les valeurs de température du véhicule. Ce capteur peut être utilisé pour représenter les thermocouples des deux voitures. Pour de plus amples informations, veuillez consulter Gérez les catalogues FleetWise de signaux AWS IoT.

Modèle de véhicule (manifeste du modèle)

Les modèles de véhicules sont des structures déclaratives que vous pouvez utiliser pour normaliser le format de vos véhicules et pour définir les relations entre les signaux des véhicules.

Les modèles de véhicules garantissent la cohérence des informations entre plusieurs véhicules du même type. Vous ajoutez des signaux pour créer des modèles de véhicules. Pour de plus amples informations, veuillez consulter Gérez les modèles FleetWise de véhicules AWS IoT.

Manifeste du décodeur

Les manifestes du décodeur contiennent des informations de décodage pour chaque signal des modèles de véhicules. Les capteurs et actionneurs des véhicules transmettent des messages de bas niveau (données binaires). Grâce aux manifestes des décodeurs, AWS l'IoT FleetWise est capable de transformer les données binaires en valeurs lisibles par l'homme. Chaque manifeste du décodeur est associé à un modèle de véhicule. Pour de plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

Interface réseau

Contient des informations sur le protocole utilisé par le réseau embarqué. AWS L'IoT FleetWise prend en charge les protocoles suivants.

Controller Area Network (bus CAN)

Protocole qui définit la manière dont les données sont communiquées entre les unités de commande électroniques (ECUs). ECUs peut être l'unité de commande du moteur, les airbags ou le système audio.

Diagnostic embarqué (OBD) II

Un protocole perfectionné qui définit la manière dont les données d'autodiagnostic sont communiquées entre ECUs eux. Il fournit un certain nombre de codes de diagnostic standard (DTCs) qui aident à identifier le problème avec votre véhicule.

Intergiciel pour véhicules

L'intergiciel du véhicule défini comme un type d'interface réseau. Parmi les exemples d'intergiciels pour véhicules, citons le système d'exploitation des robots (ROS 2) et le middleware évolutif orienté service sur IP (SOME/IP).

> **Note**
>
> AWS L'IoT FleetWise prend en charge le middleware ROS 2 pour les données du système de vision.

Interfaces personnalisées

Vous pouvez également utiliser votre propre interface pour décoder les signaux sur le Edge. Cela peut vous faire gagner du temps, car vous n'avez pas besoin de créer de règles de décodage dans le cloud.

Décodeur de signal

Fournit des informations de décodage détaillées pour un signal spécifique. Chaque signal spécifié dans le modèle du véhicule doit être associé à un décodeur de signal. Si le manifeste du décodeur contient des interfaces réseau CAN, il doit contenir les signaux du décodeur CAN. Si le manifeste du décodeur contient des interfaces réseau OBD, il doit contenir des décodeurs de signaux OBD.

Le manifeste du décodeur doit contenir des décodeurs de signaux de message s'il contient également des interfaces intergicielles pour véhicules. Ou, si le manifeste du décodeur contient des interfaces de décodage personnalisées, il doit également contenir des signaux de décodage personnalisés.

Véhicule

Une représentation virtuelle de votre véhicule physique, tel qu'une voiture ou un camion. Les véhicules sont des exemples de modèles de véhicules. Les véhicules créés à partir du même modèle de véhicule héritent du même groupe de signaux. Chaque véhicule correspond à un AWS IoT objet.

Flotte

Une flotte représente un groupe de véhicules. Avant de pouvoir gérer facilement une flotte de véhicules, vous devez associer des véhicules individuels à une flotte.

Campagne

Contient des schémas de collecte de données. Vous définissez une campagne dans le cloud et vous la déployez sur un véhicule ou une flotte. Les campagnes fournissent au logiciel Edge Agent des instructions sur la manière de sélectionner, de collecter et de transférer des données vers le cloud.

Partition de données

Configurez les données partitionnées dans une campagne pour stocker temporairement les données de signal. Vous configurez quand et comment transférer les données vers le cloud.

Schéma de collecte de données

Les schémas de collecte de données fournissent au logiciel Edge Agent des instructions sur la manière de collecter des données. Actuellement, AWS l'IoT FleetWise prend en charge le système de collecte basé sur les conditions et le système de collecte basé sur le temps.

Schéma de collecte basé sur les conditions

Utilisez une expression logique pour identifier les données à collecter. Le logiciel Edge Agent collecte des données lorsque la condition est remplie. Par exemple, si l'expression est`$variable.myVehicle.InVehicleTemperature >35.0`, le logiciel Edge Agent collecte des valeurs de température supérieures à 35,0.

Schéma de collecte basé sur le temps

Spécifiez une période en millisecondes pour définir la fréquence de collecte des données. Par exemple, si la période est de 10 000 millisecondes, le logiciel Edge Agent collecte les données toutes les 10 secondes.

Commandes

Les commandes exécutent des commandes sur un véhicule depuis le cloud. Vous pouvez envoyer des commandes à distance à un véhicule, et en quelques secondes, le véhicule exécutera la commande. Par exemple, vous pouvez configurer des commandes pour verrouiller la portière d'un véhicule ou régler la température.

La commande est une ressource gérée par AWS IoT Device Management. Il contient des configurations réutilisables qui sont appliquées lors de l'envoi d'une exécution de commande au véhicule. Pour plus d'informations, consultez AWS IoT les commandes du guide du AWS IoT Core développeur.

Modèles d'états

Les modèles d'état fournissent un mécanisme permettant aux propriétaires de véhicules de suivre l'état de leur véhicule. L'agent logiciel Edge Agent qui s'exécute sur le véhicule collecte et envoie des mises à jour des signaux au cloud. Chaque modèle d'état contient une liste de signaux à partir desquels les données sont collectées.

Caractéristiques de l' AWS IoT FleetWise

Les principales caractéristiques de l' AWS IoT sont les suivantes FleetWise.

Modélisation de véhicules

Créez des représentations virtuelles de vos véhicules et appliquez un format commun pour organiser les signaux des véhicules. AWS L'IoT FleetWise prend en charge la spécification des signaux du véhicule (VSS) que vous pouvez utiliser pour normaliser les signaux des véhicules.

Collecte de données basée sur un schéma

Définissez des schémas pour transférer uniquement les données de véhicules de grande valeur vers le cloud. Vous pouvez définir des schémas basés sur les conditions pour contrôler les données à collecter, telles que les valeurs de température du véhicule supérieures à 40 degrés. Vous pouvez également définir des schémas temporels pour contrôler la fréquence de collecte des données.

FleetWise Logiciel Edge Agent pour AWS l'IoT

Le logiciel Edge Agent exécuté dans les véhicules facilite la communication entre les véhicules et le cloud. Lorsque les véhicules sont connectés au cloud, le logiciel Edge Agent reçoit en permanence des schémas de collecte de données et collecte les données en conséquence.

AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise

Pour obtenir la liste des AWS régions qui prennent en charge AWS l'IoT FleetWise, consultez la section FleetWise Points de terminaison et quotas AWS IoT. AWS Les FleetWise fonctionnalités de l'IoT diffèrent en termes de support régional.

> **Note**
>
> L'accès à la région Asie-Pacifique (Mumbai) et à certaines FleetWise fonctionnalités de AWS l'IoT sont actuellement fermés. Pour demander l'accès à cette AWS région et à toutes les fonctionnalités protégées, contactez votre responsable de compte ou le centre de support AWS.

Le tableau suivant indique la prise en charge des fonctionnalités par région :

Fonctionnalités/Régions	USA Est (Virginie du Nord)	Europe (Francfort)	Asie-Pacifique (Mumbai) REMARQUE : Accès sécurisé uniquement
Catalogues de signaux	Oui	Oui	Fermée
Modèles de véhicules	Oui	Oui	Fermée
Manifestes du décodeur	Oui	Oui	Fermée
Véhicules	Oui	Oui	Fermée
Flottes	Oui	Oui	Fermée
Campagnes	Oui	Oui	Fermée
Données du système de vision (dans la version préliminaire)	Oui	Oui	Fermée
Sujet MQTT en tant que destination des données de campagne	Fermée	Fermée	Fermée
Stocker et transférer	Fermée	Fermée	Fermée
Commandes	Fermée	Fermée	Fermée
Dernier état connu	Fermée	Fermée	Fermée
Collecte de données indépendante du réseau à l'aide d'une interface de décodage personnalisée	Fermée	Fermée	Fermée

Fonctionnalités/Régions	USA Est (Virginie du Nord)	Europe (Francfort)	Asie-Pacifique (Mumbai) REMARQUE : Accès sécurisé uniquement
Récupération du code de diagnostic (DTC) *	Fermée	Fermée	Fermée

*La récupération de données DTC offre une gamme de fonctionnalités qui vont au-delà de la simple extraction de données DTC. Cette fonctionnalité inclut des fonctionnalités personnalisées qui vous permettent de définir des fonctions en périphérie et de les appeler par leur nom dans des expressions de campagne basées sur des conditions. En outre, il prend en charge la collecte de chaînes illimitées, offrant ainsi une gestion flexible des types de données de chaîne. L'agent Edge peut récupérer des données de manière périodique ou en fonction de conditions spécifiques, ce qui améliore son adaptabilité et son efficacité dans les processus de collecte de données. Pour plus d'informations, consultez le guide des fonctions personnalisées et l'implémentation de référence de la collecte de données DTC dans le guide du développeur de l'agent Edge.

Configurer l' AWS IoT FleetWise

Avant d'utiliser AWS l'IoT FleetWise pour la première fois, suivez les étapes décrites dans les sections suivantes.

Rubriques

- Configurez votre Compte AWS
- Démarrage dans la console
- Configurez vos FleetWise paramètres AWS IoT
- Faire des demandes à l' AWS IoT FleetWise en utilisant IPv6

Configurez votre Compte AWS

Effectuez les tâches suivantes pour vous inscrire AWS et créer un utilisateur administratif.

Inscrivez-vous pour un Compte AWS

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

Pour vous inscrire à un Compte AWS

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

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

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

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

Création d’un utilisateur doté d’un accès administratif

Après vous être inscrit à un Compte AWS, sécurisez Utilisateur racine d'un compte AWS AWS IAM Identity Center, activez et créez un utilisateur administratif afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management Consoletant que propriétaire du compte en choisissant Utilisateur root et en saisissant votre adresse Compte AWS e-mail. Sur la page suivante, saisissez votre mot de passe.

   Pour obtenir de l’aide pour vous connecter en utilisant l’utilisateur racine, consultez Connexion en tant qu’utilisateur racine dans le Guide de l’utilisateur Connexion à AWS .
2. Activez l’authentification multifactorielle (MFA) pour votre utilisateur racine.

   Pour obtenir des instructions, voir Activer un périphérique MFA virtuel pour votre utilisateur Compte AWS root (console) dans le guide de l'utilisateur IAM.

Création d’un utilisateur doté d’un accès administratif

1. Activez IAM Identity Center.

   Pour obtenir des instructions, consultez Activation d’ AWS IAM Identity Center dans le Guide de l’utilisateur AWS IAM Identity Center .
2. Dans IAM Identity Center, octroyez un accès administratif à un utilisateur.

   Pour un didacticiel sur l'utilisation du Répertoire IAM Identity Center comme source d'identité, voir Configurer l'accès utilisateur par défaut Répertoire IAM Identity Center dans le Guide de AWS IAM Identity Center l'utilisateur.

Connexion en tant qu’utilisateur doté d’un accès administratif

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

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

Attribution d'un accès à d'autres utilisateurs

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

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

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

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

> **Note**
>
> Vous pouvez utiliser un rôle lié à un service dans l'IoT AWS . FleetWise Les rôles liés aux services sont prédéfinis par l' AWS IoT FleetWise et incluent les autorisations dont l' AWS IoT FleetWise a besoin pour envoyer des métriques à Amazon. CloudWatch Pour de plus amples informations, veuillez consulter Utilisation de rôles liés à un service pour AWS IoT FleetWise.

Démarrage dans la console

Si vous n'êtes pas encore connecté à votre Compte AWS, connectez-vous, puis ouvrez la FleetWise console AWS IoT. Pour commencer à utiliser l' AWS IoT FleetWise, créez un modèle de véhicule. Un modèle de véhicule normalise le format de vos véhicules.

1. Ouvrez la FleetWise console AWS IoT.
2. Dans Commencer avec AWS IoT FleetWise, choisissez Commencer.

Pour plus d'informations sur la création d'un modèle de véhicule, consultezCréation d'un modèle de FleetWise véhicule AWS IoT.

Configurez vos FleetWise paramètres AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour configurer les paramètres des métriques Amazon CloudWatch Logs, Amazon CloudWatch Logs, et chiffrer les données avec un Clé gérée par AWS.

Grâce aux CloudWatch métriques, vous pouvez surveiller AWS l'IoT FleetWise et d'autres AWS ressources. Vous pouvez utiliser CloudWatch des métriques pour collecter et suivre des métriques, par exemple pour déterminer si une limite de service est dépassée. Pour plus d'informations sur CloudWatch les métriques, consultezSurveillez AWS l'IoT FleetWise avec Amazon CloudWatch.

Avec CloudWatch Logs, AWS l'IoT FleetWise envoie les données des journaux à un groupe de CloudWatch journaux, où vous pouvez les utiliser pour identifier et atténuer les éventuels problèmes. Pour plus d'informations sur CloudWatch les journaux, consultezConfigurer la FleetWise journalisation de AWS l'IoT.

Avec le chiffrement des données, AWS l'IoT FleetWise les utilise Clés gérées par AWS pour chiffrer les données. Vous pouvez également choisir de créer et de gérer des clés avec AWS KMS. iPour de plus amples informations sur le chiffrement, veuillez consulter Chiffrement des données dans AWS l'IoT FleetWise.

Configurer les paramètres (console)

Si vous n'êtes pas encore connecté à votre Compte AWS, connectez-vous, puis ouvrez la FleetWiseconsole AWS IoT.

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de gauche, choisissez Paramètres.
3. Dans Métriques, sélectionnez Activer. AWS L'IoT associe FleetWise automatiquement une politique CloudWatch gérée au rôle lié au service et active CloudWatch les métriques.
4. Dans Logging, choisissez Modifier.
    a. Dans la section de CloudWatch journalisation, entrez le groupe de journaux.
    b. Pour enregistrer vos modifications, choisissez Soumettre.
5. Dans la section Chiffrement, choisissez Modifier.
    a. Choisissez le type de clé que vous souhaitez utiliser. Pour de plus amples informations, veuillez consulter Gestion des clés dans AWS l'IoT FleetWise.

i. Utiliser AWS la clé : AWS l'IoT FleetWise possède et gère la clé.

ii. Choisissez une autre AWS Key Management Service clé : vous gérez celles AWS KMS keys qui se trouvent dans votre compte.

b. Pour enregistrer vos modifications, choisissez Soumettre.

Configurer les paramètres (AWS CLI)

Dans le AWS CLI, enregistrez le compte pour configurer les paramètres.

1. Pour configurer les paramètres, exécutez la commande suivante.

```
aws iotfleetwise register-account
```

2. Pour vérifier vos paramètres, exécutez la commande suivante pour récupérer le statut d'enregistrement.

> **Note**
>
> Le rôle lié à un service est uniquement utilisé pour publier des FleetWise métriques AWS IoT sur. CloudWatch Pour de plus amples informations, veuillez consulter Utilisation de rôles liés à un service pour AWS IoT FleetWise.

```
aws iotfleetwise get-register-account-status
```

Example réponse

```
{
    "accountStatus": "REGISTRATION_SUCCESS",
    "creationTime": "2022-07-28T11:31:22.603000-07:00",
    "customerAccountId": "012345678912",
    "iamRegistrationResponse": {
        "errorMessage": "",
        "registrationStatus": "REGISTRATION_SUCCESS",
        "roleArn": "arn:aws:iam::012345678912:role/AWSIoTFleetwiseServiceRole"
    },
    "lastModificationTime": "2022-07-28T11:31:22.854000-07:00",
    }
```

```
}
```

Le statut de l'enregistrement peut être l'un des suivants :

- `REGISTRATION_SUCCESS`— La AWS ressource a été enregistrée avec succès.
- `REGISTRATION_PENDING`— AWS L'IoT FleetWise est en train de traiter la demande d'enregistrement. Ce processus prend environ cinq minutes.
- `REGISTRATION_FAILURE`— AWS L'IoT ne FleetWise peut pas enregistrer la AWS ressource. Réessayez ultérieurement.

Faire des demandes à l' AWS IoT FleetWise en utilisant IPv6

Vous pouvez communiquer avec AWS IoT FleetWise via le protocole Internet version 6 (IPv6) et IPv4 pour gérer vos ressources. Les terminaux à double pile prennent en charge les demandes adressées à l' AWS IoT FleetWise APIs sur IPv6 et. IPv4 Il n'y a pas de frais supplémentaires pour la communication IPv6.

Le IPv6 protocole est la norme IP de nouvelle génération avec des fonctionnalités de sécurité supplémentaires. Il offre un espace d'adressage de 128 bits et IPv4 une adresse de 32 bits. IPv4 peut générer 4,29 x 10^9 adresses alors qu'il IPv6 peut avoir 3,4 x 10^38 adresses.

IPv6 prérequis pour les points de terminaison du plan de contrôle

IPv6 le support du protocole est automatiquement activé pour les points de terminaison du plan de contrôle. Lorsque vous utilisez les points de terminaison pour les clients du plan de contrôle, vous devez fournir l'extension SNI (Server Name Indication). Les clients peuvent utiliser l'extension SNI pour indiquer le nom du serveur contacté et indiquer s'il utilise les points de terminaison standard ou les points de terminaison à double pile. Consultez Utilisation des points de terminaison Dual-Stack.

IPv6 prise en charge des AWS PrivateLink terminaux

AWS L'IoT FleetWise prend en charge IPv6 la communication avec les points de terminaison VPC d'interface à l'aide de. AWS PrivateLink

Tester la compatibilité des IPv6 adresses

Si vous utilisez use Linux/Unix ou Mac OS X, vous pouvez vérifier si vous pouvez accéder à un point de terminaison à double pile en IPv6 utilisant la commande curl, comme indiqué dans l'exemple suivant :

```
curl -v https://iotfleetwise.<us-east-1>.api.aws
```

Les informations que vous obtenez doivent ressembler à celles de l'exemple ci-dessous. Si vous êtes connecté IPv6, l'adresse IP connectée sera une IPv6 adresse.

```
* Host iotfleetwise.us-east-1.api.aws:443 was resolved.
* IPv6: ::ffff:3.82.78.135, ::ffff:54.211.220.216, ::ffff:54.211.201.157
* IPv4: (none)
*   Trying [::ffff:3.82.78.135]:443...
* Connected to iotfleetwise.us-east-1.api.aws (::ffff:3.82.78.135) port 443
* ALPN: curl offers h2,http/1.1
```

Si vous utilisez Microsoft Windows 7 ou Windows 10, vous pouvez vérifier si vous pouvez accéder à un point de terminaison à double pile via IPv6 ou en IPv4 utilisant la commande ping, comme indiqué dans l'exemple suivant.

```
ping iotfleetwise.<us-east-1>.api.aws
```

Utilisation des IPv6 adresses dans les politiques IAM

Avant IPv6 d'utiliser vos ressources, vous devez vous assurer que toutes les politiques IAM utilisées pour le filtrage des adresses IP incluent des plages d' IPv6 adresses. Pour de plus amples informations sur la gestion des autorisations d'accès avec IAM, veuillez consulter Identity and Access Management pour AWS l'IoT FleetWise.

Les stratégies IAM qui permettent de filtrer les adresses IP utilisent des opérateurs de condition d'adresse IP. La politique suivante identifie la `54.240.143.*` plage d' IPv4 adresses autorisées à l'aide d'opérateurs de condition d'adresse IP. Étant donné que toutes les IPv6 adresses se situent en dehors de la plage autorisée, cette politique empêche la communication à l'aide d' IPv6adresses.

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

```
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "iotfleetwise:*",
      "Resource": "arn:aws:iotfleetwise:*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Pour inclure des IPv6 adresses, vous pouvez modifier l'élément Condition de la politique afin d'autoriser les plages d'adresses IPv4 (54.240.143.0/24) et IPv6 (2001 : : 1234:5678 DB8 : :/64), comme indiqué dans l'exemple suivant.

```
  "Condition": {
    "IpAddress": {
      "aws:SourceIp": [
        "54.240.143.0/24",
        "2001:DB8:1234:5678::/64"
      ]
    }
  }
```

Utilisation des points de terminaison Dual-Stack

AWS Les points de terminaison IoT FleetWise à double pile prennent en charge les demandes adressées à l' AWS IoT FleetWise APIs sur IPv6 et. IPv4 Lorsque vous envoyez une demande à un point de terminaison à double pile, elle est automatiquement résolue en une IPv4 ou une IPv6 adresse. En mode double pile, les connexions IPv6 client IPv4 et les connexions à la fois sont acceptées.

Si vous utilisez l'API REST, vous pouvez accéder directement à un point de FleetWise terminaison AWS IoT en utilisant le nom du point de terminaison (URI). AWS L'IoT ne FleetWise prend en charge que les noms de point de terminaison régionaux à double pile, ce qui signifie Région AWS que vous devez les spécifier dans le nom.

Le tableau suivant indique le format des points de terminaison du plan de contrôle pour AWS l'IoT FleetWise lors de l'utilisation IPv4 et les modes à double pile. Pour plus d'informations sur ces points de terminaison, consultez la section Points de FleetWise terminaison AWS IoT.

Point de terminaison	IPv4 adresse	Mode double pile
Plan de contrôle	au niveau de la flotte. *<region>*.amazonaws.com	au niveau de la flotte. *<region>*.api .aws

Lorsque vous utilisez le AWS CLI et AWS SDKs, vous pouvez utiliser une variable d'`AWS_USE_DUALSTACK_ENDPOINT`environnement ou le `use_dualstack_endpoint` paramètre, qui est un paramètre de fichier de configuration partagé, pour passer à un point de terminaison à double pile. Vous pouvez également spécifier le point de terminaison à double pile directement en remplacement du point de FleetWise terminaison AWS IoT dans le fichier de configuration. Pour plus d'informations, consultez les sections Points de terminaison à double pile et FIPS.

Lorsque vous utilisez le AWS CLI, vous pouvez définir la valeur de configuration `use_dualstack_endpoint` comme `true` dans un profil de votre fichier AWS Config. Cela dirigera toutes les FleetWise demandes AWS IoT effectuées par les commandes vers le point de terminaison à double pile pour la région spécifiée. Vous indiquez la Région dans le fichier de configuration ou dans une commande à l'aide de l'option `--region`.

```
$ aws configure set default.iotfleetwise.use_dualstack_endpoint true
```

Au lieu d'utiliser les points de terminaison à double pile pour toutes les commandes, utilisez ces points de terminaison pour des commandes spécifiques :

- Vous pouvez utiliser le point de terminaison à double pile pour des commandes spécifiques en définissant le `--endpoint-url` paramètre de ces commandes. Par exemple, dans la commande suivante, vous pouvez remplacer le *<endpoint-url>* `toiotfleetwise.`*<region>*`.api.aws`.

  ```
  aws iotfleetwise list-fleets \
    --endpoint-url <endpoint-url>
  ```

- Vous pouvez configurer des profils distincts dans votre fichier AWS Config. Par exemple, créez un profil défini `use_dualstack_endpoint` sur true et un profil non

définiuse_dualstack_endpoint. Lorsque vous exécutez une commande, spécifiez le profil adéquat selon que vous comptez utiliser ou non le point de terminaison Dual-Stack.

Tutoriel : Initiez-vous à AWS l'IoT FleetWise

Avec AWS l'IoT FleetWise, vous pouvez collecter, transformer et transférer les données de votre véhicule. Utilisez le didacticiel présenté dans cette section pour vous familiariser avec AWS l'IoT FleetWise.

Consultez les rubriques suivantes pour en savoir plus sur AWS l'IoT FleetWise :

- Ingérez les FleetWise données de AWS l'IoT dans le cloud
- Modélisez AWS des FleetWise véhicules IoT
- Gérez les FleetWise véhicules AWS IoT
- Gérez les flottes dans l'IoT AWS FleetWise
- Collectez AWS des FleetWise données IoT grâce à des campagnes

Introduction

Utilisez AWS l'IoT FleetWise pour collecter, transformer et transférer le format de données unique des véhicules automatisés vers le cloud en temps quasi réel. Vous avez accès à des informations sur l'ensemble de la flotte. Cela peut vous aider à détecter et à atténuer efficacement les problèmes liés à l'état du véhicule, à transférer des signaux de données de grande valeur et à diagnostiquer les problèmes à distance, tout en réduisant les coûts.

Ce didacticiel vous explique comment démarrer avec AWS l'IoT FleetWise. Vous allez apprendre à créer un modèle de véhicule (manifeste de modèle), un manifeste de décodeur, un véhicule et une campagne.

Pour plus d'informations sur les principaux composants et concepts de l' AWS IoT FleetWise, consultezConcepts et fonctionnalités clés de l' AWS IoT FleetWise.

Durée estimée : 45 minutes environ.

> **Important**
>
> Les FleetWise ressources AWS IoT créées et consommées par cette démo vous seront facturées. Pour plus d'informations, consultez la section AWS IoT sur FleetWise la page de FleetWise tarification de l'AWS IoT.

Prérequis

Pour terminer ce didacticiel de mise en route, vous devez d'abord disposer des éléments suivants :

- Un Compte AWS. Si vous n'en avez pas Compte AWS, reportez-vous à la section Création d'un Compte AWS dans le guide de Gestion de compte AWS référence.
- Accès à un outil Région AWS compatible avec AWS l'IoT FleetWise. Actuellement, AWS l'IoT FleetWise est pris en charge dans l'est des États-Unis (Virginie du Nord) et en Europe (Francfort). Vous pouvez utiliser le sélecteur de région dans le AWS Management Console pour passer à l'une de ces régions. Pour plus d'informations, consultez la section FleetWise Points de terminaison et quotas AWS IoT.
- Ressources Amazon Timestream :
  - Une base de données Amazon Timestream. Pour plus d'informations, consultez la section Création d'une base de données dans le manuel Amazon Timestream Developer Guide.
  - Une table Amazon Timestream créée dans Amazon Timestream qui contiendra vos données. Pour plus d'informations, consultez la section Création d'un tableau dans le manuel Amazon Timestream Developer Guide.
- Démonstration du logiciel Edge Agent. (Les instructions pour configurer la démo se trouvent à l'étape suivante.)
  - Vous pouvez utiliser la démo de démarrage rapide d'Explore Edge Agent pour explorer AWS l'IoT FleetWise et apprendre à développer le logiciel Edge Agent pour AWS l'IoT FleetWise. Cette démo utilise un AWS CloudFormation modèle. Il vous explique comment examiner l'implémentation de référence de l'agent Edge, développer votre agent Edge, puis déployer votre logiciel Edge agent sur un Amazon EC2 Graviton et générer des exemples de données sur le véhicule. La démo fournit également un script que vous pouvez utiliser pour créer un catalogue de signaux, un modèle de véhicule, un manifeste de décodeur, un véhicule, une flotte et une campagne, le tout dans le cloud.
  - Pour télécharger la démo, accédez à la FleetWise console AWS IoT. Sur la page d'accueil du service, dans la FleetWise section Commencer avec AWS l'IoT, choisissez Explore Edge Agent.

Étape 1 : configurer le logiciel Edge Agent pour AWS l'IoT FleetWise

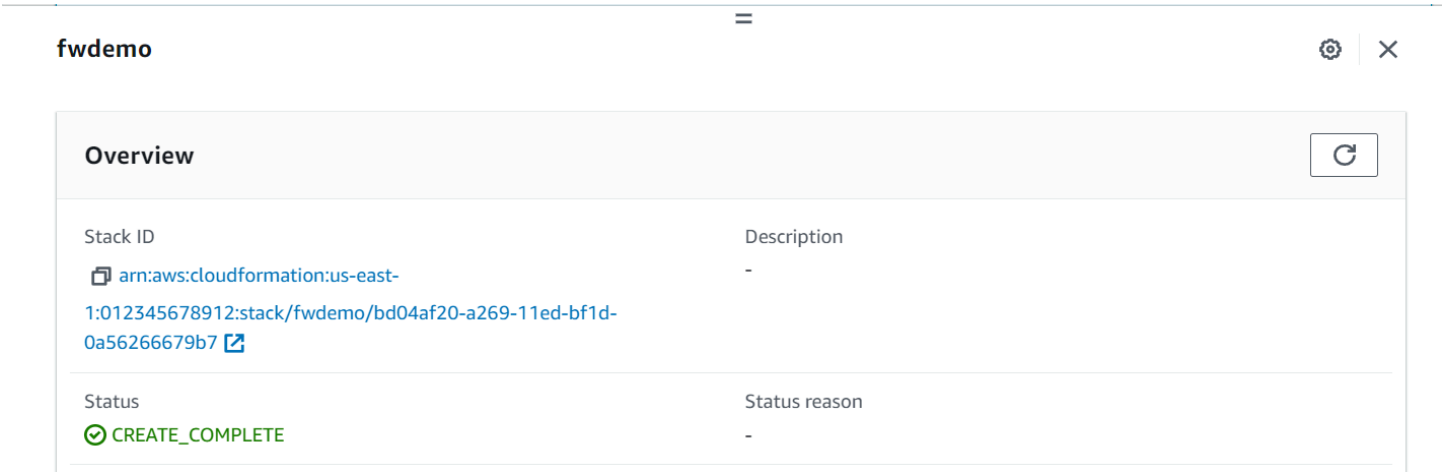
> **Note**
>
> La CloudFormation pile de cette étape utilise des données de télémétrie. Vous pouvez également créer une CloudFormation pile à l'aide des données du système de vision. Pour plus d'informations, consultez le Guide du développeur de données Vision System.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Votre logiciel Edge Agent pour AWS l'IoT FleetWise facilite la communication entre les véhicules et le cloud. Il reçoit des instructions des programmes de collecte de données sur la manière de collecter des données à partir de véhicules connectés au cloud.

Pour configurer votre logiciel Edge Agent, dans Informations générales, procédez comme suit :

1. Ouvrez le CloudFormation modèle de lancement.
2. Sur la page Création rapide d'une pile, dans Nom de la pile, entrez le nom de votre pile de FleetWise ressources AWS IoT. Une pile est un nom convivial qui apparaît sous forme de préfixe sur les noms des ressources créées par ce AWS CloudFormation modèle.
3. Sous Paramètres, entrez vos valeurs personnalisées pour les paramètres liés à votre pile.
   a. Fleetsize - Vous pouvez augmenter le nombre de véhicules de votre flotte en mettant à jour le paramètre Fleetsize.
   b. TCoreRégion Io - Vous pouvez spécifier la région dans laquelle l' AWS IoT objet est créé en mettant à jour le paramètre Io TCore Region. Vous devez utiliser la même région que celle que vous avez utilisée pour créer vos FleetWise véhicules AWS IoT. Pour plus d'informations Régions AWS, consultez Regions and Zones - Amazon Elastic Compute Cloud.
4. Dans la section Fonctionnalités, cochez la case pour confirmer la AWS CloudFormation création de ressources IAM.
5. Choisissez Create stack, puis attendez environ 15 minutes pour que le statut de la pile affiche CREATE_COMPLETE.

6. Pour confirmer que la pile a été créée, choisissez l'onglet Stack info, actualisez la vue et recherchez CREATE_COMPLETE.



> **Important**
>
> Les FleetWise ressources AWS IoT créées et consommées par cette démo vous seront facturées. Pour plus d'informations, consultez la section AWS IoT sur FleetWise la page de FleetWise tarification de l'AWS IoT.

Étape 2 : Création d'un modèle de véhicule

> **Important**
>
> Vous ne pouvez pas créer un modèle de véhicule avec les signaux de données du système de vision dans la FleetWise console AWS IoT. Utilisez plutôt le AWS CLI.

Vous utilisez des modèles de véhicules pour normaliser le format de vos véhicules et pour aider à définir la relation entre les signaux des véhicules que vous créez. Un catalogue de signaux est également créé lorsque vous créez un modèle de véhicule. Un catalogue de signaux est un ensemble de signaux normalisés qui peuvent être réutilisés pour créer des modèles de véhicules. Les signaux sont des structures fondamentales que vous définissez pour contenir les données du véhicule et ses métadonnées. À l'heure actuelle, le FleetWise service AWS IoT ne prend en charge qu'un seul catalogue de signaux Région AWS par compte. Cela permet de vérifier que les données traitées à partir d'un parc de véhicules sont cohérentes.

Pour créer un modèle de véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Sur la page Modèles de véhicules, choisissez Créer un modèle de véhicule.
4. Dans la section Informations générales, entrez le nom du modèle de votre véhicule, tel que Vehicle1, et une description facultative. Ensuite, sélectionnez Suivant.
5. Choisissez un ou plusieurs signaux dans le catalogue de signaux. Vous pouvez filtrer les signaux par nom dans le catalogue de recherche ou les sélectionner dans la liste. Par exemple, vous pouvez choisir des signaux pour la pression des pneus et la pression de freinage afin de collecter des données relatives à ces signaux. Choisissez Suivant.
6. Choisissez vos fichiers .dbc et téléchargez-les depuis votre appareil local. Choisissez Suivant.

   > **Note**
   >
   > Pour ce didacticiel, vous pouvez télécharger un exemple de fichier .dbc à télécharger pour cette étape.

7. Ajoutez des attributs au modèle de votre véhicule, puis cliquez sur Suivant.

   a. Nom - Entrez le nom de l'attribut du véhicule, tel que le nom du fabricant ou la date de fabrication.

   b. Type de données - Dans le menu Type de données, choisissez un type de données.

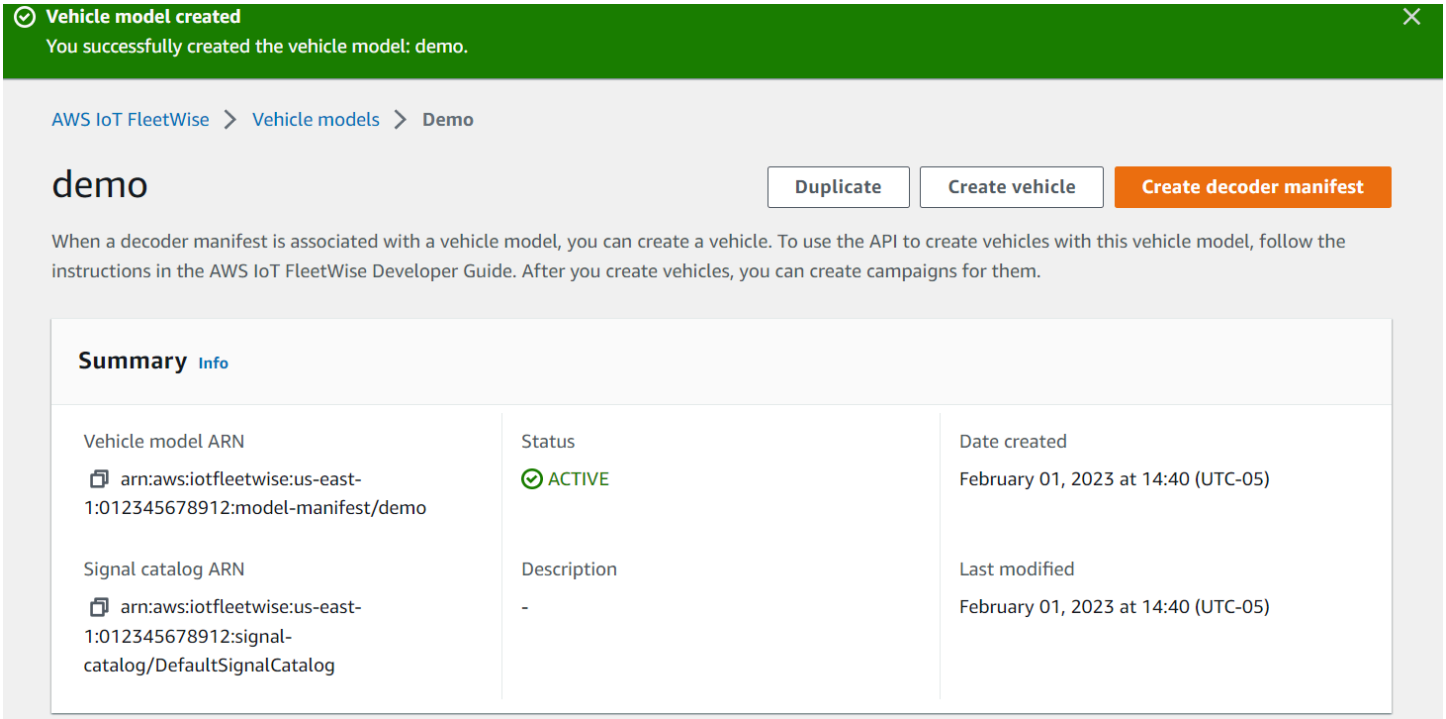
   c. Unité - (Facultatif) Entrez une valeur unitaire, telle que le kilomètre ou le Celsius.

   d. Chemin - (facultatif) Entrez un nom pour le chemin d'accès à un signal, tel que `Vehicle.Engine.Light.` Le point (.) indique qu'il s'agit d'un signal enfant.

   e. Valeur par défaut - (Facultatif) Entrez une valeur par défaut.

   f. Description - (Facultatif) Entrez une description de l'attribut.

8. Passez en revue vos configurations. Lorsque vous êtes prêt, choisissez Create (Créer). Une notification s'affiche indiquant que le modèle de votre véhicule a été créé avec succès.



Étape 3 : créer un manifeste du décodeur

Les manifestes du décodeur sont associés aux modèles de véhicules que vous créez. Ils contiennent des informations qui aident AWS l'IoT à FleetWise décoder et à transformer les données du véhicule d'un format binaire en valeurs lisibles par l'homme qui peuvent être analysées. Les interfaces réseau et les signaux du décodeur sont des composants qui aident à configurer les manifestes du décodeur. Une interface réseau contient des informations sur le protocole CAN ou OBD utilisé par le réseau de votre véhicule. Le signal du décodeur fournit des informations de décodage pour un signal spécifique.

Pour créer un manifeste de décodeur

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Dans la section Modèles de véhicules, choisissez le modèle de véhicule que vous souhaitez utiliser pour créer un manifeste de décodeur.
4. Choisissez Créer un manifeste du décodeur.

Étape 4 : Configuration d'un manifeste de décodeur

Pour configurer un manifeste de décodeur

> **Important**
>
> Vous ne pouvez pas configurer les signaux de données du système de vision dans les manifestes du décodeur à l'aide de la FleetWise console AWS IoT. Utilisez plutôt le AWS CLI. Pour de plus amples informations, veuillez consulter Création d'un manifeste de décodeur ()AWS CLI.

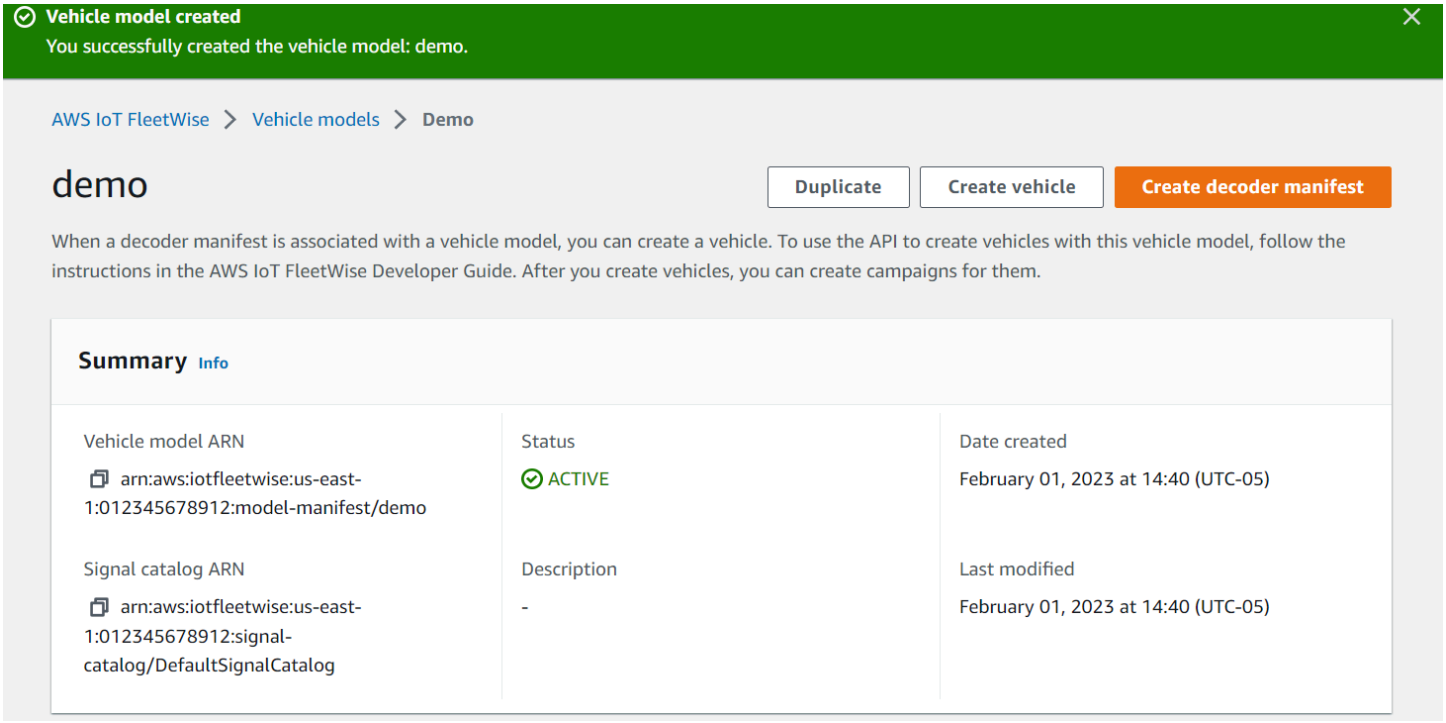
1. Pour vous aider à identifier le manifeste de votre décodeur, entrez un nom et une description facultative pour celui-ci. Ensuite, choisissez Suivant.
2. Pour ajouter une ou plusieurs interfaces réseau, choisissez le type CAN_INTERFACE ou OBD_INTERFACE.
   - Interface de diagnostic embarquée (OBD) - Choisissez ce type d'interface si vous souhaitez un protocole définissant la manière dont les données d'autodiagnostic sont communiquées entre les unités de commande électroniques (). ECUs Ce protocole fournit un certain nombre de codes de diagnostic standard (DTCs) qui peuvent vous aider à résoudre les problèmes liés à votre véhicule.
   - Interface de réseau de zone de contrôle (bus CAN) - Choisissez ce type d'interface si vous souhaitez un protocole définissant la manière dont les données sont communiquées entre elles ECUs. ECUs peuvent être les unités de commande du moteur, les airbags ou le système audio.
3. Entrez le nom de l'interface réseau.
4. Pour ajouter des signaux à l'interface réseau, sélectionnez un ou plusieurs signaux dans la liste.
5. Choisissez un signal de décodeur pour le signal que vous avez ajouté à l'étape précédente. Pour fournir des informations de décodage, téléchargez un fichier .dbc. Chaque signal du modèle de véhicule doit être associé à un signal de décodeur que vous pouvez choisir dans la liste.
6. Pour ajouter une autre interface réseau, choisissez Ajouter une interface réseau. Lorsque vous avez terminé d'ajouter des interfaces réseau, choisissez Next.
7. Passez en revue vos configurations, puis choisissez Créer. Une notification s'affiche indiquant que le manifeste de votre décodeur a été créé avec succès.

Étape 5 : Création d'un véhicule

Dans AWS l'IoT FleetWise, les véhicules sont des représentations virtuelles de votre véhicule physique réel. Tous les véhicules créés à partir du même modèle de véhicule héritent du même groupe de signaux, et chaque véhicule que vous créez correspond à un objet IoT nouvellement créé. Vous devez associer tous les véhicules à un manifeste de décodeur.

Prérequis

1. Vérifiez que vous avez déjà créé le modèle du véhicule et le manifeste du décodeur. Vérifiez également que l'état du modèle de véhicule est ACTIF.
   a. Pour vérifier que l'état du modèle de véhicule est ACTIF, ouvrez la FleetWise console AWS IoT.
   b. Dans le volet de navigation, sélectionnez Modèles de véhicules.
   c. Dans la section Résumé, sous État, vérifiez l'état de votre véhicule.



Pour créer un véhicule

1. Ouvrez la FleetWise console AWS.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Choisissez Créer un véhicule.

4. Pour définir les propriétés du véhicule, entrez le nom du véhicule, puis choisissez un manifeste de modèle (modèle de véhicule) et un manifeste de décodeur.
5. (Facultatif) Pour définir les attributs du véhicule, entrez une paire clé-valeur, puis choisissez Ajouter des attributs.
6. (Facultatif) Pour étiqueter votre ressource AWS, ajoutez des balises, puis choisissez Ajouter une nouvelle balise.
7. Choisissez Suivant.
8. Pour configurer le certificat du véhicule, vous pouvez soit télécharger votre propre certificat, soit choisir Générer automatiquement un nouveau certificat. Nous vous recommandons de générer automatiquement votre certificat pour une configuration plus rapide. Si vous possédez déjà un certificat, vous pouvez choisir de l'utiliser à la place.
9. Téléchargez les fichiers de clé publique et privée, puis choisissez Next.
10. Pour associer une politique au certificat du véhicule, vous pouvez saisir le nom d'une politique existante ou créer une nouvelle politique. Pour créer une nouvelle politique, choisissez Create policy, puis Next.
11. Passez en revue vos configurations. Lorsque vous avez terminé, choisissez Créer un véhicule.

Étape 6 : créer une campagne

Dans AWS l'IoT FleetWise, les campagnes sont utilisées pour faciliter la sélection, la collecte et le transfert des données des véhicules vers le cloud. Les campagnes contiennent des schémas de collecte de données qui fournissent au logiciel Edge Agent des instructions sur la manière de collecter des données à l'aide d'un schéma de collecte basé sur les conditions ou d'un schéma de collecte basé sur le temps.

Création d'une campagne

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Campagnes.
3. Choisissez Create campaign (Créer une campagne).
4. Entrez le nom de votre campagne et une description facultative.
5. Pour configurer le schéma de collecte de données de votre campagne, vous pouvez définir manuellement le schéma de collecte de données ou télécharger un fichier .json depuis votre appareil local. Le téléchargement d'un fichier .json définit automatiquement le schéma de collecte de données.

a. Pour définir manuellement le schéma de collecte de données, choisissez Définir le schéma de collecte de données et choisissez le type de schéma de collecte de données que vous souhaitez utiliser pour votre campagne. Vous pouvez choisir un schéma de collecte basé sur les conditions ou un schéma de collecte basé sur le temps.

b. Si vous choisissez un schéma de collecte basé sur le temps, vous devez spécifier la durée pendant laquelle votre campagne collectera les données du véhicule.

c. Si vous choisissez un schéma de collecte basé sur les conditions, vous devez spécifier une expression pour reconnaître les données à collecter. Assurez-vous de spécifier le nom du signal sous forme de variable, d'opérateur de comparaison et de valeur de comparaison.

d. (Facultatif) Choisissez la version linguistique de votre expression ou conservez-la comme valeur par défaut de 1.

e. (Facultatif) Spécifiez l'intervalle de déclenchement entre deux événements de collecte de données.

f. Pour collecter des données, choisissez la condition du mode déclencheur pour le logiciel Edge Agent. Par défaut, le FleetWise logiciel Edge Agent for AWS IoT collecte toujours des données chaque fois que la condition est remplie. Ou bien, il ne peut collecter des données que lorsque la condition est remplie pour la première fois, lors du premier déclencheur.

g. (Facultatif) Vous pouvez choisir des options de schéma plus avancées.

6. Pour spécifier les signaux à partir desquels le schéma de collecte de données collectera les données, recherchez le nom du signal dans le menu.
7. (Facultatif) Vous pouvez choisir un nombre d'échantillons maximal ou un intervalle d'échantillonnage minimal. Vous pouvez également ajouter d'autres signaux.
8. Choisissez Suivant.
9. Définissez la destination de stockage vers laquelle vous souhaitez que la campagne transfère les données. Vous pouvez stocker des données dans Amazon S3 ou Amazon Timestream.

    a. Amazon S3 — Choisissez le compartiment S3 autorisé à. AWS IoT FleetWise

    b. Amazon Timestream : choisissez la base de données Timestream et le nom de la table. Entrez un rôle IAM qui permet d'envoyer AWS IoT FleetWise des données à Timestream.

10. Choisissez Suivant.
11. Choisissez les attributs ou les noms des véhicules dans le champ de recherche.
12. Entrez la valeur associée à l'attribut ou au nom que vous avez choisi pour votre véhicule.

13. Choisissez les véhicules auprès desquels votre campagne collectera des données. Ensuite, choisissez Suivant.
14. Passez en revue les configurations de votre campagne, puis choisissez Créer une campagne. Vous ou votre équipe devez déployer la campagne sur les véhicules.

Étape 7 : nettoyer

Pour éviter des frais supplémentaires pour les ressources que vous avez utilisées au cours de ce didacticiel, supprimez la AWS CloudFormation pile et toutes les ressources de la pile.

Pour supprimer la AWS CloudFormation pile

1. Ouvrez la AWS CloudFormation console.
2. Dans la liste des piles, choisissez la pile que vous avez créée à l'étape 1.
3. Sélectionnez Delete (Supprimer).
4. Pour confirmer la suppression, choisissez Delete (Supprimer). La suppression de la pile prend environ 15 minutes.

Étapes suivantes

1. Vous pouvez traiter et visualiser les données sur les véhicules collectées par votre campagne. Pour de plus amples informations, veuillez consulter Visualisez les données des FleetWise véhicules AWS IoT.
2. Vous pouvez résoudre les problèmes liés à l' AWS IoT FleetWise. Pour de plus amples informations, veuillez consulter Résolution des problèmes liés à AWS l'IoT FleetWise.

Ingérez les FleetWise données de AWS l'IoT dans le cloud

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Le FleetWise logiciel Edge Agent for AWS IoT, une fois installé et exécuté dans des véhicules, est conçu pour faciliter la communication sécurisée entre vos véhicules et le cloud.

> **Note**
>
> - AWS L'IoT n' FleetWise est pas destiné à être utilisé dans ou en association avec le fonctionnement d'environnements dangereux ou de systèmes critiques susceptibles d'entraîner des blessures corporelles graves ou la mort ou de causer des dommages environnementaux ou matériels. Les données du véhicule collectées par le biais de votre utilisation de AWS l'IoT FleetWise sont fournies à titre informatif uniquement, et vous ne pouvez pas utiliser AWS l'IoT FleetWise pour contrôler ou faire fonctionner les fonctions du véhicule.
> - L'exactitude des données relatives aux véhicules collectées dans le cadre de votre utilisation de l' AWS IoT FleetWise doit être évaluée en fonction de votre cas d'utilisation, notamment afin de respecter les obligations de conformité que vous pourriez avoir en vertu des réglementations applicables en matière de sécurité des véhicules (telles que les obligations de surveillance de la sécurité et de production de rapports). Cette évaluation devrait inclure la collecte et l'examen d'informations par le biais d'autres moyens et sources conformes aux normes de l'industrie (tels que les rapports des conducteurs de véhicules).

Pour ingérer des données dans le cloud, procédez comme suit :

1. Développez et installez votre FleetWise logiciel Edge Agent pour AWS l'IoT dans votre véhicule. Pour plus d'informations sur l'utilisation du logiciel Edge Agent, procédez comme suit pour télécharger le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.
    1. Accédez à la FleetWiseconsole AWS IoT.

2. Sur la page d'accueil du service, dans la FleetWise section Commencer avec AWS l'IoT, choisissez Explore Edge Agent.

2. Créez ou importez un catalogue de signaux contenant des signaux que vous utiliserez pour créer un modèle de véhicule. Pour plus d'informations, consultez Création d'un catalogue FleetWise de signaux AWS IoT et Importer un catalogue de signaux (AWS CLI).

> **Note**
>
> - Si vous utilisez la FleetWise console AWS IoT pour créer le premier modèle de véhicule, il n'est pas nécessaire de créer manuellement un catalogue de signaux. Lorsque vous créez votre premier modèle de véhicule, AWS l'IoT crée FleetWise automatiquement un catalogue de signaux pour vous. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
> - AWS L'IoT prend FleetWise actuellement en charge un catalogue de signaux pour chaque AWS compte et par Région AWS.

3. Utilisez les signaux du catalogue de signaux pour créer un modèle de véhicule. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.

> **Note**
>
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, vous pouvez télécharger des fichiers .dbc pour importer des signaux. .dbc est un format de fichier pris en charge par les bases de données Controller Area Network (bus CAN). Une fois le modèle de véhicule créé, de nouveaux signaux sont automatiquement ajoutés au catalogue de signaux. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
> - Si vous utilisez l'opération `CreateModelManifest` API pour créer un modèle de véhicule, vous devez utiliser l'opération `UpdateModelManifest` API pour activer le modèle de véhicule. Pour de plus amples informations, veuillez consulter Mettre à jour un modèle de FleetWise véhicule AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, AWS l'IoT active FleetWise automatiquement le modèle de véhicule pour vous.

4. Créez un manifeste du décodeur. Le manifeste du décodeur contient des informations de décodage pour chaque signal spécifié dans le modèle de véhicule que vous avez créé à l'étape

précédente. Le manifeste du décodeur est associé au modèle de véhicule que vous avez créé. Pour de plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

> **Note**
>
> - Si vous utilisez l'opération `CreateDecoderManifest` API pour créer un manifeste de décodeur, vous devez utiliser l'opération d'`UpdateDecoderManifest`API pour activer le manifeste de décodeur. Pour de plus amples informations, veuillez consulter Mettre à jour le manifeste d'un FleetWise décodeur AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un manifeste de décodeur, AWS l'IoT active FleetWise automatiquement le manifeste du décodeur pour vous.

5. Créez des véhicules à partir du modèle du véhicule. Les véhicules créés à partir du même modèle de véhicule héritent du même groupe de signaux. Vous devez l'utiliser AWS IoT Core pour approvisionner votre véhicule avant de pouvoir ingérer des données dans le cloud. Pour de plus amples informations, veuillez consulter Gérez les FleetWise véhicules AWS IoT.
6. (Facultatif) Créez un parc pour représenter un groupe de véhicules, puis associez des véhicules individuels au parc. Cela vous permet de gérer plusieurs véhicules en même temps. Pour de plus amples informations, veuillez consulter Gérez les flottes dans l'IoT AWS FleetWise.
7. (Facultatif) Créez des campagnes. Les campagnes sont déployées sur un véhicule ou une flotte de véhicules. Les campagnes fournissent au logiciel Edge Agent des instructions sur la manière de sélectionner, de collecter et de transférer des données vers le cloud. Pour de plus amples informations, veuillez consulter Collectez AWS des FleetWise données IoT grâce à des campagnes. Vous pouvez créer des campagnes, des modèles d'états (ci-dessous) ou les deux pour collecter des données.

   > **Note**
   >
   > Vous devez utiliser l'opération `UpdateCampaign` API pour approuver la campagne avant que l' AWS IoT ne FleetWise puisse la déployer sur le véhicule ou le parc de véhicules. Pour de plus amples informations, veuillez consulter Mettre à jour une FleetWise campagne AWS IoT.

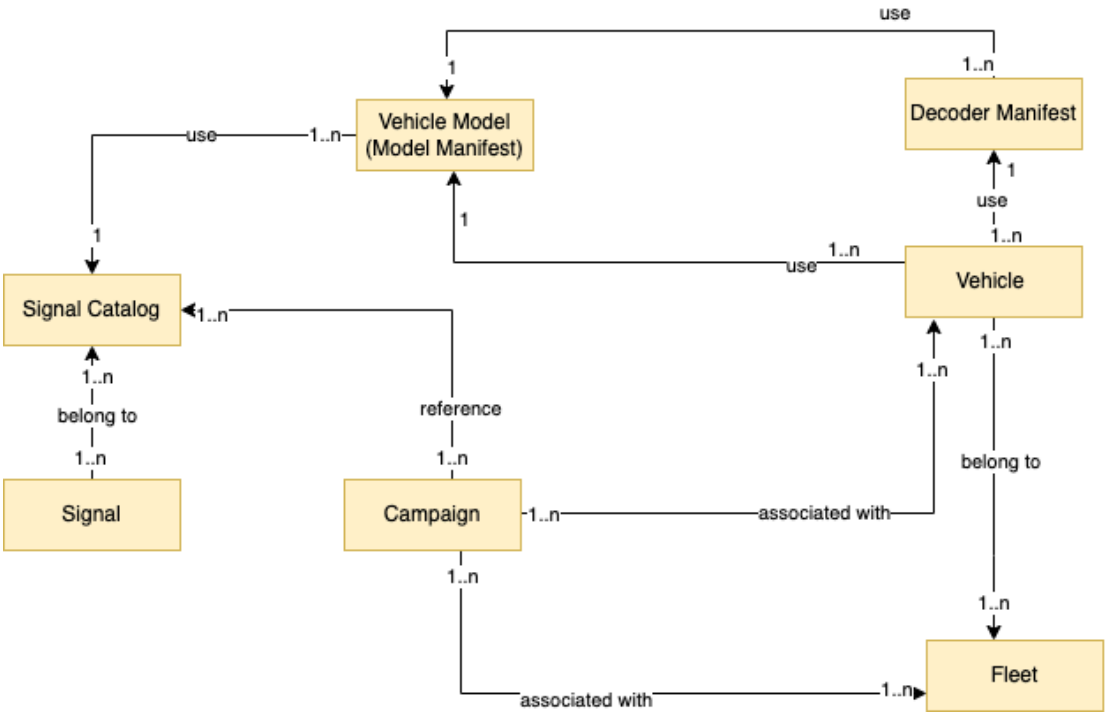
8. (Facultatif) Créez des modèles d'état. Les modèles d'état sont déployés sur un véhicule. Les modèles d'état fournissent un mécanisme permettant aux propriétaires de véhicules de suivre l'état

de leur véhicule. Pour de plus amples informations, veuillez consulter Surveillez le dernier état connu de vos véhicules.

Le logiciel Edge Agent transfère les données du véhicule AWS IoT Core vers un thème MQTT de votre choix. Pour envoyer les données à AWS l'IoT FleetWise pour les campagnes, celui-ci utilise le sujet réservé`$aws/iotfleetwise/vehicles/vehicleName/signals`. Pour le dernier état connu, l'agent Edge utilise le sujet réservé`$aws/iotfleetwise/vehicles/vehicleName/last_known_states/data`. Pour plus d'informations sur le traitement des données ingérées, consultezVisualisez les données des FleetWise véhicules AWS IoT.

Modélisez AWS des FleetWise véhicules IoT

AWS L'IoT FleetWise fournit un cadre de modélisation des véhicules que vous pouvez utiliser pour créer des représentations virtuelles de vos véhicules dans le cloud. Les signaux, les catalogues de signaux, les modèles de véhicules et les manifestes des décodeurs sont les principaux composants avec lesquels vous travaillez pour modéliser vos véhicules.



Signal

Les signaux sont des structures fondamentales que vous définissez pour contenir les données du véhicule et ses métadonnées. Un signal peut être un attribut, une branche, un capteur ou un actionneur. Par exemple, vous pouvez créer un capteur pour recevoir les valeurs de température du véhicule et pour stocker ses métadonnées, notamment le nom du capteur, le type de données et une unité. Pour de plus amples informations, veuillez consulter Gérez les catalogues FleetWise de signaux AWS IoT.

Catalogue de signaux

Un catalogue de signaux contient un ensemble de signaux. Les signaux d'un catalogue de signaux peuvent être utilisés pour modéliser des véhicules utilisant différents protocoles et formats de données. Par exemple, deux voitures sont fabriquées par des constructeurs automobiles différents : l'une utilise le protocole Control Area Network (bus CAN) ; l'autre utilise le protocole

OBD (On-Board Diagnostics). Vous pouvez définir un capteur dans le catalogue de signaux pour recevoir les valeurs de température du véhicule. Ce capteur peut être utilisé pour représenter les thermocouples des deux voitures. Pour de plus amples informations, veuillez consulter Gérez les catalogues FleetWise de signaux AWS IoT.

Modèle de véhicule (manifeste du modèle)

Les modèles de véhicules sont des structures déclaratives que vous pouvez utiliser pour normaliser le format de vos véhicules et pour définir les relations entre les signaux des véhicules. Les modèles de véhicules garantissent la cohérence des informations entre plusieurs véhicules du même type. Vous ajoutez des signaux pour créer des modèles de véhicules. Pour de plus amples informations, veuillez consulter Gérez les modèles FleetWise de véhicules AWS IoT.

Manifeste du décodeur

Les manifestes du décodeur contiennent des informations de décodage pour chaque signal des modèles de véhicules. Les capteurs et actionneurs des véhicules transmettent des messages de bas niveau (données binaires). Grâce aux manifestes des décodeurs, AWS l'IoT FleetWise est capable de transformer les données binaires en valeurs lisibles par l'homme. Chaque manifeste du décodeur est associé à un modèle de véhicule. Pour de plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour modéliser des véhicules de la manière suivante.

1. Créez ou importez un catalogue de signaux contenant des signaux que vous utiliserez pour créer un modèle de véhicule. Pour plus d'informations, consultez Création d'un catalogue FleetWise de signaux AWS IoT et Importer un catalogue de signaux (AWS CLI).

   > **Note**
   >
   > - Si vous utilisez la FleetWise console AWS IoT pour créer le premier modèle de véhicule, il n'est pas nécessaire de créer manuellement un catalogue de signaux. Lorsque vous créez votre premier modèle de véhicule, AWS l'IoT crée FleetWise automatiquement un catalogue de signaux pour vous. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
   > - AWS L'IoT prend FleetWise actuellement en charge un catalogue de signaux pour chaque AWS compte Région AWS.

2. Utilisez les signaux du catalogue de signaux pour créer un modèle de véhicule. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.

> **Note**
>
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, vous pouvez télécharger des fichiers .dbc pour importer des signaux. .dbc est un format de fichier pris en charge par les bases de données Controller Area Network (bus CAN). Une fois le modèle de véhicule créé, de nouveaux signaux sont automatiquement ajoutés au catalogue de signaux. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
> - Si vous utilisez l'opération `CreateModelManifest` API pour créer un modèle de véhicule, vous devez utiliser l'opération `UpdateModelManifest` API pour activer le modèle de véhicule. Pour de plus amples informations, veuillez consulter Mettre à jour un modèle de FleetWise véhicule AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, AWS l'IoT active FleetWise automatiquement le modèle de véhicule pour vous.

3. Créez un manifeste du décodeur. Le manifeste du décodeur contient des informations de décodage pour chaque signal spécifié dans le modèle de véhicule que vous avez créé à l'étape précédente. Le manifeste du décodeur est associé au modèle de véhicule que vous avez créé. Pour de plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

> **Note**
>
> - Si vous utilisez l'opération `CreateDecoderManifest` API pour créer un manifeste de décodeur, vous devez utiliser l'opération d'`UpdateDecoderManifest`API pour activer le manifeste de décodeur. Pour de plus amples informations, veuillez consulter Mettre à jour le manifeste d'un FleetWise décodeur AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un manifeste de décodeur, AWS l'IoT active FleetWise automatiquement le manifeste du décodeur pour vous.

Les bases de données de bus CAN prennent en charge le format de fichier .dbc. Vous pouvez télécharger des fichiers .dbc pour importer des signaux et des décodeurs de signaux. Pour obtenir un exemple de fichier .dbc, procédez comme suit.

Pour obtenir un fichier .dbc

1. Téléchargez le EngineSignalsfichier .zip.
2. Accédez au répertoire où vous avez téléchargé le fichier `EngineSignals.zip`.
3. Décompressez le fichier et enregistrez-le localement sous le nom`EngineSignals.dbc`.

Rubriques

- Gérez les catalogues FleetWise de signaux AWS IoT
- Gérez les modèles FleetWise de véhicules AWS IoT
- Gérez les AWS manifestes des FleetWise décodeurs IoT

Gérez les catalogues FleetWise de signaux AWS IoT

> **Note**
>
> Vous pouvez télécharger un script de démonstration pour convertir les messages ROS 2 en fichiers VSS .json compatibles avec le catalogue de signaux. Pour plus d'informations, consultez le Guide du développeur de données Vision System.

Un catalogue de signaux est un ensemble de signaux normalisés qui peuvent être réutilisés pour créer des modèles de véhicules. AWS L'IoT FleetWise prend en charge la spécification des signaux du véhicule (VSS) que vous pouvez suivre pour définir les signaux. Un signal peut être de l'un des types suivants.

Attribut

Les attributs représentent des informations statiques qui ne changent généralement pas, telles que le fabricant et la date de fabrication.

Branche

Les branches représentent des signaux dans une structure imbriquée. Les branches illustrent les hiérarchies de signaux. Par exemple, la `Vehicle` branche possède une branche enfant,`Powertrain`. La `Powertrain` branche possède une branche enfant,`combustionEngine`. Pour localiser la `combustionEngine` branche, utilisez l'`Vehicle.Powertrain.combustionEngine`expression.

Sensor

Les données des capteurs signalent l'état actuel du véhicule et changent au fil du temps, à mesure que l'état du véhicule change, comme le niveau du liquide, les températures, les vibrations ou la tension.

Actuator

Les données de l'actionneur indiquent l'état d'un appareil du véhicule, tel que les moteurs, les appareils de chauffage et les serrures de porte. La modification de l'état d'un dispositif du véhicule peut mettre à jour les données de l'actionneur. Par exemple, vous pouvez définir un actionneur pour représenter le réchauffeur. L'actionneur reçoit de nouvelles données lorsque vous allumez ou éteignez le chauffage.

Structure personnalisée

Une structure personnalisée (également appelée structure) représente une structure de données complexe ou d'ordre supérieur. Il facilite la liaison logique ou le regroupement de données provenant de la même source. Une structure est utilisée lorsque des données sont lues ou écrites dans le cadre d'une opération atomique, par exemple pour représenter un type de données complexe ou une forme d'ordre supérieur.

Un signal de type structure est défini dans le catalogue de signaux en utilisant une référence à un type de données de structure au lieu d'un type de données primitif. Les structures peuvent être utilisées pour tous les types de signaux, y compris les capteurs, les attributs, les actionneurs et les types de données des systèmes de vision. Si un signal de type structure est envoyé ou reçu, l' AWS IoT FleetWise s'attend à ce que tous les éléments inclus aient des valeurs valides. Tous les éléments sont donc obligatoires. Par exemple, si une structure contient les éléments Vehicle.Camera.Image.Height, Vehicle.Camera.Image.Width et Vehicle.Camera.Image.Data, on s'attend à ce que le signal envoyé contienne des valeurs pour tous ces éléments.

> **Note**
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Propriété personnalisée

Une propriété personnalisée représente un membre de la structure de données complexe. Le type de données de la propriété peut être primitif ou d'une autre structure.

Lorsque vous représentez une forme d'ordre supérieur à l'aide d'une structure et d'une propriété personnalisée, la forme d'ordre supérieur prévue est toujours définie et visualisée sous la forme d'une arborescence. La propriété personnalisée est utilisée pour définir tous les nœuds foliaires tandis que la structure est utilisée pour définir tous les nœuds non foliaires.

> **Note**
>
> - Si vous utilisez la FleetWise console AWS IoT pour créer le premier modèle de véhicule, il n'est pas nécessaire de créer manuellement un catalogue de signaux. Lorsque vous créez votre premier modèle de véhicule, AWS l'IoT crée FleetWise automatiquement un catalogue de signaux pour vous. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, vous pouvez télécharger des fichiers .dbc pour importer des signaux. .dbc est un format de fichier pris en charge par les bases de données Controller Area Network (bus CAN). Une fois le modèle de véhicule créé, de nouveaux signaux sont automatiquement ajoutés au catalogue de signaux. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.
> - AWS L'IoT prend FleetWise actuellement en charge un catalogue de signaux Compte AWS pour chaque région.

AWS L'IoT FleetWise fournit les opérations d'API suivantes que vous pouvez utiliser pour créer et gérer des catalogues de signaux.

- CreateSignalCatalog— Crée un nouveau catalogue de signaux.
- ImportSignalCatalog— Importe des signaux pour créer un catalogue de signaux en téléchargeant un fichier .json. Les signaux doivent être définis en suivant le protocole VSS et enregistrés au format JSON.
- UpdateSignalCatalog— Met à jour un catalogue de signaux existant en mettant à jour, en supprimant ou en ajoutant des signaux.
- DeleteSignalCatalog— Supprime un catalogue de signaux existant.
- ListSignalCatalogs— Récupère une liste paginée de résumés de tous les catalogues de signaux.
- ListSignalCatalogNodes— Récupère une liste paginée de résumés de tous les signaux (nœuds) d'un catalogue de signaux donné.

- GetSignalCatalog— Récupère les informations relatives à un catalogue de signaux.

Didacticiels

- Configuration AWS des FleetWise signaux IoT
- Création d'un catalogue FleetWise de signaux AWS IoT
- Importer un catalogue FleetWise de signaux AWS IoT
- Mettre à jour un catalogue FleetWise de signaux AWS IoT
- Supprimer un catalogue de FleetWise signaux AWS IoT
- Obtenez des informations sur FleetWise le catalogue de signaux AWS IoT

Configuration AWS des FleetWise signaux IoT

Cette section explique comment configurer les branches, les attributs, les capteurs et les actionneurs.

Rubriques

- Configuration des branches
- Configuration des attributs
- Configuration de capteurs ou d'actionneurs
- Configuration de types de données complexes

Configuration des branches

Pour configurer une branche, spécifiez les informations suivantes.

- `fullyQualifiedName`— Le nom complet de la branche est le chemin d'accès à la branche plus le nom de la branche. Utilisez un point (.) pour faire référence à une branche enfant. Par exemple, `Vehicle.Chassis.SteeringWheel` est le nom complet de la `SteeringWheel` branche. `Vehicle.Chassis.`est le chemin d'accès à cette branche.

  Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, deux points (:) et trait de soulignement (_).
- (Facultatif) `Description` — Description de la branche.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `deprecationMessage` — Le message d'obsolescence du nœud ou de la branche déplacé ou supprimé.

  Le DeprecationMessage peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).
- (Facultatif) `comment` — Un commentaire en plus de la description. Un commentaire peut être utilisé pour fournir des informations supplémentaires sur la succursale, telles que la justification de la succursale ou des références à des branches connexes.

  Le commentaire peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

Configuration des attributs

Pour configurer un attribut, spécifiez les informations suivantes.

- `dataType`— Le type de données de l'attribut doit être l'un des suivants : INT8,,,,,, BOOLEAN UINT8 INT16, FLOAT UINT16 INT32, DOUBLE UINT32, STRING INT64 UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 INT16 BOOLEAN_ARRAY, UINT16 DOUBLE_ARRAY, INT32 STRING_ARRAY, UINT32 INT64 UINT64 UNIX_TIMESTAMP_ARRAY, UNKNOWN, ou une structure personnalisée définie dans la branche du type de données. fullyQualifiedName
- `fullyQualifiedName`— Le nom complet de l'attribut est le chemin d'accès à l'attribut plus le nom de l'attribut. Utilisez un point (.) pour faire référence à un signal enfant. Par exemple, `Vehicle.Chassis.SteeringWheel.Diameter` est le nom complet de l'`Diameter`attribut. `Vehicle.Chassis.SteeringWheel.`est le chemin d'accès à cet attribut.

  Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).
- (Facultatif) `Description` — Description de l'attribut.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).
- (Facultatif) `unit` — L'unité scientifique de l'attribut, telle que le km ou le Celsius.
- (Facultatif) `min` — La valeur minimale de l'attribut.
- (Facultatif) `max` — La valeur maximale de l'attribut.
- (Facultatif) `defaultValue` — La valeur par défaut de l'attribut.

- (Facultatif) `assignedValue` — La valeur attribuée à l'attribut.
- (Facultatif) `allowedValues` — Liste des valeurs acceptées par l'attribut.
- (Facultatif) `deprecationMessage` — Le message d'obsolescence pour le nœud ou la branche qui est déplacé ou supprimé.

  Le DeprecationMessage peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).
- (Facultatif) `comment` — Un commentaire en plus de la description. Un commentaire peut être utilisé pour fournir des informations supplémentaires sur l'attribut, telles que la justification de l'attribut ou des références à des attributs associés.

  Le commentaire peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

Configuration de capteurs ou d'actionneurs

Pour configurer un capteur ou un actionneur, spécifiez les informations suivantes.

- `dataType`— Le type de données du signal doit être l'un des suivants : INT8,,,,, BOOLEAN UINT8, FLOAT INT16, DOUBLE UINT16 INT32, STRING UINT32 INT64 UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 INT16 BOOLEAN_ARRAY, UINT16 DOUBLE_ARRAY, INT32 STRING_ARRAY, UINT32 INT64 UINT64 UNIX_TIMESTAMP_ARRAY, UNKNOWN, ou une structure personnalisée définie dans la branche du type de données. fullyQualifiedName
- `fullyQualifiedName`— Le nom complet du signal est le chemin d'accès au signal plus le nom du signal. Utilisez un point (.) pour faire référence à un signal enfant. Par exemple, `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` est le nom complet de l'`HandsOffSteeringState`actionneur. `Vehicle.Chassis.SteeringWheel.HandsOff.`est le chemin d'accès à cet actionneur.

  Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).
- (Facultatif) `Description` — Description du signal.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).
- (Facultatif) `unit` — L'unité scientifique du signal, telle que le km ou le Celsius.

- (Facultatif) `min` — La valeur minimale du signal.
- (Facultatif) `max` — La valeur maximale du signal.
- (Facultatif) `assignedValue` — La valeur attribuée au signal.
- (Facultatif) `allowedValues` — liste des valeurs acceptées par le signal.
- (Facultatif) `deprecationMessage` — Le message d'obsolescence pour le nœud ou la branche qui est déplacé ou supprimé.

  Le DeprecationMessage peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).
- (Facultatif) `comment` — Un commentaire en plus de la description. Un commentaire peut être utilisé pour fournir des informations supplémentaires sur le capteur ou l'actionneur, telles que leur justification ou des références à des capteurs ou actionneurs associés.

  Le commentaire peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

Configuration de types de données complexes

Des types de données complexes sont utilisés lors de la modélisation de systèmes de vision. Outre les branches, ces types de données sont composés de structures (également appelées structures) et de propriétés. Une structure est un signal décrit par plusieurs valeurs, comme une image. Une propriété représente un membre de la structure, comme un type de données primitif (tel que UINT8) ou une autre structure (telle que l'horodatage). Par exemple, Vehicle.Cameras.Front représente une branche, Vehicle.Cameras.Front.Image représente une structure et Vehicle.Cameras.Timestamp représente une propriété.

L'exemple de type de données complexe suivant montre comment les signaux et les types de données sont exportés vers un seul fichier .json.

Example type de données complexe

```
{
  "Vehicle": {
    "type": "branch"
    // Signal tree
  },
  "ComplexDataTypes": {
```

```
    "VehicleDataTypes": {
      // complex data type tree
      "children": {
        "branch": {
          "children": {
            "Struct": {
              "children": {
                "Property": {
                  "type": "property",
                  "datatype": "Data type",
                  "description": "Description",
                  //                  ...
                }
              },
              "description": "Description",
              "type": "struct"
            }
          }
          "description": "Description",
          "type": "branch"
        }
      }
    }
  }
}
```

> **Note**
>
> Vous pouvez télécharger un script de démonstration pour convertir les messages ROS 2 en fichiers VSS .json compatibles avec le catalogue de signaux. Pour plus d'informations, consultez le Guide du développeur de données Vision System.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Configurer la structure

Pour configurer une structure (ou structure) personnalisée, spécifiez les informations suivantes.

- `fullyQualifiedName`— Le nom complet de la structure personnalisée.
  Par exemple, le nom complet d'une structure personnalisée peut être`ComplexDataTypes.VehicleDataTypes.SVMCamera`.

Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).

- (Facultatif) `Description` — Description du signal.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `deprecationMessage` — Le message d'obsolescence pour le nœud ou la branche qui est déplacé ou supprimé.

  Le DeprecationMessage peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `comment` — Un commentaire en plus de la description. Un commentaire peut être utilisé pour fournir des informations supplémentaires sur le capteur ou l'actionneur, telles que leur justification ou des références à des capteurs ou actionneurs associés.

  Le commentaire peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

Configurer la propriété

Pour configurer une propriété personnalisée, spécifiez les informations suivantes.

- `dataType`— Le type de données du signal doit être l'un des suivants : INT8,,,,, BOOLEAN, FLOAT UINT8, DOUBLE INT16 UINT16, STRING INT32 UINT32 INT64, UNIX_TIMESTAMP UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 BOOLEAN_ARRAY, INT16 DOUBLE_ARRAY, UINT16 STRING_ARRAY, INT32 UINT32 UNIX_TIMESTAMP_ARRAY, INT64 STRUCT, UINT64 T_ARRAY, ou UNKNOWN.
- `fullyQualifiedName`— Le nom complet de la propriété personnalisée.
  Par exemple, le nom complet d'une propriété personnalisée peut être`ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`.

  Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement)

- (Facultatif) `Description` — Description du signal.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `deprecationMessage` — Le message d'obsolescence pour le nœud ou la branche qui est déplacé ou supprimé.

  Le DeprecationMessage peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `comment` — Un commentaire en plus de la description. Un commentaire peut être utilisé pour fournir des informations supplémentaires sur le capteur ou l'actionneur, telles que leur justification ou des références à des capteurs ou actionneurs associés.

  Le commentaire peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

- (Facultatif) `dataEncoding` — Indique si la propriété est une donnée binaire. Le codage des données de la propriété personnalisée doit être l'un des suivants : BINAIRE ou TYPED.

- (Facultatif) `structFullyQualifiedName` — Le nom complet du nœud de structure (structure) pour la propriété personnalisée si le type de données de la propriété personnalisée est Struct ou. StructArray

  Le nom complet peut comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).

Création d'un catalogue FleetWise de signaux AWS IoT

Vous pouvez utiliser l'opération CreateSignalCatalogAPI pour créer un catalogue de signaux. L'exemple suivant utilise AWS CLI.

Pour créer un catalogue de signaux, exécutez la commande suivante.

Remplacez *`signal-catalog-configuration`* par le nom du fichier .json contenant la configuration.

```
aws iotfleetwise create-signal-catalog --cli-input-json file://signal-catalog-configuration.json
```

- *`signal-catalog-name`*Remplacez-le par le nom du catalogue de signaux que vous créez.
- (Facultatif) *`description`* Remplacez-le par une description pour vous aider à identifier le catalogue de signaux.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des FleetWise signaux IoT.

```
{
    "name": "signal-catalog-name",
    "description": "description",
    "nodes": [
  {
    "branch": {
      "fullyQualifiedName": "Types"
    }
  },
  {
    "struct": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
    }
  },
  {
    "struct": {
      "fullyQualifiedName": "Types.std_msgs_Header"
    }
  },
  {
    "struct": {
      "fullyQualifiedName": "Types.builtin_interfaces_Time"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.builtin_interfaces_Time.sec",
      "dataType": "INT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.builtin_interfaces_Time.nanosec",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
```

```
      "fullyQualifiedName": "Types.std_msgs_Header.stamp",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.builtin_interfaces_Time"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.std_msgs_Header.frame_id",
      "dataType": "STRING",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.header",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.std_msgs_Header"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.format",
      "dataType": "STRING",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.data",
      "dataType": "UINT8_ARRAY",
      "dataEncoding": "BINARY"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle",
      "description": "Vehicle"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.Cameras"
    }
  },
```

```
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.Cameras.Front"
    }
  },
  {
    "sensor": {
      "fullyQualifiedName": "Vehicle.Cameras.Front.Image",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
    }
  },
  {
    "struct": {
      "fullyQualifiedName": "Types.std_msgs_msg_Float64"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.std_msgs_msg_Float64.data",
      "dataType": "DOUBLE",
      "dataEncoding": "TYPED"
    }
  },
  {
    "sensor": {
      "fullyQualifiedName": "Vehicle.Velocity",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.std_msgs_msg_Float64"
    }
  },
  {
    "struct": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.x_offset",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
```

```
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.y_offset",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.height",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.width",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.do_rectify",
      "dataType": "BOOLEAN",
      "dataEncoding": "TYPED"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.Perception"
    }
  },
  {
    "sensor": {
      "fullyQualifiedName": "Vehicle.Perception.Obstacle",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
    }
  }
]
}
```

> **Note**
>
> Vous pouvez télécharger un script de démonstration pour convertir les messages ROS 2 en fichiers VSS .json compatibles avec le catalogue de signaux. Pour plus d'informations, consultez le Guide du développeur de données Vision System.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateSignalCatalog` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Importer un catalogue FleetWise de signaux AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour importer un catalogue de signaux.

Rubriques

- Importer un catalogue de signaux (console)
- Importer un catalogue de signaux (AWS CLI)

Importer un catalogue de signaux (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour importer un catalogue de signaux.

> **Important**
>
> Vous ne pouvez avoir qu'un seul catalogue de signaux. Si vous possédez déjà un catalogue de signaux, vous ne verrez pas l'option permettant d'importer un catalogue de signaux dans la console.

Pour importer un catalogue de signaux

1. Ouvrez la FleetWiseconsole AWS IoT.
2. Dans le volet de navigation, choisissez Signal catalog.
3. Sur la page récapitulative du catalogue de signaux, choisissez Importer le catalogue de signaux.
4. Importez le fichier contenant les signaux.
   - Pour télécharger un fichier depuis un compartiment S3, procédez comme suit :
     a. ChoisissezImport from S3 (Importer depuis S3).
     b. Choisissez Parcourir S3.
     c. Pour les compartiments, entrez le nom ou l'objet du compartiment, choisissez-le dans la liste, puis choisissez le fichier dans la liste. Cliquez sur le bouton Choisir un fichier.

     Ou, pour l'URI S3, entrez un URI Amazon Simple Storage Service. Pour plus d'informations, consultez la section Méthodes d'accès à un compartiment dans le guide de l'utilisateur Amazon S3.
   - Pour télécharger un fichier depuis votre ordinateur, procédez comme suit :
     a. Choisissez Importer depuis un fichier.
     b. Téléchargez un fichier .json au format VSS (Vehicle Signal Specification).
5. Vérifiez le catalogue de signaux, puis choisissez Importer un fichier.

Importer un catalogue de signaux (AWS CLI)

Vous pouvez utiliser l'opération ImportSignalCatalogAPI pour télécharger un fichier JSON qui permet de créer un catalogue de signaux. Vous devez suivre la spécification des signaux du véhicule (VSS) pour enregistrer les signaux dans le fichier JSON. L'exemple suivant utilise AWS CLI.

Pour importer un catalogue de signaux, exécutez la commande suivante.

- *signal-catalog-name*Remplacez-le par le nom du catalogue de signaux que vous créez.
- (Facultatif) Remplacez la description par un *description* pour vous aider à identifier le catalogue de signaux.
- Remplacez *signal-catalog-configuration-vss* par le nom du fichier de chaîne JSON qui contient les signaux définis dans VSS.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des FleetWise signaux IoT.

```
aws iotfleetwise import-signal-catalog \
                --name signal-catalog-name \
                --description  description \
                --vss file://signal-catalog-configuration-vss.json
```

Le JSON doit être stringifié et transmis par le champ. `vssJson` Voici un exemple de signaux définis dans VSS.

```
{
 "Vehicle": {
  "type": "branch",
  "children": {
   "Chassis": {
    "type": "branch",
    "description": "All data concerning steering, suspension, wheels, and brakes.",
    "children": {
     "SteeringWheel": {
      "type": "branch",
      "description": "Steering wheel signals",
      "children": {
       "Diameter": {
        "type": "attribute",
        "description": "The diameter of the steering wheel",
```

```
        "datatype": "float",
        "unit": "cm",
        "min": 1,
        "max": 50
       },
       "HandsOff": {
        "type": "branch",
        "children": {
         "HandsOffSteeringState": {
          "type": "actuator",
          "description": "HndsOffStrWhlDtSt. Hands Off Steering State",
          "datatype": "boolean"
         },
         "HandsOffSteeringMode": {
          "type": "actuator",
          "description": "HndsOffStrWhlDtMd. Hands Off Steering Mode",
          "datatype": "int8",
          "min": 0,
          "max": 2
         }
        }
       }
      }
     },
     "Accelerator": {
      "type": "branch",
      "description": "",
      "children": {
       "AcceleratorPedalPosition": {
        "type": "sensor",
        "description": "Throttle__Position. Accelerator pedal position as percent. 0 =
Not depressed. 100 = Fully depressed.",
        "datatype": "uint8",
        "unit": "%",
        "min": 0,
        "max": 100.000035
       }
      }
     }
    }
   },
   "Powertrain": {
    "type": "branch",
    "description": "Powertrain data for battery management, etc.",
```

```
    "children": {
     "Transmission": {
      "type": "branch",
      "description": "Transmission-specific data, stopping at the drive shafts.",
      "children": {
       "VehicleOdometer": {
        "type": "sensor",
        "description": "Vehicle_Odometer",
        "datatype": "float",
        "unit": "km",
        "min": 0,
        "max": 67108863.984375
       }
      }
     },
     "CombustionEngine": {
      "type": "branch",
      "description": "Engine-specific data, stopping at the bell housing.",
      "children": {
       "Engine": {
        "type": "branch",
        "description": "Engine description",
        "children": {
         "timing": {
          "type": "branch",
          "description": "timing description",
          "children": {
           "run_time": {
            "type": "sensor",
            "description": "Engine run time",
            "datatype": "int16",
            "unit": "ms",
            "min": 0,
            "max": 10000
           },
           "idle_time": {
            "type": "sensor",
            "description": "Engine idle time",
            "datatype": "int16",
            "min": 0,
            "unit": "ms",
            "max": 10000
           }
          }
```

```
        }
       }
      }
     }
    }
   }
  },
  "Axle": {
   "type": "branch",
   "description": "Axle signals",
   "children": {
    "TireRRPrs": {
     "type": "sensor",
     "description": "TireRRPrs. Right rear Tire pressure in kilo-Pascal",
     "datatype": "float",
     "unit": "kPaG",
     "min": 0,
     "max": 1020
    }
   }
  }
 }
},
"Cameras": {
 "type": "branch",
 "description": "Branch to aggregate all cameras in the vehicle",
 "children": {
  "FrontViewCamera": {
   "type": "sensor",
   "datatype": "VehicleDataTypes.SVMCamera",
   "description": "Front view camera"
  },
  "RearViewCamera": {
   "type": "sensor",
   "datatype": "VehicleDataTypes.SVMCamera",
   "description": "Rear view camera"
  },
  "LeftSideViewCamera": {
   "type": "sensor",
   "datatype": "VehicleDataTypes.SVMCamera",
   "description": "Left side view camera"
  },
  "RightSideViewCamera": {
   "type": "sensor",
```

```
    "datatype": "VehicleDataTypes.SVMCamera",
    "description": "Right side view camera"
   }
  }
 },
 "ComplexDataTypes": {
  "VehicleDataTypes": {
   "type": "branch",
   "description": "Branch to aggregate all camera related higher order data types",
   "children": {
    "SVMCamera": {
     "type": "struct",
     "description": "This data type represents Surround View Monitor (SVM) camera
system in a vehicle",
     "comment": "Test comment",
     "deprecation": "Test deprecation message",
     "children": {
      "Make": {
       "type": "property",
       "description": "Make of the SVM camera",
       "datatype": "string",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Description": {
       "type": "property",
       "description": "Description of the SVM camera",
       "datatype": "string",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "FPS": {
       "type": "property",
       "description": "FPS of the SVM camera",
       "datatype": "double",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Orientation": {
       "type": "property",
       "description": "Orientation of the SVM camera",
       "datatype": "VehicleDataTypes.Orientation",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
```

```
      },
      "Range": {
       "type": "property",
       "description": "Range of the SVM camera",
       "datatype": "VehicleDataTypes.Range",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "RawData": {
       "type": "property",
       "description": "Represents binary data of the SVM camera",
       "datatype": "uint8[]",
       "dataencoding": "binary",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "CapturedFrames": {
       "type": "property",
       "description": "Represents selected frames captured by the SVM camera",
       "datatype": "VehicleDataTypes.Frame[]",
       "dataencoding": "typed",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      }
     }
    },
    "Range": {
     "type": "struct",
     "description": "Range of a camera in centimeters",
     "comment": "Test comment",
     "deprecation": "Test deprecation message",
     "children": {
      "Min": {
       "type": "property",
       "description": "Minimum range of a camera in centimeters",
       "datatype": "uint32",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Max": {
       "type": "property",
       "description": "Maximum range of a camera in centimeters",
       "datatype": "uint32",
       "comment": "Test comment",
```

```
       "deprecation": "Test deprecation message"
      }
     }
    },
    "Orientation": {
     "type": "struct",
     "description": "Orientation of a camera",
     "comment": "Test comment",
     "deprecation": "Test deprecation message",
     "children": {
      "Front": {
       "type": "property",
       "description": "Indicates whether the camera is oriented to the front of the
vehicle",
       "datatype": "boolean",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Rear": {
       "type": "property",
       "description": "Indicates whether the camera is oriented to the rear of the
vehicle",
       "datatype": "boolean",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      },
      "Side": {
       "type": "property",
       "description": "Indicates whether the camera is oriented to the side of the
vehicle",
       "datatype": "boolean",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      }
     }
    },
    "Frame": {
     "type": "struct",
     "description": "Represents a camera frame",
     "comment": "Test comment",
     "deprecation": "Test deprecation message",
     "children": {
      "Data": {
       "type": "property",
```

```
       "datatype": "string",
       "dataencoding": "binary",
       "comment": "Test comment",
       "deprecation": "Test deprecation message"
      }
     }
    }
   }
  }
 }

}
```

L'exemple suivant montre les mêmes signaux définis dans VSS dans une chaîne JSON.

```
{
   "vssJson": "{\"Vehicle\":{\"type\":\"branch\",\"children\":{\"Chassis\":{\"type
\":\"branch\",\"description\":\"All data concerning steering, suspension, wheels,
 and brakes.\",\"children\":{\"SteeringWheel\":{\"type\":\"branch\",\"description
\":\"Steering wheel signals\",\"children\":{\"Diameter\":{\"type\":\"attribute\",
\"description\":\"The diameter of the steering wheel\",\"datatype\":\"float\",\"unit
\":\"cm\",\"min\":1,\"max\":50},\"HandsOff\":{\"type\":\"branch\",\"children\":
{\"HandsOffSteeringState\":{\"type\":\"actuator\",\"description\":\"HndsOffStrWhlDtSt.
 Hands Off Steering State\",\"datatype\":\"boolean\"},\"HandsOffSteeringMode\":
{\"type\":\"actuator\",\"description\":\"HndsOffStrWhlDtMd. Hands Off Steering Mode
\",\"datatype\":\"int8\",\"min\":0,\"max\":2}}}}},\"Accelerator\":{\"type\":\"branch
\",\"description\":\"\",\"children\":{\"AcceleratorPedalPosition\":{\"type\":\"sensor
\",\"description\":\"Throttle__Position. Accelerator pedal position as percent. 0
 = Not depressed. 100 = Fully depressed.\",\"datatype\":\"uint8\",\"unit\":\"%\",
\"min\":0,\"max\":100.000035}}}}},\"Powertrain\":{\"type\":\"branch\",\"description
\":\"Powertrain data for battery management, etc.\",\"children\":{\"Transmission\":
{\"type\":\"branch\",\"description\":\"Transmission-specific data, stopping at the
 drive shafts.\",\"children\":{\"VehicleOdometer\":{\"type\":\"sensor\",\"description
\":\"Vehicle_Odometer\",\"datatype\":\"float\",\"unit\":\"km\",\"min\":0,\"max
\":67108863.984375}}},\"CombustionEngine\":{\"type\":\"branch\",\"description\":
\"Engine-specific data, stopping at the bell housing.\",\"children\":{\"Engine\":
{\"type\":\"branch\",\"description\":\"Engine description\",\"children\":{\"timing\":
{\"type\":\"branch\",\"description\":\"timing description\",\"children\":{\"run_time\":
{\"type\":\"sensor\",\"description\":\"Engine run time\",\"datatype\":\"int16\",\"unit
\":\"ms\",\"min\":0,\"max\":10000},\"idle_time\":{\"type\":\"sensor\",\"description
\":\"Engine idle time\",\"datatype\":\"int16\",\"min\":0,\"unit\":\"ms\",\"max
\":10000}}}}}}}},\"Axle\":{\"type\":\"branch\",\"description\":\"Axle signals\",
\"children\":{\"TireRRPrs\":{\"type\":\"sensor\",\"description\":\"TireRRPrs. Right
```

```
 rear Tire pressure in kilo-Pascal\",\"datatype\":\"float\",\"unit\":\"kPaG\",\"min
\":0,\"max\":1020}}}}}}"
}
```

> **Note**
>
> Vous pouvez télécharger un script de démonstration pour convertir les messages ROS 2 en fichiers JSON VSS compatibles avec le catalogue de signaux. Pour plus d'informations, consultez le Guide du développeur de données Vision System.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ImportSignalCatalog` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
{
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Mettre à jour un catalogue FleetWise de signaux AWS IoT

Vous pouvez utiliser l'opération UpdateSignalCatalogAPI pour mettre à jour un catalogue de signaux existant. L'exemple suivant utilise AWS CLI.

Pour mettre à jour un catalogue de signaux existant, exécutez la commande suivante.

Remplacez *signal-catalog-configuration* par le nom du fichier .json contenant la configuration.

```
aws iotfleetwise update-signal-catalog --cli-input-json file://signal-catalog-configuration.json
```

*signal-catalog-name*Remplacez-le par le nom du catalogue de signaux que vous mettez à jour.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des FleetWise signaux IoT.

> **Important**
>
> Les structures personnalisées sont immuables. Si vous devez réorganiser ou insérer des propriétés dans une structure personnalisée existante (structure), supprimez la structure et créez une toute nouvelle structure avec l'ordre de propriétés souhaité.
> Pour supprimer une structure personnalisée, ajoutez le nom complet de la structure`nodesToRemove`. Une structure ne peut pas être supprimée si elle est référencée par des signaux. Tous les signaux faisant référence à la structure (leur type de données est défini comme étant la structure cible) doivent être mis à jour ou supprimés avant la demande de mise à jour du catalogue de signaux.

```
{
     "name": "signal-catalog-name",
     "nodesToAdd": [{
       "branch": {
        "description": "Front left of vehicle specific data.",
        "fullyQualifiedName": "Vehicle.Front.Left"
       }
      },
      {
       "branch": {
        "description": "Door-specific data for the front left of vehicle.",
        "fullyQualifiedName": "Vehicle.Front.Left.Door"
       }
      },
      {
       "actuator": {
```

```
        "fullyQualifiedName": "Vehicle.Front.Left.Door.Lock",
        "description": "Whether the front left door is locked.",
        "dataType": "BOOLEAN"
       }
      },
      {
       "branch": {
        "fullyQualifiedName": "Vehicle.Camera"
       }
      },
      {
       "struct": {
        "fullyQualifiedName": "Vehicle.Camera.SVMCamera"
       }
      },
      {
       "property": {
        "fullyQualifiedName": "Vehicle.Camera.SVMCamera.ISO",
        "dataType": "STRING"
       }
      }
     ],
     "nodesToRemove": ["Vehicle.Chassis.SteeringWheel.HandsOffSteeringState"],
     "nodesToUpdate": [{
      "attribute": {
       "dataType": "FLOAT",
       "fullyQualifiedName": "Vehicle.Chassis.SteeringWheel.Diameter",
       "max": 55
      }
     }]
    }
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateSignalCatalog` d'API.

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

```
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Vérifier la mise à jour du catalogue de signaux

Vous pouvez utiliser l'opération ListSignalCatalogNodesAPI pour vérifier si un catalogue de signaux a été mis à jour. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de tous les signaux (nœuds) d'un catalogue de signaux donné, exécutez la commande suivante.

*signal-catalog-name*Remplacez-le par le nom du catalogue de signaux que vous êtes en train de vérifier.

```
aws iotfleetwise list-signal-catalog-nodes --name signal-catalog-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListSignalCatalogNodes` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer un catalogue de FleetWise signaux AWS IoT

Vous pouvez utiliser l'opération DeleteSignalCatalogAPI pour supprimer un catalogue de signaux. L'exemple suivant utilise AWS CLI.

> **Important**
>
> Avant de supprimer un catalogue de signaux, assurez-vous qu'aucun modèle de véhicule, manifeste de décodeur, véhicule, flotte ou campagne n'y est associé. Pour obtenir des instructions, veuillez consulter les sections suivantes :
>
> - Supprimer un modèle de FleetWise véhicule AWS IoT
> - Supprimer le manifeste d'un FleetWise décodeur AWS IoT
> - Supprimer un FleetWise véhicule AWS IoT
> - Supprimer un FleetWise parc AWS IoT
> - Supprimer une FleetWise campagne AWS IoT

Pour supprimer un catalogue de signaux existant, exécutez la commande suivante. *signal-catalog-name*Remplacez-le par le nom du catalogue de signaux que vous supprimez.

```
aws iotfleetwise delete-signal-catalog --name signal-catalog-name
```

Vérifier la suppression du catalogue de signaux

Vous pouvez utiliser l'opération ListSignalCatalogsAPI pour vérifier si un catalogue de signaux a été supprimé. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de tous les catalogues de signaux, exécutez la commande suivante.

```
aws iotfleetwise list-signal-catalogs
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListSignalCatalogs` d'API.

```
{
```

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Obtenez des informations sur FleetWise le catalogue de signaux AWS IoT

Vous pouvez utiliser l'opération GetSignalCatalogAPI pour récupérer les informations du catalogue de signaux. L'exemple suivant utilise AWS CLI.

Pour récupérer des informations sur un catalogue de signaux, exécutez la commande suivante.

*signal-catalog-name*Remplacez-le par le nom du catalogue de signaux que vous souhaitez récupérer.

```
aws iotfleetwise get-signal-catalog --name signal-catalog-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `GetSignalCatalog` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
```

```
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au catalogue de signaux peuvent ne pas être répercutées immédiatement.

Gérez les modèles FleetWise de véhicules AWS IoT

Vous utilisez des signaux pour créer des modèles de véhicules qui aident à normaliser le format de vos véhicules. Les modèles de véhicules fournissent des informations cohérentes sur plusieurs véhicules du même type, afin que vous puissiez traiter les données des flottes de véhicules. Les véhicules créés à partir du même modèle de véhicule héritent du même groupe de signaux. Pour de plus amples informations, veuillez consulter Gérez les FleetWise véhicules AWS IoT.

Chaque modèle de véhicule possède un champ d'état qui contient l'état du modèle de véhicule. L'état peut avoir l'une des valeurs suivantes :

- `ACTIVE`— Le modèle du véhicule est actif.
- `DRAFT`— La configuration du modèle de véhicule est enregistrée.

> **Important**
>
> - Vous devez disposer d'un catalogue de signaux avant de pouvoir créer un modèle de véhicule à l'aide de l'opération `CreateModelManifest` API. Pour de plus amples informations, veuillez consulter Création d'un catalogue FleetWise de signaux AWS IoT.
> - Si vous utilisez la FleetWise console AWS IoT pour créer un modèle de véhicule, AWS l'IoT active FleetWise automatiquement le modèle de véhicule pour vous.
> - Si vous utilisez l'opération `CreateModelManifest` API pour créer un modèle de véhicule, le modèle de véhicule reste dans `DRAFT` cet état.

- Vous ne pouvez pas créer de véhicules à partir de modèles de véhicules qui se trouvent dans l'`DRAFT`État. Utilisez l'opération `UpdateModelManifest` API pour modifier l'`ACTIVE`état des modèles de véhicules.
- Vous ne pouvez pas modifier les modèles de véhicules qui se trouvent dans `ACTIVE` cet État.

Rubriques

- Création d'un modèle de FleetWise véhicule AWS IoT
- Mettre à jour un modèle de FleetWise véhicule AWS IoT
- Supprimer un modèle de FleetWise véhicule AWS IoT
- Obtenez des informations sur les modèles de FleetWise véhicules AWS IoT

Création d'un modèle de FleetWise véhicule AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour créer des modèles de véhicules.

Rubriques

- Création d'un modèle de véhicule (console)
- Création d'un modèle de véhicule (AWS CLI)

Création d'un modèle de véhicule (console)

Dans la FleetWise console AWS IoT, vous pouvez créer un modèle de véhicule de différentes manières :

- Utilisez un modèle fourni par AWS
- Création manuelle d'un modèle de véhicule
- Dupliquer un modèle de véhicule

Utilisez un modèle fourni par AWS

AWS L'IoT FleetWise fournit un modèle de diagnostic embarqué (OBD) II, J1979 qui crée automatiquement un catalogue de signaux, un modèle de véhicule et un manifeste de décodeur pour vous. Le modèle ajoute également des interfaces réseau OBD au manifeste du décodeur. Pour de

plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

Pour créer un modèle de véhicule à l'aide d'un modèle

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Sur la page Modèles de véhicules, choisissez Ajouter un modèle fourni.
4. Choisissez Diagnostic embarqué (OBD) II.
5. Entrez le nom de l'interface réseau OBD créée par AWS FleetWise l'IoT.
6. Choisissez Ajouter.

Création manuelle d'un modèle de véhicule

Vous pouvez ajouter des signaux à partir du catalogue de signaux ou importer des signaux en téléchargeant un ou plusieurs fichiers .dbc. Un fichier .dbc est un format de fichier pris en charge par les bases de données Controller Area Network (bus CAN).

> **Important**
>
> Vous ne pouvez pas créer un modèle de véhicule avec les signaux de données du système de vision à l'aide de la FleetWise console AWS IoT. Utilisez plutôt le AWS CLI pour créer un modèle de véhicule.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Pour créer manuellement un modèle de véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Sur la page Modèles de véhicules, choisissez Créer un modèle de véhicule, puis effectuez les opérations suivantes.

Rubriques

- Étape 1 : Configuration du modèle de véhicule

- Étape 2 : Ajouter des signaux
- Étape 3 : Importer des signaux
- (Facultatif) Étape 4 : Ajouter des attributs
- Étape 5 : Vérification et création

Étape 1 : Configuration du modèle de véhicule

Dans Informations générales, procédez comme suit.

1. Entrez un nom pour le modèle du véhicule.
2. (Facultatif) Entrez une description.
3. Choisissez Next (Suivant).

Étape 2 : Ajouter des signaux

> **Note**
>
> - Si c'est la première fois que vous utilisez l' AWS IoT FleetWise, cette étape n'est pas disponible tant que vous ne disposez pas d'un catalogue de signaux. Lorsque le premier modèle de véhicule est créé, AWS l'IoT crée FleetWise automatiquement un catalogue de signaux avec des signaux ajoutés au premier modèle de véhicule.
> - Si vous avez de l'expérience avec AWS l'IoT FleetWise, vous pouvez ajouter des signaux à votre modèle de véhicule en sélectionnant des signaux dans le catalogue de signaux ou en téléchargeant des fichiers .dbc pour importer des signaux.
> - Vous devez disposer d'au moins un signal pour créer un modèle de véhicule.

Pour ajouter des signaux

1. Choisissez un ou plusieurs signaux dans le catalogue de signaux que vous ajoutez au modèle de véhicule. Vous pouvez consulter les signaux sélectionnés dans le volet droit.

   > **Note**
   >
   > Seuls les signaux sélectionnés seront ajoutés au modèle du véhicule.

2. Choisissez Next (Suivant).

Étape 3 : Importer des signaux

> **Note**
>
> - Si c'est la première fois que vous utilisez AWS l'IoT FleetWise, vous devez télécharger au moins un fichier .dbc pour importer des signaux.
> - Si vous avez de l'expérience avec AWS l'IoT FleetWise, vous pouvez ajouter des signaux à votre modèle de véhicule en sélectionnant des signaux dans le catalogue de signaux ou en téléchargeant des fichiers .dbc pour importer des signaux.
> - Vous devez disposer d'au moins un signal pour créer un modèle de véhicule.

Pour importer des signaux

1. Choisissez Choisir des fichiers.
2. Dans la boîte de dialogue, sélectionnez le fichier .dbc contenant les signaux. Vous pouvez télécharger plusieurs fichiers .dbc.
3. AWS L'IoT FleetWise analyse vos fichiers .dbc pour récupérer des signaux.

   Dans la section Signaux, spécifiez les métadonnées suivantes pour chaque signal.

   - Nom — Le nom du signal.

     Le nom du signal doit être unique. Le nom du signal et le chemin peuvent comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).
   - Type de données — Le type de données du signal doit être l'un des suivants : INT8,,,,, BOOLEAN, FLOAT UINT8 INT16, DOUBLE UINT16, STRING INT32 UINT32 INT64 UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 BOOLEAN_ARRAY, INT16 DOUBLE_ARRAY, UINT16 STRING_ARRAY, INT32 UINT32 INT64 UNIX_TIMESTAMP_ARRAY ou UINT64 UNKNOWN.
   - Type de signal : type de signal, qui peut être un capteur ou un actionneur.
   - (Facultatif) Unité : unité scientifique du signal, telle que le km ou le Celsius.

- (Facultatif) Chemin : chemin d'accès au signal. De même JSONPath, utilisez un point (.) pour faire référence à un signal enfant. Par exemple, **`Vehicle.Engine.Light`**.

  Le nom du signal et le chemin peuvent comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement).
- (Facultatif) Min : valeur minimale du signal.
- (Facultatif) Max : valeur maximale du signal.
- (Facultatif) Description — Description du signal.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

4. Choisissez Next (Suivant).

(Facultatif) Étape 4 : Ajouter des attributs

Vous pouvez ajouter jusqu'à 100 attributs, y compris les attributs existants dans le catalogue de signaux.

Pour ajouter des attributs

1. Dans Ajouter des attributs, spécifiez les métadonnées suivantes pour chaque attribut.

   - Nom : nom de l'attribut.

     Le nom du signal doit être unique. Le nom et le chemin du signal peuvent comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement)
   - Type de données — Le type de données de l'attribut doit être l'un des suivants : INT8,,,,, BOOLEAN UINT8, FLOAT INT16, DOUBLE UINT16 INT32, STRING UINT32 INT64 UINT64, UNIX_TIMESTAMP, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 BOOLEAN_ARRAY, INT16 DOUBLE_ARRAY, UINT16 STRING_ARRAY, INT32 UINT32 INT64 UNIX_TIMESTAMP_ARRAY ou UNKNOWN UINT64
   - (Facultatif) Unité : unité scientifique de l'attribut, telle que le km ou le Celsius.
   - (Facultatif) Chemin : chemin d'accès au signal. De même JSONPath, utilisez un point (.) pour faire référence à un signal enfant. Par exemple, **`Vehicle.Engine.Light`**.

Le nom du signal et le chemin peuvent comporter jusqu'à 150 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux points) et _ (trait de soulignement)

- (Facultatif) Min : valeur minimale de l'attribut.
- (Facultatif) Max : valeur maximale de l'attribut.
- (Facultatif) Description : description de l'attribut.

  La description peut comporter jusqu'à 2 048 caractères. Caractères valides : a—z, A—Z, 0—9, : (deux-points), _ (trait de soulignement) et - (tiret).

2. Choisissez Next (Suivant).

Étape 5 : Vérification et création

Vérifiez les configurations du modèle de véhicule, puis choisissez Create.

Dupliquer un modèle de véhicule

AWS L'IoT FleetWise peut copier les configurations d'un modèle de véhicule existant pour créer un nouveau modèle. Les signaux spécifiés dans le modèle de véhicule sélectionné sont copiés sur le nouveau modèle de véhicule.

Pour dupliquer un modèle de véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Choisissez un modèle dans la liste des modèles de véhicules, puis sélectionnez Dupliquer le modèle.

Pour configurer le modèle du véhicule, suivez le Création manuelle d'un modèle de véhicule didacticiel.

L' AWS IoT peut prendre quelques minutes pour FleetWise traiter votre demande de création du modèle de véhicule. Une fois le modèle de véhicule créé avec succès, sur la page Modèles de véhicules, la colonne État indique ACTIF. Lorsque le modèle de véhicule devient actif, vous ne pouvez pas le modifier.

Création d'un modèle de véhicule (AWS CLI)

Vous pouvez utiliser l'opération CreateModelManifestAPI pour créer des modèles de véhicules (manifestes de modèles). L'exemple suivant repose sur AWS CLI.

> **Important**
>
> Vous devez disposer d'un catalogue de signaux avant de pouvoir créer un modèle de véhicule à l'aide de l'opération `CreateModelManifest` API. Pour plus d'informations sur la création d'un catalogue de signaux, consultezCréation d'un catalogue FleetWise de signaux AWS IoT.

Pour créer un modèle de véhicule, exécutez la commande suivante.

Remplacez *`vehicle-model-configuration`* par le nom du fichier .json contenant la configuration.

```
aws iotfleetwise create-model-manifest --cli-input-json file://vehicle-model-configuration.json
```

- *`vehicle-model-name`*Remplacez-le par le nom du modèle de véhicule que vous créez.
- *`signal-catalog-ARN`*Remplacez-le par le Amazon Resource Name (ARN) du catalogue de signaux.
- (Facultatif) *`description`* Remplacez-le par une description pour vous aider à identifier le modèle du véhicule.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des FleetWise signaux IoT.

```
{
    "name": "vehicle-model-name",
    "signalCatalogArn": "signal-catalog-ARN",
    "description": "description",
    "nodes": ["Vehicle.Chassis"]
}
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateModelManifest` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Mettre à jour un modèle de FleetWise véhicule AWS IoT

Vous pouvez utiliser l'opération UpdateModelManifestAPI pour mettre à jour un modèle de véhicule existant (manifestes de modèles). L'exemple suivant repose sur AWS CLI.

Pour mettre à jour un modèle de véhicule existant, exécutez la commande suivante.

Remplacez *`update-vehicle-model-configuration`* par le nom du fichier .json contenant la configuration.

```
aws iotfleetwise update-model-manifest --cli-input-json file://update-vehicle-model-configuration.json
```

- *`vehicle-model-name`*Remplacez-le par le nom du modèle de véhicule que vous mettez à jour.
- (Facultatif) Pour activer le modèle du véhicule, remplacez-le *`vehicle-model-status`* par`ACTIVE`.

> **Important**
>
> Une fois le modèle de véhicule activé, vous ne pouvez pas le modifier.

- (Facultatif) *`description`* Remplacez-le par une description mise à jour pour vous aider à identifier le modèle du véhicule.

```
{
    "name": "vehicle-model-name",
    "status": "vehicle-model-status",
    "description": "description",
    "nodesToAdd": ["Vehicle.Front.Left"],
    "nodesToRemove": ["Vehicle.Chassis.SteeringWheel"],
}
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateModelManifest` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Vérifier la mise à jour du modèle du véhicule

Vous pouvez utiliser le fonctionnement de l'ListModelManifestNodesAPI pour vérifier si un modèle de véhicule a été mis à jour. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de tous les signaux (nœuds) d'un modèle de véhicule donné, exécutez la commande suivante.

Remplacez *vehicle-model-name* par le nom du modèle de véhicule que vous êtes en train de vérifier.

```
aws iotfleetwise list-model-manifest-nodes /
                 --name vehicle-model-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListModelManifestNodes` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer un modèle de FleetWise véhicule AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour supprimer des modèles de véhicules.

> **Important**
>
> Les véhicules et les manifestes du décodeur associés au modèle du véhicule doivent d'abord être supprimés. Pour plus d'informations, consultez Supprimer un FleetWise véhicule AWS IoT et Supprimer le manifeste d'un FleetWise décodeur AWS IoT.

Supprimer un modèle de véhicule (console)

Pour supprimer un modèle de véhicule, utilisez la FleetWise console AWS IoT.

Pour supprimer un modèle de véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Sur la page Modèles de véhicules, choisissez le modèle de véhicule cible.
4. Choisissez Supprimer.
5. Dans Supprimer **`vehicle-model-name`** ? , entrez le nom du modèle de véhicule à supprimer, puis choisissez Confirmer.

Supprimer un modèle de véhicule (AWS CLI)

Vous pouvez utiliser l'opération DeleteModelManifestAPI pour supprimer un modèle de véhicule existant (manifestes de modèles). L'exemple suivant repose sur AWS CLI.

Pour supprimer un modèle de véhicule, exécutez la commande suivante.

*`model-manifest-name`*Remplacez-le par le nom du modèle de véhicule que vous souhaitez supprimer.

```
aws iotfleetwise delete-model-manifest --name model-manifest-name
```

Vérifier la suppression du modèle de véhicule

Vous pouvez utiliser le fonctionnement de l'ListModelManifestsAPI pour vérifier si un modèle de véhicule a été supprimé. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée des résumés de tous les modèles de véhicules, exécutez la commande suivante.

```
aws iotfleetwise list-model-manifests
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListModelManifests` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
{
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Obtenez des informations sur les modèles de FleetWise véhicules AWS IoT

Vous pouvez utiliser l'opération GetModelManifestAPI pour récupérer des informations sur un modèle de véhicule. L'exemple suivant utilise AWS CLI.

Pour récupérer des informations sur un modèle de véhicule, exécutez la commande suivante.

*vehicle-model*Remplacez-le par le nom du modèle de véhicule que vous souhaitez récupérer.

```
aws iotfleetwise get-model-manifest --name vehicle-model
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au modèle du véhicule peuvent ne pas être prises en compte immédiatement.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `GetModelManifest` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Gérez les AWS manifestes des FleetWise décodeurs IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les manifestes du décodeur contiennent des informations de décodage que AWS l'IoT FleetWise utilise pour transformer les données du véhicule (données binaires) en valeurs lisibles par l'homme et pour préparer vos données pour les analyses de données. L'interface réseau et les décodeurs de signaux sont les principaux composants que vous utilisez pour configurer les manifestes des décodeurs.

Interface réseau

Contient des informations sur le protocole utilisé par le réseau embarqué. AWS L'IoT FleetWise prend en charge les protocoles suivants.

Controller Area Network (bus CAN)

Protocole qui définit la manière dont les données sont communiquées entre les unités de commande électroniques (ECUs). ECUs peut être l'unité de commande du moteur, les airbags ou le système audio.

Diagnostic embarqué (OBD) II

Protocole perfectionné qui définit la manière dont les données d'autodiagnostic sont communiquées entre ECUs eux. Il fournit un certain nombre de codes de diagnostic standard (DTCs) qui aident à identifier le problème avec votre véhicule.

Intergiciel pour véhicules

L'intergiciel du véhicule défini comme un type d'interface réseau. Parmi les exemples d'intergiciels pour véhicules, citons le système d'exploitation des robots (ROS 2) et le middleware évolutif orienté service sur IP (SOME/IP).

> **Note**
>
> AWS L'IoT FleetWise prend en charge le middleware ROS 2 pour les données du système de vision.

Interfaces personnalisées

Vous pouvez également utiliser votre propre interface pour décoder les signaux au niveau de l'Edge. Cela peut vous faire gagner du temps, car vous n'avez pas besoin de créer de règles de décodage dans le cloud.

Décodeur de signal

Fournit des informations de décodage détaillées pour un signal spécifique. Chaque signal spécifié dans le modèle du véhicule doit être associé à un décodeur de signal. Si le manifeste du décodeur contient des interfaces réseau CAN, il doit contenir les signaux du décodeur CAN. Si le manifeste du décodeur contient des interfaces réseau OBD, il doit contenir des décodeurs de signaux OBD.

Le manifeste du décodeur doit contenir des décodeurs de signaux de message s'il contient également des interfaces intergicielles pour véhicules. Ou, si le manifeste du décodeur contient des interfaces de décodage personnalisées, il doit également contenir des signaux de décodage personnalisés.

Chaque manifeste du décodeur doit être associé à un modèle de véhicule. AWS L'IoT FleetWise utilise le manifeste du décodeur associé pour décoder les données des véhicules créés sur la base du modèle du véhicule.

Chaque manifeste du décodeur possède un champ d'état qui contient l'état du manifeste du décodeur. L'état peut avoir l'une des valeurs suivantes :

- `ACTIVE`— Le manifeste du décodeur est actif.
- `DRAFT`— La configuration du manifeste du décodeur n'est pas enregistrée.
- `VALIDATING`— Le manifeste du décodeur est en cours de validation pour son éligibilité. Cela s'applique uniquement aux manifestes du décodeur qui contiennent au moins un signal de données du système de vision.
- `INVALID`— Le manifeste du décodeur a échoué à la validation et ne peut pas encore être activé. Cela s'applique uniquement aux manifestes du décodeur qui contiennent au moins un signal de données du système de vision. Vous pouvez utiliser le ListDecoderManifests et GetDecoderManifest APIs pour vérifier la raison de l'échec de la validation.

> **Important**
>
> - Si vous utilisez la FleetWise console AWS IoT pour créer un manifeste de décodeur, AWS l'IoT active FleetWise automatiquement le manifeste du décodeur pour vous.
> - Si vous utilisez l'opération `CreateDecoderManifest` API pour créer un manifeste du décodeur, le manifeste du décodeur reste dans son état. `DRAFT`
> - Vous ne pouvez pas créer de véhicules à partir de modèles de véhicules associés à un manifeste de `DRAFT` décodeur. Utilisez l'opération `UpdateDecoderManifest` API pour modifier l'`ACTIVE`état du manifeste du décodeur.
> - Vous ne pouvez pas modifier les manifestes du décodeur qui sont dans `ACTIVE` cet état.

Rubriques

- Configuration AWS des interfaces FleetWise réseau IoT et des signaux du décodeur
- Création d'un manifeste de FleetWise décodeur AWS IoT
- Mettre à jour le manifeste d'un FleetWise décodeur AWS IoT
- Supprimer le manifeste d'un FleetWise décodeur AWS IoT

- Obtenez les AWS informations du manifeste du FleetWise décodeur IoT

Configuration AWS des interfaces FleetWise réseau IoT et des signaux du décodeur

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Chaque manifeste de décodeur possède au moins une interface réseau et des décodeurs de signaux couplés aux signaux spécifiés dans le modèle de véhicule associé.

Si le manifeste du décodeur contient des interfaces réseau CAN, il doit contenir des décodeurs de signaux CAN. Si le manifeste du décodeur contient des interfaces réseau OBD, il doit contenir des décodeurs de signaux OBD.

Rubriques

- Configuration des interfaces réseau
- Configuration des décodeurs de signaux

Configuration des interfaces réseau

Pour configurer une interface réseau CAN, spécifiez les informations suivantes.

- `name`— Le nom de l'interface CAN.

  Le nom de l'interface doit être unique et peut comporter de 1 à 100 caractères.
- (Facultatif) `protocolName` — Le nom du protocole.

  Valeurs valides : `CAN-FD` et `CAN`
- (Facultatif) `protocolVersion` — AWS L'IoT prend FleetWise actuellement en charge les protocoles CAN-FD et CAN 2.0b.

  Valeurs valides : `1.0` et `2.0b`

Pour configurer une interface réseau OBD, spécifiez les informations suivantes.

- `name`— Le nom de l'interface OBD.

  Le nom de l'interface doit être unique et peut comporter de 1 à 100 caractères.
- `requestMessageId`— L'ID du message demandant des données.
- (Facultatif) `dtcRequestIntervalSeconds` — Fréquence à laquelle vous devez demander des codes de diagnostic (DTCs) au véhicule en quelques secondes. Par exemple, si la valeur spécifiée est 120, le logiciel Edge collecte les données stockées DTCs toutes les 2 minutes.
- (Facultatif) `hasTransmissionEcu` — Si le véhicule est équipé d'un module de commande de transmission (TCM).

  Valeurs valides : `true` et `false`
- (Facultatif) `obdStandard` — Norme OBD prise FleetWise en charge par AWS l'IoT. AWS L'IoT prend FleetWise actuellement en charge la norme WWH-OBD (World Wide Harmonization On-Board Diagnostics) -4. ISO15765
- (Facultatif) `pidRequestIntervalSeconds` — À quelle fréquence demander l'OBD II PIDs au véhicule. Par exemple, si la valeur spécifiée est 120, le logiciel Edge Agent collecte le signal OBD II toutes PIDs les 2 minutes.
- (Facultatif) `useExtendedIds` — S'il faut utiliser l'extension IDs dans le message.

  Valeurs valides : `true` et `false`

Pour configurer l'interface réseau d'un intergiciel de véhicule, spécifiez les informations suivantes.

- `name`— Le nom de l'interface intergicielle du véhicule.

  Le nom de l'interface doit être unique et peut comporter de 1 à 100 caractères.
- `protocolName`— Le nom du protocole.

  Valeurs valides : `ROS_2`

Pour configurer une interface de décodage personnalisée, spécifiez les informations suivantes.

- `name`— Le nom du décodeur que vous utilisez pour décoder les signaux au niveau de l'Edge.

  Le nom de l'interface du décodeur peut comporter de 1 à 100 caractères.

Configuration des décodeurs de signaux

Pour configurer un décodeur de signal CAN, spécifiez les informations suivantes.

- `factor`— Le multiplicateur utilisé pour décoder le message.
- `isBigEndian`— Si l'ordre des octets du message est big-endian. S'il s'agit d'une valeur big-endian, la valeur la plus significative de la séquence est stockée en premier, à l'adresse de stockage la plus basse.
- `isSigned`— Si le message est signé. S'il est signé, le message peut représenter à la fois des nombres positifs et négatifs.
- `length`— Longueur du message en octets.
- `messageId`— L'ID du message.
- `offset`— Le décalage utilisé pour calculer la valeur du signal. Combiné au facteur, le calcul est`value = raw_value * factor + offset`.
- `startBit`— Indique l'emplacement du premier bit du message.
- (Facultatif) `name` — Le nom du signal.
- (Facultatif) `signalValueType` — Type de valeur du signal. Le type de valeur par défaut est le nombre entier.

Pour configurer un décodeur de signal OBD, spécifiez les informations suivantes.

- `byteLength`— Longueur du message en octets.
- `offset`— Le décalage utilisé pour calculer la valeur du signal. Combiné à la mise à l'échelle, le calcul est`value = raw_value * scaling + offset`.
- `pid`— Le code de diagnostic utilisé pour demander un message à un véhicule pour ce signal.
- `pidResponseLength`— La longueur du message demandé.
- `scaling`— Le multiplicateur utilisé pour décoder le message.
- `serviceMode`— Le mode de fonctionnement (service de diagnostic) indiqué dans un message.
- `startByte`— Indique le début du message.
- (Facultatif) `bitMaskLength` — Nombre de bits masqués dans un message.
- (Facultatif) `bitRightShift` — Le nombre de positions est décalé vers la droite.
- (Facultatif) `isSigned` — Indique si le message est signé. S'il est signé, le message peut représenter à la fois des nombres positifs et négatifs. Le message n'est pas signé par défaut (`false`).

- (Facultatif) `signalValueType` — Type de valeur du signal. Le type de valeur par défaut est le nombre entier.

Pour configurer un décodeur de signal de message, spécifiez les informations suivantes.

- `topicName`— Le nom du sujet du signal de message. Il correspond aux rubriques de ROS 2. Pour plus d'informations sur l'objet de message structuré, consultez StructuredMessage.
- `structuredMessage`— Le message structuré pour le signal de message. Il peut être défini avec a primitiveMessageDefinition, structuredMessageList Definition ou de structuredMessageDefinition manière récursive.

Pour configurer un signal de décodage personnalisé, spécifiez les informations suivantes.

- (Facultatif) `id` — L'identifiant du signal que vous décodez vous-même à l'aide de l'interface de votre décodeur. L'identifiant du signal peut comporter de 1 à 150 caractères. Si ce n'est pas spécifié, la `id` valeur par défaut est le `fullyQualifiedName` du signal.

Création d'un manifeste de FleetWise décodeur AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour créer un manifeste de décodeur pour votre modèle de véhicule.

Rubriques

- Création d'un manifeste de décodeur (console)
- Création d'un manifeste de décodeur ()AWS CLI

Création d'un manifeste de décodeur (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour créer un manifeste de décodeur associé à votre modèle de véhicule.

> **Important**
>
> Vous ne pouvez pas configurer les signaux de données du système de vision dans les manifestes du décodeur à l'aide de la FleetWise console AWS IoT. Utilisez plutôt le AWS CLI. Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Pour créer un manifeste de décodeur

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Choisissez le modèle de véhicule cible.
4. Sur la page récapitulative du modèle de véhicule, choisissez Create decoder manifest, puis procédez comme suit.

Rubriques

- Étape 1 : Configuration du manifeste du décodeur
- Étape 2 : Cartographier l'interface CAN
- Étape 3 : Examen et création

Étape 1 : Configuration du manifeste du décodeur

Dans Informations générales, procédez comme suit.

1. Entrez un nom unique pour le manifeste du décodeur.
2. (Facultatif) Entrez une description.
3. Choisissez Next (Suivant).

Ajouter des interfaces réseau

Chaque manifeste du décodeur doit comporter au moins une interface réseau. Vous pouvez ajouter plusieurs interfaces réseau à un manifeste de décodeur.

Pour ajouter une interface réseau

1. Téléchargez un fichier d'interface réseau. Vous pouvez télécharger un fichier .dbc pour les protocoles CAN, ou un fichier .json pour ROS 2 ou des interfaces personnalisées.
2. Entrez le nom de votre interface réseau. Si vous avez chargé une interface personnalisée, le nom est déjà fourni.

Cartographier les signaux manquants

Si certains signaux du modèle de véhicule ne comportent pas de décodeurs de signaux couplés dans les interfaces réseau téléchargées, vous pouvez créer un décodeur personnalisé par défaut qui cartographiera les signaux manquants. Ceci est facultatif car vous pouvez mapper manuellement les signaux à l'étape suivante.

Pour créer un décodeur personnalisé par défaut

1. Sélectionnez Créer un décodeur personnalisé par défaut pour les signaux manquants.
2. Choisissez Next (Suivant).

Étape 2 : Cartographier l'interface CAN

Vous pouvez mapper les signaux CAN à l'aide de décodeurs de signaux CAN. Si vous avez coché la case Créer un décodeur personnalisé par défaut pour les signaux manquants, tous les signaux auxquels il manque un signal de décodeur sont automatiquement mappés aux décodeurs de signaux personnalisés par défaut.

Pour mapper les signaux CAN

1. Dans le mappage du signal CAN, sélectionnez un décodeur de signal.
2. Choisissez Next (Suivant).

> **Note**
>
> Si vous avez ajouté un ROS 2 ou une interface personnalisée, vous pouvez vérifier les mappages avant de créer le manifeste du décodeur.

Étape 3 : Examen et création

Vérifiez les configurations du manifeste du décodeur, puis choisissez Create.

Création d'un manifeste de décodeur ()AWS CLI

Vous pouvez utiliser l'opération CreateDecoderManifestAPI pour créer des manifestes de décodeur. L'exemple suivant repose sur AWS CLI.

> **Important**
>
> Vous devez disposer d'un modèle de véhicule avant de pouvoir créer un manifeste de décodeur. Chaque manifeste du décodeur doit être associé à un modèle de véhicule. Pour de plus amples informations, veuillez consulter Création d'un modèle de FleetWise véhicule AWS IoT.

Pour créer un manifeste de décodeur, exécutez la commande suivante.

Remplacez *decoder-manifest-configuration* par le nom du fichier .json contenant la configuration.

```
aws iotfleetwise create-decoder-manifest --cli-input-json file://decoder-manifest-configuration.json
```

- *decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous créez.
- *vehicle-model-ARN*Remplacez-le par le Amazon Resource Name (ARN) du modèle de véhicule.
- (Facultatif) *description* Remplacez-le par une description pour vous aider à identifier le manifeste du décodeur.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des interfaces FleetWise réseau IoT et des signaux du décodeur.

```
{
    "name": "decoder-manifest-name",
    "modelManifestArn": "vehicle-model-arn",
    "description": "description",
    "networkInterfaces": [
        {
            "canInterface": {
                "name": "myNetworkInterface",
                "protocolName": "CAN",
                "protocolVersion": "2.0b"
            },
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_INTERFACE"
        }
    ],
    "signalDecoders": [
        {
            "canSignal": {
                "name": "Engine_Idle_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
                "offset": 0,
                "startBit": 16
            },
            "fullyQualifiedName": "Vehicle.EngineIdleTime",
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        },
        {
            "canSignal": {
                "name": "Engine_Run_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
```

```
                "offset": 0,
                "startBit": 40
            },
            "fullyQualifiedName": "Vehicle.EngineRunTime",
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        }
    ]
}
```

- *decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous créez.
- *vehicle-model-ARN*Remplacez-le par le Amazon Resource Name (ARN) du modèle de véhicule.
- (Facultatif) *description* Remplacez-le par une description pour vous aider à identifier le manifeste du décodeur.

L'ordre des nœuds de propriétés au sein d'une structure (structure) doit rester cohérent tel que défini dans le catalogue de signaux et le modèle de véhicule (manifeste du modèle). Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des interfaces FleetWise réseau IoT et des signaux du décodeur.

```
{
 "name": "decoder-manifest-name",
 "modelManifestArn": "vehicle-model-arn",
 "description": "description",
 "networkInterfaces": [{
  "canInterface": {
   "name": "myNetworkInterface",
   "protocolName": "CAN",
   "protocolVersion": "2.0b"
  },
  "interfaceId": "Qq1acaenByOB3sSM39SYm",
  "type": "CAN_INTERFACE"
 }, {
  "type": "VEHICLE_MIDDLEWARE",
  "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
  "vehicleMiddleware": {
   "name": "ROS2_test",
   "protocolName": "ROS_2"
  }
```

```
}],
"signalDecoders": [{
  "canSignal": {
   "name": "Engine_Idle_Time",
   "factor": 1,
   "isBigEndian": true,
   "isSigned": false,
   "length": 24,
   "messageId": 271343712,
   "offset": 0,
   "startBit": 16
  },
  "fullyQualifiedName": "Vehicle.EngineIdleTime",
  "interfaceId": "Qq1acaenByOB3sSM39SYm",
  "type": "CAN_SIGNAL"
 },
 {
  "canSignal": {
   "name": "Engine_Run_Time",
   "factor": 1,
   "isBigEndian": true,
   "isSigned": false,
   "length": 24,
   "messageId": 271343712,
   "offset": 0,
   "startBit": 40
  },
  "fullyQualifiedName": "Vehicle.EngineRunTime",
  "interfaceId": "Qq1acaenByOB3sSM39SYm",
  "type": "CAN_SIGNAL"
 },
 {
  "fullyQualifiedName": "Vehicle.CompressedImageTopic",
  "type": "MESSAGE_SIGNAL",
  "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
  "messageSignal": {
   "topicName": "CompressedImageTopic:sensor_msgs/msg/CompressedImage",
   "structuredMessage": {
    "structuredMessageDefinition": [{
      "fieldName": "header",
      "dataType": {
       "structuredMessageDefinition": [{
         "fieldName": "stamp",
         "dataType": {
```

```
          "structuredMessageDefinition": [{
            "fieldName": "sec",
            "dataType": {
             "primitiveMessageDefinition": {
              "ros2PrimitiveMessageDefinition": {
               "primitiveType": "INT32"
              }
             }
            }
           },
           {
            "fieldName": "nanosec",
            "dataType": {
             "primitiveMessageDefinition": {
              "ros2PrimitiveMessageDefinition": {
               "primitiveType": "UINT32"
              }
             }
            }
           }
          ]
         }
        },
        {
         "fieldName": "frame_id",
         "dataType": {
          "primitiveMessageDefinition": {
           "ros2PrimitiveMessageDefinition": {
            "primitiveType": "STRING"
           }
          }
         }
        }
       ]
      }
     },
     {
      "fieldName": "format",
      "dataType": {
       "primitiveMessageDefinition": {
        "ros2PrimitiveMessageDefinition": {
         "primitiveType": "STRING"
        }
       }
```

```
        }
       },
       {
        "fieldName": "data",
        "dataType": {
         "structuredMessageListDefinition": {
          "name": "listType",
          "memberType": {
           "primitiveMessageDefinition": {
            "ros2PrimitiveMessageDefinition": {
             "primitiveType": "UINT8"
            }
           }
          },
          "capacity": 0,
          "listType": "DYNAMIC_UNBOUNDED_CAPACITY"
         }
        }
       }
      ]
     }
    }
   }
  ]
}
```

- *decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous créez.
- *vehicle-model-ARN*Remplacez-le par le Amazon Resource Name (ARN) du modèle de véhicule.
- (Facultatif) *description* Remplacez-le par une description pour vous aider à identifier le manifeste du décodeur.

Pour plus d'informations sur la configuration des branches, des attributs, des capteurs et des actionneurs, consultezConfiguration AWS des interfaces FleetWise réseau IoT et des signaux du décodeur.

```
{
 "name": "decoder-manifest-name",
 "modelManifestArn": "vehicle-model-arn",
 "description": "description",
```

```
 "networkInterfaces": [
        {
      "interfaceId": "myCustomInterfaceId",
      "type": "CUSTOM_DECODING_INTERFACE",
           "customDecodingInterface": {
               "name": "myCustomInterface"
           }
        }
    ],
    "signalDecoders": [
        {
            "customDecodingSignal": {
                "fullyQualifiedName": "Vehicle.actuator1",
                "interfaceId": "myCustomInterfaceId",
                "type": "CUSTOM_DECODING_SIGNAL",
                "customDecodingSignal": {
                    "id": "Vehicle.actuator1"
                }
            }
        },
        {
            "customDecodingSignal": {
                "fullyQualifiedName": "Vehicle.actuator2",
                "interfaceId": "myCustomInterfaceId",
                "type": "CUSTOM_DECODING_SIGNAL",
                "customDecodingSignal": {
                    "id": "Vehicle.actuator2"
                }
            }
        }
    ]
}
```

> **Note**
>
> Vous pouvez télécharger un script de démonstration pour créer un manifeste de décodeur avec les signaux du système de vision. Pour plus d'informations, consultez le Guide du développeur de données Vision System.
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateDecoderManifest` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Mettre à jour le manifeste d'un FleetWise décodeur AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l'opération UpdateDecoderManifestAPI pour mettre à jour le manifeste d'un décodeur. Vous pouvez ajouter, supprimer et mettre à jour des interfaces réseau et des décodeurs de signaux. Vous pouvez également modifier le statut du manifeste du décodeur. L'exemple suivant repose sur AWS CLI.

Pour mettre à jour le manifeste d'un décodeur, exécutez la commande suivante.

*`decoder-manifest-name`*Remplacez-le par le nom du manifeste du décodeur que vous mettez à jour.

```
aws iotfleetwise update-decoder-manifest /
```

```
                --name decoder-manifest-name /
                --status ACTIVE
```

Si les signaux n'ont pas de règles de décodage spécifiées, vous pouvez créer des règles de décodage par défaut. Les signaux sont ajoutés à une interface décodée personnalisée avec le nom complet du signal `CustomDecodingSignal$id` défini. Pour mettre à jour le manifeste d'un décodeur avec les règles de décodage par défaut, exécutez la commande suivante.

*decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous mettez à jour.

```
aws iotfleetwise update-decoder-manifest /
                --name decoder-manifest-name /
                --status ACTIVE
                --default-for-unmapped-signals CUSTOM_DECODING
```

> **Important**
>
> Une fois que vous avez activé le manifeste du décodeur, vous ne pouvez pas le modifier.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateDecoderManifest` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
```

```
}
```

Vérifier la mise à jour du manifeste du décodeur

Vous pouvez utiliser l'opération ListDecoderManifestSignalsAPI pour vérifier si les signaux du décodeur dans le manifeste du décodeur ont été mis à jour. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de tous les signaux du décodeur (nœuds) dans un manifeste de décodeur donné, exécutez la commande suivante.

*decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous êtes en train de vérifier.

```
aws iotfleetwise list-decoder-manifest-signals /
                --name decoder-manifest-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListDecoderManifestSignals` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Vous pouvez utiliser l'opération ListDecoderManifestNetworkInterfacesAPI pour vérifier si les interfaces réseau du manifeste du décodeur ont été mises à jour. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de toutes les interfaces réseau dans un manifeste de décodeur donné, exécutez la commande suivante.

*decoder-manifest-name*Remplacez-le par le nom du manifeste du décodeur que vous êtes en train de vérifier.

```
aws iotfleetwise list-decoder-manifest-network-interfaces /
                --name decoder-manifest-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListDecoderManifestNetworkInterfaces` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer le manifeste d'un FleetWise décodeur AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour supprimer un manifeste de décodeur.

> **Important**
>
> Les véhicules associés au manifeste du décodeur doivent d'abord être supprimés. Pour de plus amples informations, veuillez consulter Supprimer un FleetWise véhicule AWS IoT.

Rubriques

- Supprimer un manifeste de décodeur (console)
- Supprimer un manifeste de décodeur ()AWS CLI

Supprimer un manifeste de décodeur (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour supprimer le manifeste d'un décodeur.

Pour supprimer un manifeste de décodeur

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Modèles de véhicules.
3. Choisissez le modèle de véhicule cible.
4. Sur la page récapitulative du modèle de véhicule, choisissez l'onglet Decoder manifest.
5. Choisissez le manifeste du décodeur cible, puis choisissez Supprimer.
6. Dans Supprimer **`decoder-manifest-name`** ? , entrez le nom du manifeste du décodeur à supprimer, puis choisissez Confirmer.

Supprimer un manifeste de décodeur ()AWS CLI

Vous pouvez utiliser l'opération DeleteDecoderManifestAPI pour supprimer un manifeste de décodeur. L'exemple suivant utilise AWS CLI.

> **Important**
>
> Avant de supprimer le manifeste du décodeur, supprimez d'abord les véhicules associés. Pour de plus amples informations, veuillez consulter Supprimer un FleetWise véhicule AWS IoT.

Pour supprimer un manifeste de décodeur, exécutez la commande suivante.

*`decoder-manifest-name`*Remplacez-le par le nom du manifeste du décodeur que vous supprimez.

```
aws iotfleetwise delete-decoder-manifest --name decoder-manifest-name
```

Vérifier la suppression du manifeste du décodeur

Vous pouvez utiliser l'opération ListDecoderManifestsAPI pour vérifier si un manifeste de décodeur a été supprimé. L'exemple suivant utilise AWS CLI.

Pour récupérer une liste paginée de résumés de tous les manifestes du décodeur, exécutez la commande suivante.

```
aws iotfleetwise list-decoder-manifests
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListDecoderManifests` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Obtenez les AWS informations du manifeste du FleetWise décodeur IoT

Vous pouvez utiliser l'opération GetDecoderManifestAPI pour vérifier si les interfaces réseau et les décodeurs de signaux du manifeste du décodeur ont été mis à jour. L'exemple suivant utilise AWS CLI.

Pour récupérer des informations sur un manifeste de décodeur, exécutez la commande suivante.

*`decoder-manifest`*Remplacez-le par le nom du manifeste du décodeur que vous souhaitez récupérer.

```
aws iotfleetwise get-decoder-manifest --name decoder-manifest
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au manifeste du décodeur peuvent ne pas être reflétées immédiatement.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `GetDecoderManifest` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Gérez les FleetWise véhicules AWS IoT

Les véhicules sont des exemples de modèles de véhicules. Les véhicules doivent être créés à partir d'un modèle de véhicule et associés à un manifeste du décodeur. Vehicles télécharge un ou plusieurs flux de données dans le cloud. Par exemple, un véhicule peut envoyer des données sur le kilométrage, la température du moteur et l'état du chauffage vers le cloud. Chaque véhicule contient les informations suivantes :

`vehicleName`

Un identifiant identifiant le véhicule.

N'ajoutez pas d'informations personnelles identifiables (PII) ou d'autres informations confidentielles ou sensibles dans le nom de votre véhicule. Les noms des véhicules sont accessibles par d'autres AWS services, notamment Amazon CloudWatch. Les noms des véhicules ne sont pas destinés à être utilisés pour des données privées ou sensibles.

`modelManifestARN`

Le nom de ressource Amazon (ARN) d'un modèle de véhicule (manifeste du modèle). Chaque véhicule est créé à partir d'un modèle de véhicule. Les véhicules créés à partir du même modèle de véhicule sont constitués du même groupe de signaux hérités du modèle de véhicule. Ces signaux sont définis et normalisés dans le catalogue de signaux.

`decoderManifestArn`

L'ARN du manifeste du décodeur. Un manifeste de décodage fournit des informations de décodage que l' AWS IoT FleetWise peut utiliser pour transformer les données de signal brutes (données binaires) en valeurs lisibles par l'homme. Un manifeste du décodeur doit être associé à un modèle de véhicule. AWS L'IoT FleetWise utilise le même manifeste de décodage pour décoder les données brutes des véhicules créés sur la base du même modèle de véhicule.

`attributes`

Les attributs sont des paires clé-valeur qui contiennent des informations statiques. Les véhicules peuvent contenir des attributs hérités du modèle du véhicule. Vous pouvez ajouter des attributs supplémentaires pour distinguer un véhicule individuel des autres véhicules créés à partir du même modèle de véhicule. Par exemple, si vous avez une voiture noire, vous pouvez spécifier la valeur suivante pour un attribut : `{"color": "black"}`

> **Important**
>
> Les attributs doivent être définis dans le modèle de véhicule associé avant de pouvoir les ajouter à des véhicules individuels.

Pour plus d'informations sur les modèles de véhicules, les manifestes des décodeurs et les attributs, consultezModélisez AWS des FleetWise véhicules IoT.

AWS L'IoT FleetWise fournit les opérations d'API suivantes que vous pouvez utiliser pour créer et gérer des véhicules.

- CreateVehicle— Crée un nouveau véhicule.
- BatchCreateVehicle— Crée un ou plusieurs nouveaux véhicules.
- UpdateVehicle— Met à jour un véhicule existant.
- BatchUpdateVehicle— Met à jour un ou plusieurs véhicules existants.
- DeleteVehicle— Supprime un véhicule existant.
- ListVehicles— Récupère une liste paginée des résumés de tous les véhicules.
- GetVehicle— Récupère des informations sur un véhicule.

Didacticiels

- Fournir AWS des FleetWise véhicules IoT
- Sujets réservés dans le domaine de AWS l'IoT FleetWise
- Créez un FleetWise véhicule AWS IoT
- Créez plusieurs FleetWise véhicules AWS IoT
- Mettre à jour un FleetWise véhicule AWS IoT
- Mettre à jour plusieurs FleetWise véhicules AWS IoT
- Supprimer un FleetWise véhicule AWS IoT
- Obtenez des informations sur les FleetWise véhicules AWS IoT

Fournir AWS des FleetWise véhicules IoT

Le FleetWise logiciel Edge Agent pour AWS l'IoT exécuté dans votre véhicule collecte et transfère les données vers le cloud. AWS L'IoT FleetWise s'intègre AWS IoT Core pour garantir une

communication sécurisée entre le logiciel Edge Agent et le cloud via MQTT. Chaque véhicule correspond à un AWS IoT objet. Vous pouvez utiliser un AWS IoT objet existant pour créer un véhicule ou configurer AWS l'IoT FleetWise pour créer automatiquement un AWS IoT objet pour votre véhicule. Pour de plus amples informations, veuillez consulter Créez un FleetWise véhicule AWS IoT.

AWS IoT Core prend en charge l'authentification et l'autorisation qui permettent de contrôler en toute sécurité l'accès aux FleetWise ressources de AWS l'IoT. Les véhicules peuvent utiliser des certificats X.509 pour s'authentifier (se connecter) afin d'utiliser l' AWS IoT FleetWise et des AWS IoT Core politiques pour obtenir l'autorisation (disposer d'autorisations) pour effectuer des actions spécifiques.

Authentifier les véhicules

Vous pouvez créer des AWS IoT Core politiques pour authentifier vos véhicules.

Pour authentifier votre véhicule

- Pour créer une AWS IoT Core politique, exécutez la commande suivante.
  - *policy-name*Remplacez-le par le nom de la politique que vous souhaitez créer.
  - *file-name*Remplacez-le par le nom du fichier JSON contenant la AWS IoT Core politique.

```
aws iot create-policy --policy-name policy-name --policy-document file://file-
name.json
```

Avant d'utiliser l'exemple de politique, procédez comme suit :

- Remplacez *region* par la AWS région dans laquelle vous avez créé FleetWise des ressources AWS IoT.
- *awsAccount*Remplacez-le par votre identifiant de AWS compte.

Cet exemple inclut les sujets réservés par AWS l'IoT FleetWise. Vous devez ajouter les sujets à la politique. Pour de plus amples informations, veuillez consulter Sujets réservés dans le domaine de AWS l'IoT FleetWise.

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

```
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:region:awsAccount:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/checkins",
                "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/signals"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:region:awsAccount:topicfilter/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/collection_schemes",
                "arn:aws:iot:region:awsAccount:topicfilter/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/decoder_manifests"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/collection_schemes",
                "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/decoder_manifests"
            ]
```

```
        }
    ]
}
```

Autoriser les véhicules

Vous pouvez créer des certificats X.509 pour autoriser vos véhicules.

Pour autoriser votre véhicule

> **Important**
>
> Nous vous recommandons de créer un nouveau certificat pour chaque véhicule.

1. Pour créer une paire de clés RSA et émettre un certificat X.509, exécutez la commande suivante.

   - Remplacez *cert* par le nom du fichier qui enregistre le contenu de sortie de commande de CertificatePEM.
   - Remplacez *public-key* par le nom du fichier qui enregistre le contenu de sortie de commande de KeyPair. PublicKey.
   - Remplacez *private-key* par le nom du fichier qui enregistre le contenu de sortie de commande de KeyPair. PrivateKey.

   ```
   aws iot create-keys-and-certificate \
       --set-as-active \
       --certificate-pem-outfile cert.pem \
       --public-key-outfile public-key.key" \
       --private-key-outfile  private-key.key"
   ```

2. Copiez le nom de ressource Amazon (ARN) du certificat à partir de la sortie.
3. Pour associer la politique au certificat, exécutez la commande suivante.

   - *policy-name*Remplacez-le par le nom de la AWS IoT Core politique que vous avez créée.
   - *certificate-arn*Remplacez-le par l'ARN du certificat que vous avez copié.

```
aws iot attach-policy \
    --policy-name policy-name\
    --target "certificate-arn"
```

4. Pour associer le certificat à l'objet, exécutez la commande suivante.

   - Remplacez *thing-name* par le nom de votre AWS IoT objet ou l'identifiant de votre véhicule.
   - *certificate-arn*Remplacez-le par l'ARN du certificat que vous avez copié.

```
aws iot attach-thing-principal \
    --thing-name thing-name \
    --principal "certificate-arn"
```

Sujets réservés dans le domaine de AWS l'IoT FleetWise

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

AWS L'IoT FleetWise se réserve l'utilisation des rubriques suivantes. Si le sujet réservé le permet, vous pouvez vous y abonner ou y publier. Toutefois, vous ne pouvez pas créer de nouveaux sujets commençant par le signe dollar ($). Si vous utilisez des opérations de publication ou d'abonnement non prises en charge avec des sujets réservés, la connexion peut être interrompue.

Rubrique	Opération du client autorisée	Description	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/checkins`	Publication	Le logiciel Edge Agent publie des informations sur l'état du véhicule dans cette rubrique.	

Rubrique	Opération du client autorisée	Description	
		Les informations relatives à l'état du véhicule sont échangées au format Protobuf (Protobuf). Pour plus d'informations, consultez le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/signals`	Publication	Le logiciel Edge Agent publie des signaux relatifs à cette rubrique. Les informations de signal sont échangées dans des tampons de protocole (Protobuf). Pour plus d'informations, consultez le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.	

Rubrique	Opération du client autorisée	Description	
`$aws/iotfleetwise/vehicles/vehicleName/collection_schemes`	S'abonner	AWS L'IoT FleetWise publie des schémas de collecte de données sur ce sujet. Les véhicules utilisent ces systèmes de collecte de données.	
`$aws/iotfleetwise/vehicles/vehicleName/decoder_manifests`	S'abonner	AWS L'IoT FleetWise publie des manifestes de décodeurs sur ce sujet. Les véhicules consomment ces manifestes du décodeur.	
`$aws/iotfleetwise/vehicles/vehicleName/command/request`	S'abonner	AWS L'IoT FleetWise publie des demandes d'exécution de commandes dans cette rubrique. Les véhicules consomment ensuite ces demandes de commande.	

Rubrique	Opération du client autorisée	Description	
`$aws/iotfleetwise/vehicles/vehicleName/command/response`	Publication	Le logiciel Edge Agent publie les réponses aux commandes du véhicule relatives à cette rubrique. Les réponses aux commandes sont échangées sous forme de tampons de protocole (Protobuf). Pour plus d'informations, consultez le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.	
`$aws/iotfleetwise/vehicles/vehicleName/command/notification`	S’abonner	AWS L'IoT FleetWise publie des mises à jour sur l'état des commandes sur cette rubrique. Les notifications sont envoyées au format JSON.	
`$aws/iotfleetwise/vehicles/$vehicle_name/last_known_states/config`	S’abonner	AWS L'IoT FleetWise publie des configurations de modèles d'état dans cette rubrique. Les véhicules utilisent ces configurations de modèles d'état.	

Rubrique	Opération du client autorisée	Description	
`$aws/iotfleetwise/vehicles/ $vehicle_name /last_known_states/data`	Publication	Le logiciel Edge Agent publie les données collectées à partir des signaux dans cette rubrique.	

Rubrique	Opération du client autorisée	Description	
`$aws/iotfleetwise/vehicles/` *`$vehicle_name`* `/last_known_state/` *`$state_template_name`* `/data`	S'abonner	AWS L'IoT FleetWise publie les données collectées à partir des signaux configurés dans la section spécifiée *`$state_template_name`* dans cette rubrique. Les mises à jour peuvent être partielles. Par exemple, si une association de modèles d'état contient plusieurs signaux avec la stratégie de mise à jour en cours de modification, seuls les signaux modifiés sont contenus dans un message donné. Les informations de signal sont échangées dans des tampons de protocole (Protobuf). Pour plus d'informations, consultez le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.	

Créez un FleetWise véhicule AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour créer un véhicule.

> **Important**
>
> Avant de commencer, vérifiez les points suivants :
>
> - Vous devez avoir un modèle de véhicule et le statut du modèle de véhicule doit être le cas`ACTIVE`. Pour de plus amples informations, veuillez consulter Gérez les modèles FleetWise de véhicules AWS IoT.
> - Le modèle de votre véhicule doit être associé à un manifeste du décodeur, et le statut du manifeste du décodeur doit être le même. `ACTIVE` Pour de plus amples informations, veuillez consulter Gérez les AWS manifestes des FleetWise décodeurs IoT.

Rubriques

- Création d'un véhicule (console)
- Création d'un véhicule (AWS CLI)

Création d'un véhicule (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour créer un véhicule.

Pour créer un véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Sur la page récapitulative du véhicule, choisissez Créer un véhicule, puis effectuez les étapes suivantes.

Rubriques

- Étape 1 : définir les propriétés du véhicule
- Étape 2 : Configuration du certificat du véhicule
- Étape 3 : associer des politiques au certificat
- Étape 4 : vérifier et créer

Étape 1 : définir les propriétés du véhicule

Au cours de cette étape, vous nommez le véhicule et vous l'associez au manifeste du modèle et au manifeste du décodeur.

1. Entrez un nom unique pour le véhicule.

   > **Important**
   >
   > Un véhicule correspond à n'importe quelle AWS IoT chose. Si un objet portant ce nom existe déjà, choisissez Associer le véhicule à un objet IoT pour le mettre à jour avec le véhicule. Vous pouvez également choisir un autre nom de véhicule et AWS l'IoT FleetWise créera automatiquement un nouvel élément pour le véhicule.

2. Choisissez un modèle de véhicule (manifeste du modèle) dans la liste.
3. Choisissez un manifeste de décodeur dans la liste. Le manifeste du décodeur est associé au modèle du véhicule.
4. (Facultatif) Pour associer des attributs de véhicule, choisissez Ajouter des attributs. Si vous ignorez cette étape, vous devez ajouter des attributs après la création du véhicule avant de pouvoir le déployer dans des campagnes.
5. (Facultatif) Pour associer des étiquettes au véhicule, choisissez Ajouter une nouvelle étiquette. Vous pouvez également ajouter des tags après la création du véhicule.
6. Choisissez Suivant.

Étape 2 : Configuration du certificat du véhicule

Pour utiliser votre véhicule en tant qu' AWS IoT objet, vous devez configurer un certificat de véhicule accompagné d'une politique. Si vous ignorez cette étape, vous devez configurer un certificat une fois le véhicule créé avant de pouvoir le déployer dans des campagnes.

1. Choisissez Générer automatiquement un nouveau certificat (recommandé).
2. Choisissez Suivant.

Étape 3 : associer des politiques au certificat

Attachez une politique au certificat que vous avez configuré à l'étape précédente.

1. Pour Politiques, entrez un nom de stratégie existant. Pour créer une nouvelle politique, choisissez Create policy.
2. Choisissez Suivant.

Étape 4 : vérifier et créer

Vérifiez les configurations du véhicule, puis choisissez Create vehicle.

> **Important**
>
> Une fois le véhicule créé, vous devez télécharger le certificat et les clés. Vous utiliserez le certificat et la clé privée pour connecter le véhicule au FleetWise logiciel Edge Agent for AWS IoT.

Création d'un véhicule (AWS CLI)

Lorsque vous créez un véhicule, vous devez utiliser un modèle de véhicule associé à un manifeste de décodeur. Vous pouvez utiliser l'opération CreateVehicleAPI pour créer un véhicule. L'exemple suivant repose sur AWS CLI.

Pour créer un véhicule, exécutez la commande suivante.

Remplacez *`file-name`* par le nom du fichier .json contenant la configuration du véhicule.

```
aws iotfleetwise create-vehicle --cli-input-json file://file-name.json
```

Example — configuration du véhicule

- (Facultatif) La `associationBehavior` valeur peut être l'une des suivantes :

- `CreateIotThing`— Lorsque votre véhicule est créé, AWS l'IoT crée FleetWise automatiquement un AWS IoT objet portant le nom de l'identifiant de votre véhicule.
- `ValidateIotThingExists`— Utilisez un AWS IoT objet existant pour créer un véhicule.

  Pour créer un AWS IoT objet, exécutez la commande suivante. *`thing-name`*Remplacez-le par le nom de l'objet que vous souhaitez créer.

  ```
  aws iot create-thing --thing-name thing-name
  ```

Si ce n'est pas spécifié, AWS l'IoT crée FleetWise automatiquement AWS IoT quelque chose pour votre véhicule.

> **Important**
>
> Assurez-vous que l' AWS IoT objet est approvisionné une fois le véhicule créé. Pour de plus amples informations, veuillez consulter Fournir AWS des FleetWise véhicules IoT.

- *`vehicle-name`*Remplacez-le par l'un des modèles suivants.
  - Le nom de votre AWS IoT objet s'il `associationBehavior` est configuré pour`ValidateIotThingExists`.
  - L'ID du véhicule pour lequel il doit être créé `associationBehavior` est configuré`CreateIotThing`.

    L'identifiant du véhicule peut comporter de 1 à 100 caractères. Caractères valides : a—z, A—Z, 0—9, tiret (-), trait de soulignement (_) et deux-points (:).
- *`model-manifest-ARN`*Remplacez-le par l'ARN de votre modèle de véhicule (manifeste du modèle).
- Remplacez *`decoder-manifest-ARN`* par l'ARN du manifeste du décodeur associé au modèle de véhicule spécifié.
- (Facultatif) Vous pouvez ajouter des attributs supplémentaires pour distinguer ce véhicule des autres véhicules créés à partir du même modèle de véhicule. Par exemple, si vous avez une voiture électrique, vous pouvez spécifier la valeur suivante pour un attribut : `{"fuelType": "electric"}`

> **Important**
>
> Les attributs doivent être définis dans le modèle de véhicule associé avant de pouvoir les ajouter à des véhicules individuels.

```
{
    "associationBehavior": "associationBehavior",
    "vehicleName": "vehicle-name",
    "modelManifestArn": "model-manifest-ARN",
    "decoderManifestArn": "decoder-manifest-ARN",
    "attributes": {
        "key": "value"
    }
}
```

Example — associe un modèle d'état au véhicule

Vous pouvez associer des modèles d'état au véhicule pour permettre la collecte de mises à jour d'état à partir du véhicule dans le cloud en utilisant le `stateTemplates` champ.

Dans cet exemple, il *`stateTemplateUpdateStrategy`* peut s'agir de l'un des suivants :

- `periodic`: vous permet de définir un taux fixe auquel le logiciel Edge Agent enverra les mises à jour des signaux au cloud (le logiciel Edge Agent enverra les mises à jour même si la valeur du signal n'a pas changé entre les mises à jour).
- `onChange`: Le logiciel Edge Agent envoie des mises à jour du signal chaque fois que le signal change.

```
aws iotfleetwise create-vehicle --cli-input-json file://create-vehicle.json
```

Où le *`create-vehicle.json`* fichier contient (par exemple) :

```
{
    "associationBehavior": "associationBehavior",
    "vehicleName": "vehicle-name",
    "modelManifestArn": "model-manifest-ARN",
    "decoderManifestArn": "decoder-manifest-ARN",
```

```
    "attributes": {
        "key": "value"
    },
    "stateTemplates": [
        {
            "identifier": "state-template-name",
            "stateTemplateUpdateStrategy": {
                "periodic": {
                    "stateTemplateUpdateRate": {
                        "unit": "SECOND",
                        "value": 10
                    }
                }
            }
        }
    ]
}
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateVehicle` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
{
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Créez plusieurs FleetWise véhicules AWS IoT

Vous pouvez utiliser l'opération BatchCreateVehicleAPI pour créer plusieurs véhicules à la fois. L'exemple suivant repose sur AWS CLI.

Pour créer plusieurs véhicules, exécutez la commande suivante.

*file-name*Remplacez-le par le nom du fichier .json qui contient les configurations de plusieurs véhicules.

```
aws iotfleetwise batch-create-vehicle --cli-input-json file://file-name.json
```

Example — configurations des véhicules

```
{
    "vehicles": [
        {
                "associationBehavior": "associationBehavior",
                "vehicleName": "vehicle-name",
                "modelManifestArn": "model-manifest-ARN",
                "decoderManifestArn": "decoder-manifest-ARN",
                "attributes": {
                    "key": "value"
                }
        },
        {
                "associationBehavior": "associationBehavior",
                "vehicleName": "vehicle-name",
                "modelManifestArn": "model-manifest-ARN",
                "decoderManifestArn": "decoder-manifest-ARN",
                "attributes": {
                    "key": "value"
                }
        }
    ]
}
```

Vous pouvez créer jusqu'à 10 véhicules pour chaque opération par lot. Pour plus d'informations sur la configuration du véhicule, voirCréez un FleetWise véhicule AWS IoT.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `BatchCreateVehicle` d'API.

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

```
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Mettre à jour un FleetWise véhicule AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l'opération UpdateVehicleAPI pour mettre à jour un véhicule existant. L'exemple suivant repose sur AWS CLI.

Pour mettre à jour un véhicule, exécutez la commande suivante.

*file-name*Remplacez-le par le nom du fichier .json contenant la configuration de votre véhicule.

```
aws iotfleetwise update-vehicle --cli-input-json file://file-name.json
```

Example — configuration du véhicule

- *vehicle-name*Remplacez-le par l'identifiant du véhicule que vous souhaitez mettre à jour.
- (Facultatif) *model-manifest-ARN* Remplacez-le par l'ARN du modèle de véhicule (manifeste du modèle) que vous utilisez pour remplacer le modèle de véhicule utilisé.
- (Facultatif) *decoder-manifest-ARN* Remplacez-le par l'ARN du manifeste de votre décodeur associé au nouveau modèle de véhicule que vous avez spécifié.
- (Facultatif) Remplacez *attribute-update-mode* par les attributs du véhicule.

- `Merge`— Fusionnez les nouveaux attributs dans les attributs existants en mettant à jour les attributs existants avec de nouvelles valeurs et en ajoutant de nouveaux attributs s'ils n'existent pas.

  Par exemple, si un véhicule possède les attributs suivants :`{"color": "black", "fuelType": "electric"}`, et que vous mettez à jour le véhicule avec les attributs suivants :`{"color": "", "fuelType": "gasoline", "model": "x"}`, le véhicule mis à jour possède les attributs suivants :`{"fuelType": "gasoline", "model": "x"}`.
- `Overwrite`— Remplacez les attributs existants par de nouveaux attributs.

  Par exemple, si un véhicule possède les attributs suivants :`{"color": "black", "fuelType": "electric"}`, et que vous mettez à jour le véhicule avec l'`{"model": "x"}`attribut, le véhicule mis à jour possède l'`{"model": "x"}`attribut.

  Cela est obligatoire si des attributs sont présents dans l'entrée.
- (Facultatif) Pour ajouter de nouveaux attributs ou mettre à jour les attributs existants avec de nouvelles valeurs, configurez`attributes`. Par exemple, si vous avez une voiture électrique, vous pouvez spécifier la valeur suivante pour un attribut : `{"fuelType": "electric"}`

  Pour supprimer des attributs, configurez `attributeUpdateMode` sur`Merge`.

  > **Important**
  >
  > Les attributs doivent être définis dans le modèle de véhicule associé avant de pouvoir les ajouter à des véhicules individuels.

```
{
        "vehicleName": "vehicle-name",
        "modelManifestArn": "model-manifest-arn",
        "decoderManifestArn": "decoder-manifest-arn",
        "attributeUpdateMode": "attribute-update-mode"
        }
}
```

Example — ajouter ou supprimer des modèles d'état associés au véhicule

Vous pouvez associer des modèles d'état supplémentaires ou supprimer des associations existantes du véhicule à l'aide des champs suivants :

- `stateTemplatesToAdd`
- `stateTemplatesToRemove`

```
aws iotfleetwise update-vehicle --cli-input-json file://update-vehicle.json
```

Où le *update-vehicle.json* fichier contient (par exemple) :

```
{
    "vehicleName": "vehicle-name",
    "modelManifestArn": "model-manifest-arn",
    "decoderManifestArn": "decoder-manifest-arn",
    "attributeUpdateMode": "attribute-update-mode",
    "stateTemplatesToAdd": [
        {
            "identifier": "state-template-name",
            "stateTemplateUpdateStrategy": {
                "onChange": {}
            }
        }
    ],
    "stateTemplatesToRemove": ["state-template-name"]
}
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateVehicle` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
```

```
  ]
}
```

Mettre à jour plusieurs FleetWise véhicules AWS IoT

Vous pouvez utiliser l'opération BatchUpdateVehicleAPI pour mettre à jour plusieurs véhicules existants en même temps. L'exemple suivant repose sur AWS CLI.

Pour mettre à jour plusieurs véhicules, exécutez la commande suivante.

*file-name*Remplacez-le par le nom du fichier .json qui contient les configurations de plusieurs véhicules.

```
aws iotfleetwise batch-update-vehicle --cli-input-json file://file-name.json
```

Example — configurations des véhicules

```
{
   "vehicles": [
      {
         "vehicleName": "vehicle-name",
         "modelManifestArn": "model-manifest-arn",
         "decoderManifestArn": "decoder-manifest-arn",
         "mergeAttributes": true,
         "attributes": {
         "key": "value"
         }
      },
      {
         "vehicleName": "vehicle-name",
         "modelManifestArn": "model-manifest-arn",
         "decoderManifestArn": "decoder-manifest-arn",
         "mergeAttributes": true,
         "attributes": {
         "key": "value"
         }
      }
   ]
}
```

Vous pouvez mettre à jour jusqu'à 10 véhicules pour chaque opération par lot. Pour plus d'informations sur la configuration de chaque véhicule, consultezMettre à jour un FleetWise véhicule AWS IoT.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `BatchUpdateVehicle` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer un FleetWise véhicule AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour supprimer des véhicules.

> **Important**
>
> Après la suppression d'un véhicule, AWS l'IoT le retire FleetWise automatiquement des flottes et des campagnes associées. Pour plus d’informations, consultez Gérez les flottes dans l'IoT AWS FleetWise et Collectez AWS des FleetWise données IoT grâce à des campagnes. Cependant, le véhicule existe toujours en tant qu'objet ou est toujours associé à un objet dans AWS IoT Core. Pour obtenir des instructions sur la suppression d'un objet, voir Supprimer un objet dans le Guide du AWS IoT Core développeur.

Supprimer un véhicule (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour supprimer un véhicule.

Pour supprimer un véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Sur la page Véhicules, sélectionnez le bouton situé à côté du véhicule que vous souhaitez supprimer.
4. Sélectionnez Delete (Supprimer).
5. Dans Supprimer **`vehicle-name`**, entrez le nom du véhicule, puis choisissez Supprimer.

Supprimer un véhicule (AWS CLI)

Vous pouvez utiliser l'opération DeleteVehicleAPI pour supprimer un véhicule. L'exemple suivant utilise AWS CLI.

Pour supprimer un véhicule, exécutez la commande suivante.

*`vehicle-name`*Remplacez-le par l'identifiant du véhicule que vous souhaitez supprimer.

```
aws iotfleetwise delete-vehicle --vehicle-name vehicle-name
```

Vérifier la suppression du véhicule

Vous pouvez utiliser le fonctionnement de l'ListVehiclesAPI pour vérifier si un véhicule a été supprimé. L'exemple suivant repose sur AWS CLI.

Pour récupérer une liste paginée des résumés de tous les véhicules, exécutez la commande suivante.

```
aws iotfleetwise list-vehicles
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListVehicles` d'API.

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Obtenez des informations sur les FleetWise véhicules AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l'opération GetVehicleAPI pour récupérer des informations sur le véhicule. L'exemple suivant repose sur AWS CLI.

Pour récupérer les métadonnées d'un véhicule, exécutez la commande suivante.

*vehicle-name*Remplacez-le par l'identifiant du véhicule que vous souhaitez récupérer.

```
aws iotfleetwise get-vehicle --vehicle-name vehicle-name
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées au véhicule peuvent ne pas être prises en compte immédiatement.

Vous pouvez utiliser l'opération GetVehicleStatusAPI pour récupérer l'état des ressources associées à un véhicule. L'exemple suivant repose sur AWS CLI.

Pour récupérer l'état des ressources associées à un véhicule, exécutez la commande suivante.

- Remplacez *vehicle-name* par l'ID du véhicule auquel les ressources sont associées.
- Remplacez *type* par le type de ressource dont vous souhaitez récupérer le statut. Les valeurs valides pour `type` sont `CAMPAIGN`, `STATE_TEMPLATE` et `DECODER`.

```
aws iotfleetwise get-vehicle-status --vehicle-name vehicle-name --type type
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer les opérations `GetVehicleStatus` d'API `GetVehicle` ou.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Gérez les flottes dans l'IoT AWS FleetWise

Une flotte représente un groupe de véhicules. Une flotte sans véhicules associés est une entité vide. Avant de pouvoir utiliser le parc pour gérer plusieurs véhicules en même temps, vous devez associer des véhicules au parc. Un véhicule peut appartenir à plusieurs flottes. Vous pouvez contrôler les données à collecter auprès d'un parc de véhicules et le moment où elles doivent être collectées en déployant une campagne. Pour de plus amples informations, veuillez consulter Collectez AWS des FleetWise données IoT grâce à des campagnes.

Une flotte contient les informations suivantes.

`fleetId`

L'identifiant de la flotte.

(Facultatif) `description`

Une description qui vous aide à trouver la flotte.

`signalCatalogArn`

Le nom de ressource Amazon (ARN) du catalogue de signaux.

AWS L'IoT FleetWise fournit les opérations d'API suivantes que vous pouvez utiliser pour créer et gérer des flottes.

- CreateFleet— Crée un groupe de véhicules contenant le même groupe de signaux.
- AssociateVehicleFleet— Associe un véhicule à une flotte.
- DisassociateVehicleFleet— Dissocie un véhicule d'une flotte.
- UpdateFleet— Met à jour la description d'une flotte existante.
- DeleteFleet— Supprime une flotte existante.
- ListFleets— Récupère une liste paginée des résumés de toutes les flottes.
- ListFleetsForVehicle— Récupère une liste paginée IDs de toutes les flottes auxquelles appartient le véhicule.
- ListVehiclesInFleet— Récupère une liste paginée des résumés de tous les véhicules d'une flotte.
- GetFleet— Récupère des informations sur une flotte.

Rubriques

- Créez une FleetWise flotte AWS IoT
- Associer un FleetWise véhicule AWS IoT à une flotte
- Dissocier un FleetWise véhicule AWS IoT d'une flotte
- Mettre à jour un FleetWise parc AWS IoT
- Supprimer un FleetWise parc AWS IoT
- Obtenez des informations sur le FleetWise parc AWS IoT

Créez une FleetWise flotte AWS IoT

Vous pouvez utiliser l'opération CreateFleetAPI pour créer un parc de véhicules. L'exemple suivant utilise AWS CLI.

> **Important**
>
> Vous devez disposer d'un catalogue de signaux avant de pouvoir créer une flotte. Pour de plus amples informations, veuillez consulter Création d'un catalogue FleetWise de signaux AWS IoT.

Pour créer une flotte, exécutez la commande suivante.

- *`fleet-id`*Remplacez-le par l'identifiant de la flotte que vous créez.

  L'identifiant de flotte doit être unique et comporter de 1 à 100 caractères. Caractères valides : lettres (A-Z et a-z), chiffres (0-9), deux-points (:), tirets (-) et traits de soulignement (_).
- (Facultatif) Remplacez *`description`* par une description.

  La description peut comporter de 1 à 2048 caractères.
- Remplacez *`signal-catalog-arn`* par l'ARN du catalogue de signaux.

```
aws iotfleetwise create-fleet \
   --fleet-id fleet-id \
   --description description \
   --signal-catalog-arn signal-catalog-arn
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateFleet` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Associer un FleetWise véhicule AWS IoT à une flotte

Vous pouvez utiliser l'opération AssociateVehicleFleetAPI pour associer un véhicule à une flotte. L'exemple suivant utilise AWS CLI.

> **Important**
>
> - Vous devez disposer d'un véhicule et d'une flotte avant de pouvoir associer un véhicule à une flotte. Pour de plus amples informations, veuillez consulter Gérez les FleetWise véhicules AWS IoT.
> - Si vous associez un véhicule à une flotte ciblée par une campagne, AWS l'IoT déploie FleetWise automatiquement la campagne sur le véhicule.

Pour associer un véhicule à une flotte, exécutez la commande suivante.

- Remplacez *`fleet-id`* par le numéro d'identification de la flotte.
- Remplacez *`vehicle-name`* par le numéro d'identification du véhicule.

```
aws iotfleetwise associate-vehicle-fleet --fleet-id fleet-id --vehicle-name vehicle-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `AssociateVehicleFleet` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Dissocier un FleetWise véhicule AWS IoT d'une flotte

Vous pouvez utiliser l'opération DisassociateVehicleFleetAPI pour dissocier un véhicule d'une flotte. L'exemple suivant utilise AWS CLI.

Pour dissocier un véhicule d'une flotte, exécutez la commande suivante.

- Remplacez *`fleet-id`* par le numéro d'identification de la flotte.
- Remplacez *`vehicle-name`* par le numéro d'identification du véhicule.

```
aws iotfleetwise disassociate-vehicle-fleet --fleet-id fleet-id --vehicle-name vehicle-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `DisassociateVehicleFleet` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Mettre à jour un FleetWise parc AWS IoT

Vous pouvez utiliser l'opération UpdateFleetAPI pour mettre à jour la description d'une flotte. L'exemple suivant utilise AWS CLI.

Pour mettre à jour une flotte, exécutez la commande suivante.

- *fleet-id*Remplacez-le par l'identifiant de la flotte que vous mettez à jour.
- Remplacez *description* par une nouvelle description.

  La description peut comporter de 1 à 2048 caractères.

```
aws iotfleetwise update-fleet --fleet-id fleet-id --description description
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateFleet` d'API.

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

```
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer un FleetWise parc AWS IoT

Vous pouvez utiliser l'opération DeleteFleetAPI pour supprimer une flotte. L'exemple suivant utilise AWS CLI.

> **Important**
>
> Avant de supprimer une flotte, assurez-vous qu'aucun véhicule n'y est associé. Pour obtenir des instructions sur la façon de dissocier un véhicule d'une flotte, reportez-vous Dissocier un FleetWise véhicule AWS IoT d'une flotte à.

Pour supprimer une flotte, exécutez la commande suivante.

*fleet-id*Remplacez-le par l'identifiant de la flotte que vous supprimez.

```
aws iotfleetwise delete-fleet --fleet-id fleet-id
```

Vérifier la suppression de la flotte

Vous pouvez utiliser l'opération ListFleetsAPI pour vérifier si une flotte a été supprimée. L'exemple suivant repose sur AWS CLI.

Pour récupérer une liste paginée des résumés de toutes les flottes, exécutez la commande suivante.

```
aws iotfleetwise list-fleets
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListFleets` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Obtenez des informations sur le FleetWise parc AWS IoT

Vous pouvez utiliser l'opération ListFleetsForVehicleAPI pour récupérer une liste paginée IDs de toutes les flottes auxquelles appartient le véhicule. L'exemple suivant repose sur AWS CLI.

Pour récupérer une liste paginée IDs de toutes les flottes auxquelles appartient le véhicule, exécutez la commande suivante.

Remplacez *vehicle-name* par le numéro d'identification du véhicule.

```
aws iotfleetwise list-fleets-for-vehicle \
            --vehicle-name vehicle-name
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListFleetsForVehicle` d'API.

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

```
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Vous pouvez utiliser l'opération ListVehiclesInFleetAPI pour récupérer une liste paginée de résumés de tous les véhicules d'une flotte. L'exemple suivant repose sur AWS CLI.

Pour récupérer une liste paginée des résumés de tous les véhicules d'une flotte, exécutez la commande suivante.

Remplacez *fleet-id* par le numéro d'identification de la flotte.

```
aws iotfleetwise list-vehicles-in-fleet \
            --fleet-id fleet-id
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `ListVehiclesInFleet` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Vous pouvez utiliser l'opération GetFleetAPI pour récupérer des informations sur la flotte. L'exemple suivant repose sur AWS CLI.

Pour récupérer les métadonnées d'une flotte, exécutez la commande suivante.

Remplacez *fleet-id* par le numéro d'identification de la flotte.

```
aws iotfleetwise get-fleet \
            --fleet-id fleet-id
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées à la flotte peuvent ne pas être prises en compte immédiatement.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `GetFleet` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Collectez AWS des FleetWise données IoT grâce à des campagnes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Une campagne est une orchestration de règles de collecte de données. Les campagnes fournissent au FleetWise logiciel Edge Agent for AWS IoT des instructions sur la manière de sélectionner, de collecter et de transférer des données vers le cloud.

Vous créez des campagnes dans le cloud. Une fois que vous ou votre équipe avez approuvé une campagne, l' AWS IoT FleetWise définit la campagne comme étant prête à être déployée, et elle sera déployée lors du prochain enregistrement du véhicule. Vous pouvez choisir de déployer une campagne sur un véhicule ou sur une flotte de véhicules. Le logiciel Edge Agent ne commence pas à collecter des données tant qu'une campagne en cours n'est pas déployée sur le véhicule.

> **Important**
>
> Les campagnes ne fonctionneront pas tant que vous n'aurez pas les éléments suivants.
>
> - Le logiciel Edge Agent est en cours d'exécution dans votre véhicule. Pour plus d'informations sur le développement, l'installation et l'utilisation du logiciel Edge Agent, procédez comme suit.
>
>   1. Ouvrez la FleetWise console AWS IoT.
>   2. Sur la page d'accueil du service, dans la FleetWise section Commencer avec AWS l'IoT, choisissez Explore Edge Agent.
>
> - Vous avez pris les dispositions nécessaires AWS IoT Core pour approvisionner votre véhicule. Pour de plus amples informations, veuillez consulter Fournir AWS des FleetWise véhicules IoT.

> **Note**
>
> Vous pouvez également Surveillez le dernier état connu de vos véhicules (et non pas les flottes) utiliser en temps quasi réel des modèles d'état qui vous permettent de diffuser des données de télémétrie avec une stratégie de mise à jour « en cours de modification » ou « périodique ». Cette fonctionnalité fournit également des fonctionnalités « à la demande » permettant d'activer ou de désactiver des modèles précédemment déployés ou de demander l'état actuel du véhicule une seule fois (récupération).
> L'accès au dernier état connu est actuellement bloqué. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Chaque campagne contient les informations suivantes.

`signalCatalogArn`

Le nom de ressource Amazon (ARN) du catalogue de signaux associé à la campagne.

(Facultatif) `tags`

Les tags sont des métadonnées qui peuvent être utilisées pour gérer la campagne. Vous pouvez attribuer le même tag à des ressources provenant de différents services pour indiquer que les ressources sont liées.

`TargetArn`

L'ARN d'un véhicule ou d'une flotte sur lequel la campagne est déployée.

`name`

Un nom unique qui permet d'identifier la campagne.

`collectionScheme`

Les schémas de collecte de données fournissent au logiciel Edge Agent des instructions sur les données à collecter ou à quel moment les collecter. AWS L'IoT prend FleetWise actuellement en charge le système de collecte basé sur les conditions et le système de collecte basé sur le temps.

- `conditionBasedCollectionScheme`— le schéma de collecte basé sur les conditions utilise une expression logique pour identifier les données à collecter. Le logiciel Edge Agent collecte des données lorsque la condition est remplie.

- `expression`— expression logique utilisée pour identifier les données à collecter. Par exemple, si l'`$variable.`myVehicle.InVehicleTemperature` > 50.0`expression est spécifiée, le logiciel Edge Agent collecte des valeurs de température supérieures à 50,0. Pour obtenir des instructions sur la façon d'écrire des expressions, consultezExpressions logiques pour les FleetWise campagnes AWS IoT.
- (Facultatif) `conditionLanguageVersion` : version du langage d'expression conditionnelle.
- (Facultatif) `minimumTriggerIntervalMs` : durée minimale entre deux événements de collecte de données, en millisecondes. Si un signal change souvent, il est possible que vous collectiez des données à un rythme plus lent.
- (Facultatif) `triggerMode` — Il peut s'agir de l'une des valeurs suivantes :
  - `RISING_EDGE`— le logiciel Edge Agent collecte les données uniquement lorsque la condition est remplie pour la première fois. Par exemple, `$variable.`myVehicle.AirBagDeployed` == true`.
  - `ALWAYS`— Le logiciel Edge Agent collecte des données chaque fois que la condition est remplie.
- `timeBasedCollectionScheme`— lorsque vous définissez un schéma de collecte basé sur le temps, spécifiez une période en millisecondes. Le logiciel Edge Agent utilise cette période pour décider de la fréquence de collecte des données. Par exemple, si la période est de 120 000 millisecondes, le logiciel Edge Agent collecte les données toutes les deux minutes.
  - `periodMs`— le délai (en millisecondes) pour décider de la fréquence de collecte des données.

(Facultatif) `compression`

Pour économiser de la bande passante sans fil et réduire le trafic réseau, vous pouvez spécifier SNAPPY pour compresser les données dans les véhicules.

Par défaut (`OFF`), le logiciel Edge Agent ne compresse pas les données.

`dataDestinationConfigs`

Choisissez la destination unique où la campagne transférera les données du véhicule. Vous pouvez envoyer les données vers une rubrique MQTT ou les stocker dans Amazon S3 ou Amazon Timestream.

MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie léger et largement adopté. Vous pouvez publier des données sur une rubrique MQTT pour créer vos propres architectures axées sur les événements à l'aide de règles. AWS IoT AWS IoT le

support pour MQTT est basé sur les spécifications MQTT v3.1.1 et MQTT v5.0, avec quelques différences. Pour plus d'informations, consultez la section Différences entre MQTT.

S3 peut être un mécanisme de stockage de données rentable qui offre des capacités de gestion des données durables et des services de données en aval. Vous pouvez utiliser S3 pour les données relatives aux comportements de conduite ou pour analyser la maintenance à long terme.

Timestream est un mécanisme de persistance des données qui peut vous aider à identifier les tendances et les modèles en temps quasi réel. Vous pouvez utiliser Timestream pour les données de séries chronologiques, par exemple pour analyser les tendances historiques en matière de vitesse ou de freinage du véhicule.

> **Note**
>
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

(Facultatif) `dataExtraDimensions`

Vous pouvez ajouter un ou plusieurs attributs afin de fournir des informations supplémentaires pour un signal.

(Facultatif) `dataPartitions`

Créez une partition de données pour stocker temporairement les données de signal sur un véhicule. Vous configurez quand et comment transférer les données vers le cloud.

- Spécifiez la manière dont l' AWS IoT FleetWise stocke les données sur un véhicule ou une flotte en définissant la taille de stockage maximale, la durée de vie minimale et l'emplacement de stockage.
- La campagne `spoolingMode` doit l'être`TO_DISK`.
- Les configurations de téléchargement incluent la définition de la version du langage de condition et de l'expression logique.

(Facultatif) `description`

Ajoutez une description pour aider à identifier l'objectif de la campagne.

(Facultatif) `diagnosticsMode`

Lorsque le mode diagnostic est configuré sur`SEND_ACTIVE_DTCS`, la campagne envoie des codes de diagnostic standard enregistrés (DTCs) qui aident à identifier le problème avec votre véhicule. Par exemple, P0097 indique que le module de commande du moteur (ECM) a déterminé

que l'entrée du capteur de température d'air d'admission (2IAT2) est inférieure à la plage normale du capteur.

Par défaut (`OFF`), le logiciel Edge Agent n'envoie pas de codes de diagnostic.

(Facultatif) `expiryTime`

Définissez la date d'expiration de votre campagne. Lorsque la campagne expire, le logiciel Edge Agent arrête de collecter les données comme indiqué dans cette campagne. Si plusieurs campagnes sont déployées sur le véhicule, le logiciel Edge Agent utilise d'autres campagnes pour collecter des données.

Valeur par défaut : `253402243200` (31 décembre 1999, 00:00:00 UTC)

(Facultatif) `postTriggerCollectionDuration`

Vous pouvez définir une durée de collecte après le déclenchement, afin que le logiciel Edge Agent continue de collecter des données pendant une période spécifiée après l'appel d'un schéma. Par exemple, si un schéma de collecte basé sur des conditions avec l'expression suivante est invoqué :`$variable.`myVehicle.Engine.RPM` > 7000.0`, le logiciel Edge Agent continue de collecter les valeurs de tours par minute (RPM) pour le moteur. Même si le régime ne dépasse les 7000 qu'une seule fois, cela peut indiquer un problème mécanique. Dans ce cas, vous souhaiterez peut-être que le logiciel Edge Agent continue à collecter des données pour aider à surveiller la maladie.

Valeur par défaut : `0`

(Facultatif) `priority`

Spécifiez un entier pour indiquer le niveau de priorité de la campagne. Les campagnes dont le nombre est inférieur sont prioritaires. Si vous déployez plusieurs campagnes sur un véhicule, les campagnes les plus prioritaires sont lancées en premier.

Valeur par défaut : `0`

(Facultatif) `signalsToCollect`

Liste des signaux à partir desquels les données sont collectées lorsque le schéma de collecte de données est invoqué.

- `name`— le nom du signal à partir duquel les données sont collectées lorsque le schéma de collecte de données est invoqué.
- `dataPartitionId`— l'ID de la partition de données à utiliser dans le signal. L'identifiant doit correspondre à l'un des identifiants IDs fournis dans`dataPartitions`. Si vous téléchargez

un signal comme condition dans votre partition de données, ces mêmes signaux doivent y être inclus`signalsToCollect`.

- (Facultatif) `maxSampleCount` : nombre maximal d'échantillons de données que le logiciel Edge Agent collecte et transfère vers le cloud lorsque le schéma de collecte de données est invoqué.
- (Facultatif) `minimumSamplingIntervalMs` : durée minimale entre deux événements de collecte d'échantillons de données, en millisecondes. Si un signal change souvent, vous pouvez utiliser ce paramètre pour collecter des données plus lentement.

  Plage valide : 0-4294967295

(Facultatif) `spoolingMode`

S'il `spoolingMode` est configuré pour`TO_DISK`, le logiciel Edge Agent stocke temporairement les données localement lorsqu'un véhicule n'est pas connecté au cloud. Une fois la connexion rétablie, les données stockées localement sont automatiquement transférées vers le cloud.

Valeur par défaut : `OFF`

(Facultatif) `startTime`

Une campagne approuvée est activée au début.

Valeur par défaut : `0`

Le statut d'une campagne peut être l'une des valeurs suivantes.

- `CREATING`— AWS FleetWise L'IoT traite votre demande de création de la campagne.
- `WAITING_FOR_APPROVAL`— Une fois qu'une campagne est créée, elle entre dans l'`WAITING_FOR_APPROVAL`état. Pour approuver la campagne, utilisez l'opération `UpdateCampaign` API. Une fois la campagne approuvée, AWS l'IoT la déploie FleetWise automatiquement sur le véhicule ou le parc cible. Pour de plus amples informations, veuillez consulter Mettre à jour une FleetWise campagne AWS IoT.
- `RUNNING` — La campagne est active.
- `SUSPENDED`— La campagne est suspendue. Pour reprendre la campagne, utilisez l'opération `UpdateCampaign` API.

AWS L'IoT FleetWise fournit les opérations d'API suivantes que vous pouvez utiliser pour créer et gérer des campagnes.

- CreateCampaign— Crée une nouvelle campagne.
- UpdateCampaign— Met à jour une campagne existante. Après la création d'une campagne, vous devez utiliser cette opération d'API pour l'approuver.
- DeleteCampaign— Supprime une campagne existante.
- ListCampaigns— Récupère une liste paginée de résumés pour toutes les campagnes.
- GetCampaign— Récupère les informations relatives à une campagne.

Didacticiels

- Créez une FleetWise campagne AWS IoT
- Mettre à jour une FleetWise campagne AWS IoT
- Supprimer une FleetWise campagne AWS IoT
- Obtenez des informations sur les FleetWise campagnes AWS IoT
- Stockage et transfert des données de campagne
- Collectez les données relatives aux codes de diagnostic à l'aide de l' AWS IoT FleetWise
- Visualisez les données des FleetWise véhicules AWS IoT

Créez une FleetWise campagne AWS IoT

Important

L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour créer des campagnes de collecte de données sur les véhicules.

Important

Pour que votre campagne fonctionne, vous devez disposer des éléments suivants :

- Le logiciel Edge Agent est en cours d'exécution dans votre véhicule. Pour plus d'informations sur le développement, l'installation et l'utilisation du logiciel Edge Agent, procédez comme suit :

1. Ouvrez la FleetWise console AWS IoT.
2. Sur la page d'accueil du service, dans la FleetWise section Commencer avec AWS l'IoT, choisissez Explore Edge Agent.

- Vous avez pris les dispositions nécessaires AWS IoT Core pour approvisionner votre véhicule. Pour de plus amples informations, veuillez consulter Fournir AWS des FleetWise véhicules IoT.

Rubriques

- Création d'une campagne (console)
- Création d'une campagne (AWS CLI)
- Expressions logiques pour les FleetWise campagnes AWS IoT

Création d'une campagne (console)

Utilisez la FleetWise console AWS IoT pour créer une campagne visant à sélectionner, collecter et transférer les données des véhicules vers le cloud.

Création d'une campagne

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Campagnes.
3. Sur la page Campagnes, choisissez Créer une campagne, puis suivez les étapes décrites dans les rubriques suivantes.

Rubriques

- Étape 1 : Configuration de la campagne
- Étape 2 : Spécifier les conditions de stockage et de téléchargement
- Étape 3 : Configuration de la destination des données
- Étape 4 : Ajouter des véhicules
- Étape 5 : Vérification et création
- Étape 6 : Déployer une campagne

> **Important**
>
> - Vous devez disposer d'un catalogue de signaux et d'un véhicule avant de créer une campagne. Pour plus d'informations, consultez Gérez les catalogues FleetWise de signaux AWS IoT et Gérez les FleetWise véhicules AWS IoT.
> - Après avoir créé une campagne, vous devez l'approuver. Pour de plus amples informations, veuillez consulter Mettre à jour une FleetWise campagne AWS IoT.

Étape 1 : Configuration de la campagne

Dans Informations générales, procédez comme suit :

1. Entrez le nom de la campagne.
2. (Facultatif) Entrez une description.

Configurez le schéma de collecte de données de la campagne. Un schéma de collecte de données fournit au logiciel Edge Agent des instructions sur les données à collecter ou à quel moment les collecter. Dans la FleetWise console AWS IoT, vous pouvez configurer un schéma de collecte de données de la manière suivante :

- Définissez manuellement le schéma de collecte de données.
- Téléchargez un fichier pour définir automatiquement le schéma de collecte de données.

Dans l'option Configuration, choisissez l'une des options suivantes :

- Pour spécifier manuellement le type de schéma de collecte de données et définir les options permettant de personnaliser le schéma, choisissez Définir le schéma de collecte de données.

  Spécifiez manuellement le type de schéma de collecte de données et définissez les options pour personnaliser le schéma.

  1. Dans la section Détails du schéma de collecte de données, choisissez le type de schéma de collecte de données que vous souhaitez utiliser pour cette campagne. Pour utiliser une expression logique afin de reconnaître les données du véhicule à collecter, choisissez Condition-based. Pour utiliser une période spécifique afin de décider de la fréquence de collecte des données du véhicule, choisissez Basé sur le temps.

2. Définissez la durée pendant laquelle la campagne collecte des données.

   > **Note**
   >
   > Par défaut, une campagne approuvée est activée immédiatement et n'a pas d'heure de fin définie. Pour éviter des frais supplémentaires, vous devez spécifier une plage horaire.

3. Si vous avez spécifié un schéma de collecte de données basé sur des conditions, vous devez définir une expression logique pour identifier les données à collecter. AWS L'IoT FleetWise utilise une expression logique pour identifier les données à collecter dans le cadre d'un schéma basé sur les conditions. L'expression doit spécifier le nom complet d'un signal en tant que variable, opérateur de comparaison et valeur de comparaison.

   Par exemple, si vous spécifiez l'`$variable.`myVehicle.InVehicleTemperature` > 50.0`expression, l' AWS IoT FleetWise collecte des valeurs de température supérieures à 50,0. Pour obtenir des instructions sur la façon d'écrire des expressions, consultezExpressions logiques pour les FleetWise campagnes AWS IoT.

   Entrez l'expression logique utilisée pour identifier les données à collecter.

4. (Facultatif) Spécifiez la version linguistique de l'expression conditionnelle. La valeur par défaut est 1.

5. (Facultatif) Spécifiez l'intervalle de déclenchement minimal, qui correspond à la plus petite durée entre deux événements de collecte de données. Par exemple, si un signal change souvent, vous souhaiterez peut-être collecter des données à un rythme plus lent.

6. Spécifiez la condition du mode déclencheur pour que le logiciel Edge Agent collecte des données. Par défaut, le FleetWise logiciel Edge Agent for AWS IoT collecte toujours des données chaque fois que la condition est remplie. Ou bien, il ne peut collecter des données que lorsque la condition est remplie pour la première fois, lors du premier déclencheur.

7. Si vous avez spécifié un schéma de collecte de données basé sur le temps, vous devez spécifier une période, en millisecondes, comprise entre 10 000 et 60 000 millisecondes. Le logiciel Edge Agent utilise cette période pour décider de la fréquence de collecte des données.

8. (Facultatif) Modifiez les options avancées du schéma.

   a. Pour économiser de la bande passante sans fil et réduire le trafic réseau en compressant les données, choisissez Snappy.

b. (Facultatif) Pour définir la durée, en millisecondes, pendant laquelle vous pouvez continuer à collecter des données après un événement de collecte de données, vous pouvez spécifier la durée de collecte après le déclenchement.

c. (Facultatif) Pour indiquer le niveau de priorité de la campagne, spécifiez la priorité de la campagne. Les campagnes dont le nombre de priorités est inférieur sont déployées en premier et sont considérées comme ayant une priorité plus élevée.

d. Le logiciel Edge Agent peut stocker temporairement des données localement lorsqu'un véhicule n'est pas connecté au cloud. Une fois la connexion rétablie, les données stockées localement sont automatiquement transférées vers le cloud. Spécifiez si vous souhaitez que l'agent Edge stocke les données localement en cas de perte de connexion.

e. (Facultatif) Pour fournir des informations supplémentaires pour un signal, ajoutez jusqu'à cinq attributs en tant que dimensions de données supplémentaires.

- Pour télécharger un fichier afin de définir le schéma de collecte de données, sélectionnez Télécharger un fichier .json depuis votre appareil local. AWS L'IoT définit FleetWise automatiquement les options que vous pouvez définir dans le fichier. Vous pouvez consulter et mettre à jour les options sélectionnées.

  Téléchargez un fichier .json contenant des informations détaillées sur le schéma de collecte de données.

  1. Pour importer des informations sur le schéma de collecte de données, choisissez Choisir des fichiers. Pour plus d'informations sur le format de fichier requis, consultez la documentation de l'CreateCampaignAPI.

     > **Note**
     >
     > AWS L'IoT prend FleetWise actuellement en charge l'extension de format de fichier .json.

  2. AWS L'IoT définit FleetWise automatiquement le schéma de collecte de données en fonction des informations contenues dans votre fichier. Passez en revue les options que AWS l'IoT a FleetWise sélectionnées pour vous. Vous pouvez mettre à jour les options, si nécessaire.

Étape 2 : Spécifier les conditions de stockage et de téléchargement

Pour choisir si le logiciel Edge Agent stockera temporairement les données localement lorsqu'un véhicule n'est pas connecté au cloud, spécifiez le mode de spoulage.

- En mode Data spooling, choisissez l'une des options suivantes :
  - Non stocké — Le logiciel Edge Agent collecte mais ne stocke pas temporairement les données localement lorsqu'un véhicule est hors ligne. Le logiciel Edge Agent transfère les données vers le cloud lorsque le véhicule se reconnecte.
  - Stocké sur disque — Le logiciel Edge Agent collecte et stocke temporairement les données localement lorsqu'un véhicule est hors ligne. Les données collectées sont temporairement stockées à un emplacement défini dans la section « persistance » du fichier de configuration de l'agent Edge. L'agent Edge transfère les données vers le cloud lorsque le véhicule se reconnecte.
  - Stocké sur disque avec des partitions — Le véhicule stocke toujours temporairement les données sur l'Edge dans la partition de données que vous avez spécifiée. Vous pouvez choisir à quel moment vous souhaitez transférer vos données stockées vers le cloud.
    1. (Facultatif) Entrez un ID de partition pour désigner un ensemble de données particulier.
    2. Entrez un nom de dossier comme emplacement où les données seront stockées. Le chemin absolu de l'emplacement de stockage est`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.
    3. Entrez la taille de stockage maximale des données stockées dans la partition. Les données les plus récentes remplacent les anciennes lorsque la partition atteint la taille maximale.
    4. Entrez la durée minimale pendant laquelle les données de cette partition seront conservées sur le disque.
    5. (Facultatif) Entrez les conditions de téléchargement pour la partition.

Spécifier les signaux

Vous pouvez définir les signaux à partir desquels collecter les données pendant la campagne.

Pour spécifier les signaux à partir desquels collecter des données

1. Sélectionnez le nom du signal.
2. (Facultatif) Pour Nombre maximum d'échantillons, entrez le nombre maximum d'échantillons de données que le logiciel Edge Agent collecte et transfère vers le cloud pendant la campagne.
3. (Facultatif) Pour Intervalle d'échantillonnage minimal, entrez la durée minimale entre deux événements de collecte d'échantillons de données, en millisecondes. Si un signal change souvent, vous pouvez utiliser ce paramètre pour collecter des données plus lentement.

4. Pour ajouter un autre signal, choisissez Ajouter d'autres signaux. Vous pouvez ajouter jusqu'à 999 signaux.
5. Choisissez Suivant.

Étape 3 : Configuration de la destination des données

> **Note**
>
> Si la campagne contient des signaux de données du système de vision, vous ne pouvez stocker les données du véhicule que dans Amazon S3. Vous ne pouvez pas le stocker dans Timestream ou l'envoyer à un sujet MQTT.
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

Choisissez la destination vers laquelle vous souhaitez envoyer ou stocker les données collectées par la campagne. Vous pouvez envoyer les données du véhicule vers une rubrique MQTT ou les stocker dans Amazon S3 ou Amazon Timestream.

Dans les paramètres de destination, procédez comme suit :

- Choisissez le sujet Amazon S3, Amazon Timestream ou MQTT dans la liste déroulante.

Amazon S3

> **Important**
>
> Vous ne pouvez transférer des données vers S3 que si AWS l'IoT FleetWise est autorisé à écrire dans le compartiment S3. Pour plus d'informations sur l'octroi d'accès, consultez la section Contrôle de l'accès avec AWS l'IoT FleetWise.

Pour stocker les données du véhicule dans un compartiment S3, choisissez Amazon S3. S3 est un service de stockage d'objets qui stocke les données sous forme d'objets dans des compartiments. Pour plus d'informations, consultez la section Création, configuration et utilisation des compartiments Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.

S3 optimise le coût du stockage des données et fournit des mécanismes supplémentaires pour utiliser les données des véhicules, tels que les lacs de données, le stockage centralisé des données, les pipelines de traitement des données et les analyses. Vous pouvez utiliser S3 pour stocker des données à des fins de traitement et d'analyse par lots. Par exemple, vous pouvez créer des rapports sur les événements de freinage brusque pour votre modèle d'apprentissage automatique (ML). Les données entrantes du véhicule sont mises en mémoire tampon pendant 10 minutes avant la livraison.

Dans les paramètres de destination S3, procédez comme suit :

1. Pour le compartiment S3, choisissez un AWS IoT FleetWise compartiment autorisé à.
2. (Facultatif) Entrez un préfixe personnalisé que vous pouvez utiliser pour organiser les données stockées dans le compartiment S3.
3. Choisissez le format de sortie, c'est-à-dire le format des fichiers enregistrés dans le compartiment S3.
4. Choisissez si vous souhaitez compresser les données stockées dans le compartiment S3 sous forme de fichier .gzip. Nous recommandons de compresser les données car cela permet de minimiser les coûts de stockage.
5. Les options que vous sélectionnez dans les paramètres de destination S3 modifient l'exemple d'URI de l'objet S3. Il s'agit d'un exemple du format sous lequel les fichiers sont enregistrés dans S3.

Amazon Timestream

> **Important**
>
> Vous ne pouvez transférer des données vers une table que si AWS l'IoT FleetWise est autorisé à écrire des données dans Timestream. Pour plus d'informations sur l'octroi d'accès, consultez la section Contrôle de l'accès avec AWS l'IoT FleetWise.
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

Pour stocker les données du véhicule dans un tableau Timestream, choisissez Amazon Timestream. Vous pouvez utiliser Timestream pour interroger les données du véhicule afin d'identifier les tendances et les modèles. Par exemple, vous pouvez utiliser Timestream pour créer une alarme indiquant le niveau de carburant du véhicule. Les données entrantes du véhicule sont transférées vers Timestream quasiment en temps réel. Pour plus d'informations, consultez Qu'est-ce qu'Amazon Timestream ? dans le guide du développeur Amazon Timestream.

Dans les paramètres du tableau Timestream, procédez comme suit :

1. Pour le nom de la base de données Timestream, choisissez le nom de votre base de données Timestream dans la liste déroulante.
2. Pour le nom de la table Timestream, choisissez le nom de votre table Timestream dans la liste déroulante.

Dans Accès au service pour Timestream, procédez comme suit :

- Choisissez un rôle IAM dans la liste déroulante.

Rubrique MQTT

> **Important**
>
> Vous ne pouvez acheminer des données vers un sujet MQTT que si AWS l'IoT FleetWise dispose d'autorisations AWS IoT sur les sujets. Pour plus d'informations sur l'octroi d'accès, consultez la section Contrôle de l'accès avec AWS l'IoT FleetWise.

Pour envoyer les données du véhicule à un sujet MQTT, choisissez un sujet MQTT.

Les données du véhicule envoyées par messagerie MQTT sont transmises en temps quasi réel et vous permettent d'utiliser des règles pour prendre des mesures ou d'acheminer des données vers d'autres destinations. Pour plus d'informations sur l'utilisation de MQTT, consultez la section Protocoles et règles de communication des appareils AWS IoT dans le guide du AWS IoT Core développeur.

1. Sous rubrique MQTT, entrez le nom de la rubrique.
2. Dans la rubrique Accès aux services pour MQTT, indiquez si vous souhaitez laisser AWS IoT FleetWise créer et utiliser un nouveau rôle de service pour vous. Si vous souhaitez utiliser un rôle de service existant, sélectionnez-le dans la liste déroulante sous Sélectionner un rôle.

- Choisissez Suivant.

Étape 4 : Ajouter des véhicules

Pour choisir les véhicules sur lesquels déployer votre campagne, sélectionnez-les dans la liste des véhicules. Filtrez les véhicules en recherchant les attributs et leurs valeurs que vous avez ajoutés lors de la création des véhicules, ou par nom de véhicule.

Dans Filtrer les véhicules, procédez comme suit :

1. Dans le champ de recherche, trouvez l'attribut ou le nom du véhicule et choisissez-le dans la liste.

   > **Note**
   >
   > Chaque attribut ne peut être utilisé qu'une seule fois.

2. Entrez la valeur de l'attribut ou le nom du véhicule sur lequel vous souhaitez déployer la campagne. Par exemple, si le nom complet de l'attribut est`fuelType`, entrez `gasoline` sa valeur.
3. Pour rechercher un autre attribut de véhicule, répétez les étapes précédentes. Vous pouvez rechercher jusqu'à cinq attributs de véhicules et un nombre illimité de noms de véhicules.
4. Les véhicules correspondant à votre recherche sont répertoriés sous Nom du véhicule. Choisissez les véhicules sur lesquels vous souhaitez déployer la campagne.

   > **Note**
   >
   > Jusqu'à 100 véhicules sont affichés dans les résultats de recherche. Choisissez Tout sélectionner pour ajouter tous les véhicules à la campagne.

5. Choisissez Suivant.

Étape 5 : Vérification et création

Vérifiez les configurations de la campagne, puis choisissez Créer une campagne.

> **Note**
>
> Après la création d'une campagne, vous ou votre équipe devez la déployer sur les véhicules.

Étape 6 : Déployer une campagne

Après avoir créé une campagne, vous ou votre équipe devez la déployer sur les véhicules.

Pour déployer une campagne

1. Sur la page de résumé de la campagne, choisissez Déployer.
2. Vérifiez et confirmez que vous souhaitez démarrer le déploiement et commencer à collecter des données auprès des véhicules connectés à la campagne.
3. Choisissez Déployer.

Si vous souhaitez suspendre la collecte de données auprès des véhicules connectés à la campagne, sur la page récapitulative de la campagne, choisissez Suspendre. Pour reprendre la collecte de données auprès des véhicules connectés à la campagne, sélectionnez Reprendre.

Création d'une campagne (AWS CLI)

Vous pouvez utiliser l'opération CreateCampaignAPI pour créer une campagne. L'exemple suivant repose sur AWS CLI.

Lorsque vous créez une campagne, les données collectées auprès des véhicules peuvent être envoyées à un sujet MQTT ou stockées dans Amazon S3 (S3) ou Amazon Timestream. Choisissez Timestream pour une base de données de séries chronologiques rapide, évolutive et sans serveur, par exemple pour stocker des données nécessitant un traitement en temps quasi réel. Choisissez S3 pour le stockage d'objets avec une évolutivité, une disponibilité des données, une sécurité et des performances de pointe. Choisissez MQTT pour fournir des données en temps quasi réel et pour utiliser des règles AWS IoT pour effectuer des actions que vous définissez ou acheminez les données vers d'autres destinations.

> **Important**
>
> Vous ne pouvez transférer les données du véhicule vers une rubrique MQTT, Amazon S3 ou Amazon AWS Timestream que FleetWise si l'IoT est autorisé à envoyer des messages MQTT en votre nom ou à écrire des données dans S3 ou Timestream. Pour plus d'informations sur l'octroi d'accès, consultez la section Contrôle de l'accès avec AWS l'IoT FleetWise.
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

Créer une campagne

> **Important**
>
> - Vous devez disposer d'un catalogue de signaux et d'un véhicule ou d'une flotte avant de créer une campagne. Pour plus d'informations, consultez Gérez les catalogues FleetWise de signaux AWS IoT, Gérez les FleetWise véhicules AWS IoT et Gérez les flottes dans l'IoT AWS FleetWise.
> - Après la création d'une campagne, vous devez utiliser l'opération `UpdateCampaign` API pour l'approuver. Pour de plus amples informations, consultez Mettre à jour une FleetWise campagne AWS IoT.

Pour créer une campagne, exécutez la commande suivante.

*`file-name`*Remplacez-le par le nom du fichier .json qui contient la configuration de la campagne.

```
aws iotfleetwise create-campaign --cli-input-json file://file-name.json
```

- *`campaign-name`*Remplacez-le par le nom de la campagne que vous créez.
- *`signal-catalog-arn`*Remplacez-le par le Amazon Resource Name (ARN) du catalogue de signaux.
- *`target-arn`*Remplacez-le par l'ARN d'une flotte ou d'un véhicule que vous avez créé.
- *`bucket-arn`*Remplacez-le par l'ARN du compartiment S3.

```
{
    "name": "campaign-name",
    "targetArn": "target-arn",
    "signalCatalogArn": "signal-catalog-arn",
    "collectionScheme": {
        "conditionBasedCollectionScheme": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
            "minimumTriggerIntervalMs": 1000,
            "triggerMode": "ALWAYS"
        }
```

```
    },
    "compression": "SNAPPY",
    "diagnosticsMode": "OFF",
    "postTriggerCollectionDuration": 1000,
    "priority": 0,
    "signalsToCollect": [
        {
         "maxSampleCount": 100,
         "minimumSamplingIntervalMs": 0,
         "name": "Vehicle.DemoEngineTorque"
        },
        {
         "maxSampleCount": 100,
         "minimumSamplingIntervalMs": 0,
         "name": "Vehicle.DemoBrakePedalPressure"
        }
    ],
    "spoolingMode": "TO_DISK",
    "dataDestinationConfigs": [
        {
         "s3Config": {
             "bucketArn": "bucket-arn",
             "dataFormat": "PARQUET",
             "prefix": "campaign-name",
              "storageCompressionFormat": "GZIP"
      }
    }
  ],
     "dataPartitions": [
      { ...  }
  ]
}
```

> **Note**
>
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

- *campaign-name*Remplacez-le par le nom de la campagne que vous créez.
- Remplacez *signal-catalog-arn* par l'ARN du catalogue de signaux.
- *target-arn*Remplacez-le par l'ARN d'une flotte ou d'un véhicule que vous avez créé.

- *role-arn*Remplacez-le par l'ARN du rôle d'exécution des tâches qui FleetWise autorise AWS l'IoT à fournir des données à la table Timestream.
- Remplacez *table-arn* par l'ARN de la table Timestream.

```
{
  "name": "campaign-name",
  "targetArn": "target-arn",
  "signalCatalogArn": "signal-catalog-arn",
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
    {
      "timestreamConfig": {
        "executionRoleArn": "role-arn",
        "timestreamTableArn": "table-arn"
      }
    }
  ],
   "dataPartitions": [
```

```
      { ...  }
  ]
}
```

- *campaign-name*Remplacez-le par le nom de la campagne que vous créez.
- *signal-catalog-arn*Remplacez-le par le Amazon Resource Name (ARN) du catalogue de signaux.
- *target-arn*Remplacez-le par l'ARN d'une flotte ou d'un véhicule que vous avez créé.
- *topic-arn*Remplacez-le par l'ARN de la rubrique MQTT que vous avez spécifiée comme destination pour les messages contenant des données du véhicule.
- *role-arn*Remplacez-le par l'ARN du rôle d'exécution des tâches qui accorde à AWS l'IoT FleetWise l'autorisation d'envoyer, de recevoir et d'agir sur des messages relatifs au sujet MQTT que vous avez spécifié.

```
{
  "name": "campaign-name",
  "targetArn": "target-arn",
  "signalCatalogArn": "signal-catalog-arn",
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
```

```
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
      {
          "mqttTopicConfig": {
              "mqttTopicArn": "topic-arn",
              "executionRoleArn": "role-arn"
          }
      }
  ]
}
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `CreateCampaign` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Expressions logiques pour les FleetWise campagnes AWS IoT

AWS L'IoT FleetWise utilise une expression logique pour identifier les données à collecter dans le cadre d'une campagne. Pour plus d'informations sur les expressions, consultez la section Expressions du manuel du AWS IoT Events développeur.

La variable d'expression doit être construite conformément aux règles relatives au type de données collectées. Pour les données du système de télémétrie, la variable d'expression doit être le nom

complet du signal. Pour les données du système de vision, l'expression combine le nom complet du signal avec le chemin menant du type de données du signal à l'une de ses propriétés.

Par exemple, si le catalogue de signaux contient les nœuds suivants :

```
{
    myVehicle.ADAS.Camera:
    type: sensor
    datatype: Vehicle.ADAS.CameraStruct
    description: "A camera sensor"

    myVehicle.ADAS.CameraStruct:
    type: struct
    description: "An obstacle detection camera output struct"
}
```

Si les nœuds suivent la définition ROS 2 :

```
{
    Vehicle.ADAS.CameraStruct.msg:
    boolean obstaclesExists
    uint8[] image
    Obstacle[30] obstacles
}
{
    Vehicle.ADAS.Obstacle.msg:
    float32: probability
    uint8 o_type
    float32: distance
}
```

Toutes les variables d'expression d'événement possibles sont les suivantes :

```
{
...
    $variable.`myVehicle.ADAS.Camera.obstaclesExists`
    $variable.`myVehicle.ADAS.Camera.Obstacle[0].probability`
    $variable.`myVehicle.ADAS.Camera.Obstacle[1].probability`
...
    $variable.`myVehicle.ADAS.Camera.Obstacle[29].probability`
    $variable.`myVehicle.ADAS.Camera.Obstacle[0].o_type`
    $variable.`myVehicle.ADAS.Camera.Obstacle[1].o_type`
```

```
...
    $variable.`myVehicle.ADAS.Camera.Obstacle[29].o_type`
    $variable.`myVehicle.ADAS.Camera.Obstacle[0].distance`
    $variable.`myVehicle.ADAS.Camera.Obstacle[1].distance`
...
    $variable.`myVehicle.ADAS.Camera.Obstacle[29].distance`
}
```

Mettre à jour une FleetWise campagne AWS IoT

Vous pouvez utiliser l'opération UpdateCampaignAPI pour mettre à jour une campagne existante. La commande suivante utilise AWS CLI.

- *campaign-name*Remplacez-le par le nom de la campagne que vous mettez à jour.
- Remplacez *action*par l'une des informations suivantes :
  - APPROVE— Approuve la campagne visant à permettre FleetWise à AWS l'IoT de le déployer sur un véhicule ou une flotte.
  - SUSPEND— Suspend la campagne. La campagne est supprimée des véhicules et tous les véhicules de la campagne suspendue cesseront d'envoyer des données.
  - RESUME— Réactive la SUSPEND campagne. La campagne devrait être redéployée sur tous les véhicules lors du prochain enregistrement et les véhicules recommenceront à envoyer des données.
  - UPDATE— Met à jour la campagne en définissant des attributs et en les associant à la campagne.
- Remplacez *description* par une nouvelle description.

  La description peut comporter jusqu'à 2 048 caractères.
- *data-extra-dimensions*Remplacez-le par les attributs du véhicule spécifiés pour enrichir les données collectées au cours de la campagne. Par exemple, vous pouvez ajouter la marque et le modèle du véhicule à la campagne, et l' AWS IoT FleetWise associera les données à ces attributs sous forme de dimensions dans Amazon Timestream. Vous pouvez ensuite interroger les données par rapport à la marque et au modèle du véhicule.

```
aws iotfleetwise update-campaign \
            --name campaign-name \
            --action action \
```

```
            --description description \
            --data-extra-dimensions data-extra-dimensions
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `UpdateCampaign` d'API.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Supprimer une FleetWise campagne AWS IoT

Vous pouvez utiliser la FleetWise console ou l'API AWS IoT pour supprimer des campagnes.

Supprimer une campagne (console)

Pour supprimer une campagne, utilisez la FleetWise console AWS IoT.

Pour supprimer une campagne

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Campagnes.
3. Sur la page Campagnes, choisissez la campagne cible.
4. Sélectionnez Delete (Supprimer).
5. Dans Supprimer **`campaign-name`** ? , entrez le nom de la campagne à supprimer, puis choisissez Confirmer.

Supprimer une campagne (AWS CLI)

Vous pouvez utiliser l'opération DeleteCampaignAPI pour supprimer une campagne. L'exemple suivant utilise AWS CLI.

Pour supprimer une campagne, exécutez la commande suivante.

Remplacez *campaign-name* par le nom du véhicule que vous souhaitez supprimer.

```
aws iotfleetwise delete-campaign --name campaign-name
```

> **Les partitions de données supprimées ne sont pas récupérables**
>
> La suppression d'une campagne entraîne la suppression de toutes les données des appareils, et les données d'une partition ne seront pas téléchargées vers le cloud.

Vérifier la suppression de la campagne

Vous pouvez utiliser l'opération ListCampaignsAPI pour vérifier si une campagne a été supprimée. L'exemple suivant repose sur AWS CLI.

Pour récupérer une liste paginée des résumés de toutes les campagnes, exécutez la commande suivante.

```
aws iotfleetwise list-campaigns
```

Obtenez des informations sur les FleetWise campagnes AWS IoT

Vous pouvez utiliser l'opération GetCampaignAPI pour récupérer des informations sur le véhicule. L'exemple suivant repose sur AWS CLI.

Pour récupérer les métadonnées d'une campagne, exécutez la commande suivante.

*campaign-name*Remplacez-le par le nom de la campagne que vous souhaitez récupérer.

```
aws iotfleetwise get-campaign --name campaign-name
```

> **Note**
>
> Cette opération est cohérente à terme. En d'autres termes, les modifications apportées à la campagne peuvent ne pas être prises en compte immédiatement.

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer l'opération `GetCampaign` d'API.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Stockage et transfert des données de campagne

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Utilisez les partitions de données au sein des campagnes pour stocker temporairement les données de signal sur l'Edge pour les véhicules et les flottes. En configurant les options de téléchargement et de stockage pour les partitions de données, vous pouvez optimiser les conditions idéales pour le transfert de données vers les destinations de données que vous avez désignées (comme un compartiment Amazon S3). Par exemple, vous pouvez configurer la partition de données pour stocker les données d'un véhicule jusqu'à ce qu'il se connecte au Wi-Fi. Ensuite, une fois le véhicule

connecté, la campagne déclenche l'envoi des données de cette partition vers le cloud. Vous pouvez également collecter des données à l'aide de AWS IoT Jobs.

Rubriques

- Création de partitions de données
- Télécharger les données de campagne
- Téléchargez des données à l'aide de AWS IoT Jobs

Création de partitions de données

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Une partition de données dans une campagne stocke temporairement les données de signal. Vous configurez quand et comment transférer les données vers le cloud.

Une partition de données fonctionne en désignant d'abord un ensemble de données particulier à l'aide du nom `dataPartitionId` pour une campagne. Vous pouvez ensuite définir plus en détail les options de stockage de partition, telles que la taille maximale, la durée minimale de conservation de la partition de données (sur disque) et l'emplacement de stockage des données sur le périphérique. Vous pouvez déterminer l'emplacement de stockage sur le véhicule à l'aide `destorageLocation`. L'emplacement de stockage détermine le nom du dossier pour la partition de données située sous le dossier de stockage des campagnes. Le dossier de stockage des campagnes se trouve sous un dossier nommé d'après le nom du véhicule sous un chemin de persistance défini dans le fichier de configuration Edge. Il s'agit du chemin absolu de l'emplacement de stockage :`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.

Le mode spouling défini sur `TO_DISK` indique que les données partitionnées doivent être enregistrées sur un disque du véhicule. Le stockage des données pour les partitions de données fonctionne sur une base FIFO (premier entré, premier sorti). Si vous supprimez une campagne, vous supprimez également les données de la partition de données associée. Si vous ne spécifiez pas de partition de données pour les cas d'utilisation de la connectivité activée/désactivée, AWS l'IoT

stocke FleetWise toujours les données dans une mémoire tampon sur le véhicule en l'absence de connectivité. Lorsque la connectivité reprend, AWS l'IoT FleetWise télécharge les données vers le cloud. Ce comportement est configurable dans le FleetWise logiciel Edge Agent for AWS IoT.

> **Important**
>
> Si votre partition de données dépasse la limite de stockage maximale définie, les données les plus récentes remplacent les anciennes lorsque la partition atteint la taille maximale. Les données perdues sur le Edge ne sont pas récupérables. La taille de stockage est déterminée par votre limite de stockage Edge.
>
> Lorsque les données sont téléchargées dans le cloud, elles peuvent être supprimées après le délai minimum de mise en ligne. Définissez la durée minimale de vie de manière appropriée afin d'éviter toute suppression involontaire.

Les options de téléchargement déterminent les expressions des variables et le langage des conditions. Si des options de téléchargement sont spécifiées, vous devez également spécifier des options de stockage. Vous pouvez également demander que les signaux des partitions de données soient téléchargés dans le cloud. Pour de plus amples informations, veuillez consulter Télécharger les données de campagne.

Une fois les conditions de partition des données définies, `signalsToCollect` permet de spécifier les signaux à prendre en compte dans la partition de données. Vous pouvez soit spécifier IDs des partitions de données, soit configurer le `dataPartitionId` `default` pour qu'il utilise une partition de données par défaut établie. Un signal sans valeur spécifiée `dataPartitionId` sera associé à la valeur par défaut`dataPartition`.

Pour créer une partition de données

À l'aide de l'exemple suivant, créez une campagne avec une condition de stockage de partition de données. Cet exemple de campagne est configuré pour stocker les données du véhicule dans Amazon Timestream.

1. *`campaign-name`*Remplacez-le par le nom de la campagne que vous créez.
2. (Facultatif) Fournissez une description.
3. *`role-arn`*Remplacez-le par le Amazon Resource Name (ARN) du rôle d'exécution des tâches qui FleetWise autorise AWS l'IoT à fournir des données à la table Timestream.
4. Remplacez *`table-arn`* par l'ARN de la table Timestream.

5. Remplacez *signal-catalog-arn* par l'ARN du catalogue de signaux.
6. Remplacez à la *data-partition-id* fois l'`dataPartitions`ID et l'ID à associer`signalsToCollect`. Remplacez d'abord l'ID de la partition de données à utiliser dans le signal. En `signalsToCollect` effet, l'identifiant doit correspondre à l'un des identifiants IDs fournis dans`dataPartitions`.

   > **Note**
   >
   > Établissez une partition de données par défaut pour une campagne en l'utilisant `default` comme identifiant.

7. *target-arn*Remplacez-le par l'ARN d'une flotte ou d'un véhicule que vous avez créé.

```
{
    "name": "campaign-name",
    "description": "Measurement of SOC, SOH, thermal, and power optimization for Fleet
 2704",
    "targetArn": "target-arn",
    "collectionScheme": {
        "conditionBasedCollectionScheme": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS` > 50",
            "minimumTriggerIntervalMs": 1000,
            "triggerMode": "ALWAYS"
        }
    },
    "compression": "SNAPPY",
    "dataDestinationConfigs": [{
        "timestreamConfig": {
            "executionRoleArn": "role-arn",
            "timestreamTableArn": "table-arn"
        }
    }],
    "dataPartitions": [{
        "id": "data-partition-id",
        "storageOptions": {
            "maximumSize": {
                "unit": "GB",
                "value": 1024
            },
            "minimumTimeToLive": {
```

```
                "unit": "WEEKS",
                "value": 6
            },
            "storageLocation": "string"
        },
        "uploadOptions": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS.PowerOptimization` > 90"
        }
    }],
    "signalCatalogArn": "signal-catalog-arn",
    "signalsToCollect": [{
        "dataPartitionId": "data-partition-id",
        "maxSampleCount": 50000,
        "minimumSamplingIntervalMs": 100,
        "name": "Below-90-percent"
    }],
    "spoolingMode": "TO_DISK",
    "tags": [{
        "Key": "BMS",
        "Value": "Under-90"
    }]
}
```

Une fois toutes les conditions spécifiées remplies, les données partitionnées sont transmises au cloud, ce qui permet de collecter et de stocker de nouveaux signaux partitionnés.

Vous allez ensuite appeler l'`UpdateCampaign`API pour la déployer sur le FleetWise logiciel Edge Agent for AWS IoT. Pour de plus amples informations, veuillez consulter Télécharger les données de campagne.

Télécharger les données de campagne

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Il existe deux manières de télécharger des données de campagne sur l'Edge :

- Les campagnes qui répondent à vos conditions de téléchargement téléchargeront automatiquement les données dans le cloud une fois approuvées. Pour approuver une campagne, utilisez l'opération `updateCampaign` API.
- Grâce à AWS IoT Jobs, vous pouvez forcer le chargement des données même lorsque les conditions spécifiées ne sont pas remplies. Pour de plus amples informations, veuillez consulter Téléchargez des données à l'aide de AWS IoT Jobs.

Pour télécharger les données de campagne à l'aide de l'opération **`UpdateCampaign`** API

Après avoir créé la campagne, le statut de la campagne s'affiche `WAITING_FOR_APPROVAL` jusqu'à ce que vous le `action` remplaciez par`APPROVED`.

- Utilisez l'exemple suivant pour mettre à jour la campagne `action` en appelant l'opération UpdateCampaignAPI.

```
{
   "action": "APPROVED",
   "dataExtraDimensions": [ "string" ],
   "description": "string",
   "name": "string"
}
```

Téléchargez des données à l'aide de AWS IoT Jobs

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Avec AWS IoT Jobs, vous pouvez configurer des campagnes pour télécharger les données des véhicules stockées dans le cloud chaque fois que vous en avez besoin.

Pour créer un document d'emploi pour votre campagne

- Utilisez l'exemple suivant pour créer un document de travail pour la campagne. Un document de travail est un fichier .json qui contient des informations sur les véhicules ou les flottes

nécessaires à l'exécution d'une tâche. Pour plus d'informations sur la création d'un document de travail, voir Créer et gérer des tâches AWS CLIà l'aide du guide du AWS IoT développeur.

Pour demander qu'un seul véhicule télécharge des données, définissez l'objectif de la tâche sur l' AWS IoT élément associé au véhicule. Pour demander à plusieurs véhicules (dans la même campagne) de télécharger des données, créez un groupe d'objets contenant tous les éléments correspondant aux véhicules, puis définissez l'objectif du travail sur le groupe d'objets.

```
{
  "version": "1.0",
  "parameters": {
     "campaignArn": ${aws:iot:parameter:campaignArn},
     "endTime": ${aws:iot:parameter:endTime}
  }
}
```

a. `CampaignArn`Remplacez-le par le Amazon Resource Name (ARN) d'une campagne dans la même région et le même compte. L'ARN de la campagne est obligatoire.

b. (Facultatif) Remplacez `endTime` par l'horodatage des données collectées sur le véhicule au format ISO 8601 UTC (sans millisecondes). Par exemple, `2024-03-05T23:00:00Z`. L'horodatage est exclusif et détermine le dernier point de données à télécharger. Si vous omettez`endTime`, le logiciel Edge Agent continue à charger jusqu'à ce que toutes les données stockées d'une campagne soient téléchargées. Une fois toutes les données téléchargées, le statut d'exécution de la tâche est mis à jour sur`SUCCEEDED`. L'état de la tâche est mis à jour vers`COMPLETED`.

Pour créer une tâche à l'aide d'un modèle de tâche géré

1. Choisissez Iot-IO TFleet Wise- dans CollectCampaignData la liste des modèles gérés. Pour plus d'informations, voir Création d'une tâche à partir de modèles AWS gérés dans le Guide du AWS IoT développeur.
2. Le modèle géré possède les `endTime` paramètres `CampaignArn` et.

   a. `CampaignArn`Remplacez-le par le Amazon Resource Name (ARN) d'une campagne dans la même région et le même compte. L'ARN de la campagne est obligatoire.

   b. (Facultatif) Remplacez `endTime` par l'horodatage des données collectées sur le véhicule au format ISO 8601 UTC (sans millisecondes). Par exemple, `2024-03-05T23:00:00Z`.

L'horodatage est exclusif et détermine le dernier point de données à télécharger. Si vous omettez`endTime`, le logiciel Edge Agent continue à charger jusqu'à ce que toutes les données stockées d'une campagne soient téléchargées. Une fois toutes les données téléchargées, le statut d'exécution de la tâche est mis à jour sur`SUCCEEDED`. L'état de la tâche est mis à jour vers`COMPLETED`.

Pour les rubriques de résolution des problèmes connexes, voirStocker et transférer les numéros.

Pour plus d'informations sur les AWS IoT tâches, consultez la section Offres d'emploi du guide du AWS IoT développeur.

Collectez les données relatives aux codes de diagnostic à l'aide de l' AWS IoT FleetWise

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Lorsqu'un véhicule détecte une erreur, il génère un code de diagnostic (DTC) et enregistre un instantané des capteurs ou actionneurs concernés. DTCs vous aider à détecter les erreurs en temps quasi réel, à en comprendre la cause et à prendre des mesures correctives. AWS L'IoT FleetWise prend en charge la collecte DTCs, y compris des instantanés DTC correspondants et des données étendues par le biais d'une campagne de collecte de données. Cette rubrique présente les concepts, les flux de travail et les mots clés qui facilitent la collecte de données DTC, illustrés par des exemples.

Les concepts clés de l'utilisation du DTC sont présentés ci-dessous.

Fonctions définies sur mesure

Une fonction personnalisée est la capacité d'invoquer et d'exécuter vos propres fonctions prédéfinies sur l'agent Edge, étendant ainsi le concept de décodage personnalisé. Ces fonctions sont utilisées en coordination avec l' FleetWise agent AWS IoT. Le FleetWise logiciel Edge Agent for AWS IoT fournit des fonctions intégrées pour calculer les statistiques des signaux, telles

que les valeurs minimales, maximales et moyennes. Une fonction personnalisée étend cette fonctionnalité en vous permettant de créer une logique adaptée à des cas d'utilisation spécifiques. Pour la collecte de données relatives aux codes de diagnostic (DTC), les développeurs peuvent tirer parti de fonctions personnalisées pour mettre en œuvre des mécanismes de récupération de données avancés, tels que la récupération de codes DTC, d'instantanés et de données étendues directement depuis le périphérique du véhicule via des services de diagnostic unifiés (UDS) ou d'autres interfaces de diagnostic.

Pour plus d'informations, consultez le guide des fonctions personnalisées et l'implémentation de référence de la collecte de données DTC dans le Guide du développeur de l'agent Edge.

Récupération du signal

Dans les campagnes de collecte de données, les signaux sont généralement collectés en continu à partir d'un appareil et mis en mémoire tampon sur le logiciel Edge Agent. Les signaux sont ensuite téléchargés ou stockés périodiquement dans le cadre de campagnes basées sur le temps ou déclenchés par des conditions spécifiques dans le cadre de campagnes basées sur des conditions. Cependant, en raison des préoccupations liées à la congestion du trafic des appareils, les signaux DTC ne peuvent pas être collectés à partir des appareils et mis en mémoire tampon en continu. Pour résoudre ce problème, AWS l'IoT FleetWise fournit une fonction de récupération du signal, qui garantit que le signal cible est extrait de manière discontinue depuis un appareil.

La récupération de signaux prend en charge les actions périodiques et conditionnelles. Vous pouvez définir la méthode pilotée par la récupération, les conditions et les actions exactes à l'aide de fonctions personnalisées pour chaque signal qui ne doit pas être collecté en continu par un appareil. Pour les signaux gérés par le mécanisme de récupération des signaux, le type de déclencheur et les conditions du stockage local ou du téléchargement dans le cloud sont toujours régis par les `CollectionScheme` deux `timeBasedCollectionScheme` et `conditionBasedCollectionScheme` sont pris en charge, ce qui est identique à celui des signaux ordinaires.

Les rubriques suivantes expliquent comment créer et utiliser DTCs.

Rubriques

- Mots clés du code de diagnostic des problèmes
- Créez une campagne de collecte de données pour les codes d'erreur diagnostiques
- Cas d'utilisation du code de diagnostic

Mots clés du code de diagnostic des problèmes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

`signalsToFetch`paramètre pour créer une campagne

Utilisez la signalsToFetchsyntaxe pour configurer la manière dont les informations de signal peuvent être extraites sur le Edge. L'extraction de signaux standard est contrôlée par la modélisation sous forme de règles définies explicitement dans un manifeste du décodeur ou définies de manière personnalisée via Edge First Modeling. Grâce aux signaux à récupérer, vous pouvez définir quand et comment les données sont récupérées pendant les campagnes.

Les signaux à récupérer permettent de collecter des informations DTC. Par exemple, vous pouvez créer un signal de type chaîne nommé `DTC_Info` qui peut contenir des informations DTC pour chaque unité de commande moteur (ECU). Vous pouvez également filtrer pour un ECU spécifique.

- `SignalFetchInformation`définitions de structure et de paramètres.

```
structure SignalFetchInformation {
    @required
    fullyQualifiedName: NodePath,
    @required
    signalFetchConfig: SignalFetchConfig,
    // Conditional language version for this config
    conditionLanguageVersion: languageVersion,
    @required
    actions: EventExpressionList,
}
```

  - `fullyQualifiedName`: le nom complet (FQDN) du signal pour lequel vous souhaitez utiliser la fonction de récupération personnalisée.
  - `signalFetchConfig`: définit les règles relatives à la manière dont les signaux définis ci-dessus doivent être récupérés. Il prend en charge l'extraction basée sur le temps et les conditions.

- `conditionLanguageVersion`: la version du langage conditionnel utilisée pour analyser l'expression dans la configuration.
- `actions`: une liste de toutes les expressions d'action évaluées sur le Edge. The Edge obtiendra la valeur du signal défini.

> **Important**
>
> Les actions ne peuvent être utilisées que`custom_function`.

Mots clés d'expression de campagne

L'expression suivante prend le nom complet d'un signal pris en charge par le véhicule et renvoie la valeur true si le signal ne contient aucune donnée dans les tampons de signaux du Edge. Sinon, elle renvoie false.

```
isNull(signalFqdn:String): Boolean
```

Example utilisation

```
isNull($variable.`Vehicle.ECU1.DTC_INFO`) == false

We want to make sure DTC_Info signal is being generated
on edge.
```

Cette expression prend les données suivantes :

Nom de la fonction : chaîne

Le nom de la fonction personnalisée prise en charge par l'Edge

paramètres : varargs ***Expression***

Paramètres pour`functionName`. Il peut s'agir de n'importe quelle liste d'expressions.

Les paramètres prennent en charge le type littéral : String, Int, Boolean ou Double.

```
custom_function(functionName:String, params: varargsExpression): Void
```

Example utilisation

```
{
       "fullyQualifiedName":"Vehicle.ECU1.DTC_INFO",
       "signalFetchConfig":{
          "timeBased":{
             "executionFrequencyMs":2000
          }
       },
       "actions":"custom_function(“DTC_QUERY”, -1, 2, -1)"
    }
```

Créez une campagne de collecte de données pour les codes d'erreur diagnostiques

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Cette rubrique explique comment créer une campagne de collecte de données pour les codes de diagnostic (DTC).

1. Définissez un signal personnalisé sur le Edge. Vous devez définir les règles de décodage pour le signal DTC sur le Edge en tant que signal décodé personnalisé. Pour de plus amples informations, veuillez consulter Tutoriel : Configuration de la collecte de données indépendante du réseau à l'aide d'une interface de décodage personnalisée.
2. Définissez une fonction personnalisée sur le Edge. Vous devez définir une fonction personnalisée pour collecter les signaux DTC sur le Edge lors de la compilation.

   Pour plus d'informations, consultez le guide des fonctions personnalisées et l'implémentation de référence de la collecte de données DTC dans le Guide du développeur de l'agent Edge.

> **Note**
>
> Un exemple de fonction personnalisée est DTC_QUERY illustré dans le script de démonstration.

3. Créez un catalogue de signaux qui modélise un signal DTC sous forme de chaîne.

```
[
 {
    "branch": {
        "fullyQualifiedName": "Vehicle",
        "description": "Vehicle"
        }
      },
      {
    "branch": {
        "fullyQualifiedName": "Vehicle.ECU1",
        "description": "Vehicle.ECU1"
        }
      },
      {
    "sensor": {
        "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
        "description": "Vehicle.ECU1.DTC_INFO",
        "dataType": "STRING"
      }
   }
 ]
```

4. Créez et activez un modèle de véhicule avec le signal DTC ajouté.
5. Créez et activez un manifeste du décodeur avec le signal DTC ajouté. Le signal DTC doit être de type décodeur de CUSTOM_DECODING_SIGNAL signal avec un type d'interface CUSTOM_DECODING_INTERFACE réseau.

   Example décodeur de signal

```
[
  {
    "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
    "interfaceId": "UDS_DTC",
    "type": "CUSTOM_DECODING_SIGNAL",
```

```
    "customDecodingSignal": {
      "id": "Vehicle.ECU1.DTC_INFO"
    }
  }
 ]
```

Example interface réseau

```
[
  {
    "interfaceId": "UDS_DTC",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
```

> **Note**
>
> Les signaux CAN (Controller Area Network) ne prennent pas en charge le type de données chaîne.

6. Fournir et créer des véhicules. Les véhicules doivent utiliser un modèle de véhicule (manifeste du modèle) et un manifeste de décodeur qui ont été activés lors des étapes précédentes.
7. Créez et approuvez la campagne. Vous devez créer une campagne en définissant des signaux DTC (éventuellement avec des signaux de télémétrie) et en les déployant sur les véhicules.
8. Accédez aux données dans la destination définie. Les données DTC incluent le `DTCCodeDTCSnapshot`, et `DTCExtendedDatastrings` sous forme de chaîne brute dans la destination de données définie dans la campagne.

Cas d'utilisation du code de diagnostic

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les cas d'utilisation suivants supposent que la `DTC_QUERY` fonction a été définie dans le script de démonstration.

Récupération périodique

Récupérez une collection DTC à des intervalles configurés.

L'exemple suivant est une campagne qui consiste à récupérer périodiquement des signaux `Vehicle.DTC_INFO` pour tous DTCs avec un masque de statut pour tous ECUs. Il existe une condition pour laquelle les données sont collectées pour`Vehicle.DTC_INFO`.

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "timeBased": {
        // The FleetWise Edge Agent will query the UDS module for all DTCs every five
 seconds.
          "executionFrequencyMs": 5000
        }
      },
      "actions": [
      // Every five seconds, this action is called and its output is stored in the
      // signal history buffer of Vehicle.DTC_INFO
        "custom_function(\"DTC_QUERY\", -1, 2, -1)"
      ]
    }
  ],
  "signalsToCollect": [
    {
```

```
      "name": "Vehicle.ECU1.DTC_INFO"
    }
  ],
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      // Whenever a new DTC is filled into the signal, the data is ingested.
      "expression": "!isNull($variable.`Vehicle.ECU1.DTC_INFO`)",
      "minimumTriggerIntervalMs": 1000,
      // Make sure that data is ingested only when there are new DTCs.
      "triggerMode": "RISING_EDGE"
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Récupération basée sur les conditions

Récupère une collection DTC lorsqu'une condition est remplie. Par exemple, lorsque le signal CAN est émis`Vehicle.Ignition == 1`, récupérez et téléchargez les données DTC.

L'exemple de campagne suivant comporte une récupération de signal basée sur les conditions `Vehicle.ECU1.DTC_INFO` pour vérifier si le DTC (» AAA123 «) est en attente avec RecordNumber 1 pour ECU-1. Cette campagne comprend une collecte et un téléchargement de données basés sur le temps.

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
```

```
        "conditionBased": {
        // The action will only run when the ignition is on.
          "conditionExpression": "$variable.`Vehicle.Ignition` == 1",
          "triggerMode": "ALWAYS"
        }
      },
      // The UDS module is only requested for the specific ECU address and the specific
 DTC Number/Status.
      "actions": ["custom_function(\"DTC_QUERY\", 1, 2, 8, \"0xAAA123\")"]
    }
  ],
  "signalsToCollect": [
    {
      "name": "Vehicle.ECU1.DTC_INFO"
    },
    {
      "name": "Vehicle.Ignition"
    }
  ],
  "collectionScheme": {
    "timeBasedCollectionScheme": {
      "periodMs": 10000
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Récupération à la demande

Récupérez un DTC spécifique pour une flotte.

Pour un cas d'utilisation à la demande, vous pouvez utiliser la même campagne que celle définie dans la récupération périodique. L'effet à la demande est obtenu en suspendant la campagne peu

après son déploiement à l'aide de la FleetWise console AWS IoT ou en exécutant la commande CLI suivante.

- Remplacez *command-name* par le nom de la commande.

```
aws iotfleetwise update-campaign \
    --name campaign-name \
    --action APPROVE
```

Ensuite, suspendez la campagne après l'arrivée des données DTC.

```
aws iotfleetwise update-campaign \
    --name campaign-name \
    --action SUSPEND
```

Vous pouvez reprendre la campagne pour récupérer les données DTC.

```
aws iotfleetwise update-campaign \
    --name campaign-name \
    --action RESUME
```

Visualisez les données des FleetWise véhicules AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Le FleetWise logiciel Edge Agent for AWS IoT envoie les données du véhicule sélectionnées vers une rubrique MQTT, ou les transfère vers Amazon Timestream ou Amazon Simple Storage Service (Amazon S3). Une fois que vos données sont arrivées dans leur destination, vous pouvez utiliser d'autres AWS services pour les traiter, les rediriger, les visualiser et les partager.

> **Note**
>
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

Traitement des données du véhicule envoyées à un sujet MQTT

Les données du véhicule envoyées par messagerie MQTT sont transmises en temps quasi réel et vous permettent d'utiliser les règles pour agir ou d'acheminer les données vers d'autres destinations. Pour plus d'informations sur l'utilisation de MQTT, consultez la section Protocoles et règles de communication des appareils AWS IoT dans le guide du AWS IoT Core développeur.

Le schéma par défaut des données envoyées dans un message MQTT contient les champs suivants.

Nom de champ	Type de données	Description
`eventId`	varchar	ID de l'événement de collecte de données.
`vehicleName`	varchar	L'identifiant du véhicule à partir duquel les données ont été collectées.
`name`	varchar	Nom de la campagne utilisée par le logiciel Edge Agent pour collecter des données.
`time`	timestamp	Horodatage du point de données.
`measure_name`	varchar	Le nom du signal.
`measure_value::bigint`	bigint	Valeurs de signal de type Integer.

Nom de champ	Type de données	Description
`measure_value::double`	double	Valeurs de signal de type Double.
`measure_value::boolean`	boolean	Valeurs de signal de type booléen.
`measure_value::varchar`	varchar	Valeurs de signal de type varchar.

Traitez les données du véhicule dans Timestream

Timestream est une base de données de séries chronologiques entièrement gérée qui peut stocker et analyser des milliards de points de données de séries chronologiques par jour. Vos données sont stockées dans une table Timestream gérée par le client. Vous pouvez utiliser Timestream pour interroger les données des véhicules afin d'obtenir des informations sur vos véhicules. Pour plus d'informations, consultez Qu'est-ce qu'Amazon Timestream ?

Le schéma par défaut des données transférées vers Timestream contient les champs suivants.

Nom de champ	Type de données	Description
`eventId`	varchar	ID de l'événement de collecte de données.
`vehicleName`	varchar	L'identifiant du véhicule à partir duquel les données ont été collectées.
`name`	varchar	Nom de la campagne utilisée par le logiciel Edge Agent pour collecter des données.

Nom de champ	Type de données	Description
`time`	timestamp	Horodatage du point de données.
`measure_name`	varchar	Le nom du signal.
`measure_value::bigint`	bigint	Valeurs de signal de type Integer.
`measure_value::double`	double	Valeurs de signal de type Double.
`measure_value::boolean`	boolean	Valeurs de signal de type booléen.
`measure_value::varchar`	varchar	Valeurs de signal de type varchar.

Visualisez les données du véhicule stockées dans Timestream

Une fois les données de votre véhicule transférées vers Timestream, vous pouvez utiliser les AWS services suivants pour visualiser, surveiller, analyser et partager vos données.

- Visualisez et surveillez les données dans des tableaux de bord à l'aide de Grafana ou d'Amazon Managed Grafana. Vous pouvez visualiser les données provenant de plusieurs AWS sources (telles qu'Amazon CloudWatch et Timestream) et d'autres sources de données avec un seul tableau de bord Grafana.
- Analysez et visualisez les données dans des tableaux de bord en utilisant QuickSight.

Traitez les données du véhicule dans Amazon S3

Amazon S3 est un service de stockage d'objets qui stocke et protège n'importe quel volume de données. Vous pouvez utiliser S3 pour divers cas d'utilisation, tels que les lacs de données, la sauvegarde et la restauration, l'archivage, les applications d'entreprise, AWS IoT les appareils et l'analyse des mégadonnées. Vos données sont stockées dans S3 sous forme d'objets dans des buckets. Pour plus d’informations, consultez la section Qu’est-ce qu’Amazon S3 ?.

Le schéma par défaut des données transférées vers Amazon S3 contient les champs suivants.

Nom de champ	Type de données	Description
`eventId`	varchar	ID de l'événement de collecte de données.
`vehicleName`	varchar	L'identifiant du véhicule à partir duquel les données ont été collectées.
`name`	varchar	Nom de la campagne utilisée par le logiciel Edge Agent pour collecter des données.
`time`	timestamp	Horodatage du point de données.
`measure_name`	varchar	Le nom du signal.
`measure_value_BIGINT`	bigint	Valeurs de signal de type Integer.
`measure_value_DOUBLE`	double	Valeurs de signal de type Double.
`measure_value_BOOLEAN`	boolean	Valeurs de signal de type booléen.
`measure_value_STRUCT`	struct	Valeurs de signal de type Struct.
`measure_value_VARCHAR`	varchar	Valeurs de signal de type varchar.

Format d'objet Amazon S3

AWS L'IoT FleetWise transfère les données du véhicule vers S3 où elles sont enregistrées sous forme d'objet. Vous pouvez utiliser l'URI de l'objet qui identifie les données de manière unique pour rechercher les données de la campagne. Le format d'URI de l'objet S3 dépend du fait que les données collectées sont des données non structurées ou traitées.

Données non structurées

Les données non structurées sont stockées dans S3 d'une manière non prédéfinie. Il peut être sous différents formats, tels que des images ou des vidéos.

Les messages du véhicule transmis à AWS l'IoT FleetWise avec des données de signal provenant de fichiers Amazon Ion sont décodés et transférés vers S3 sous forme d'objets. Les objets S3 représentent chaque signal et sont codés en binaire.

L'URI de l'objet S3 de données non structurées utilise le format suivant :

```
s3://bucket-name/prefix/unstructured-data/random-ID-yyyy-MM-dd-HH-mm-ss-SSS-
vehicleName-signalName-fieldName
```

Données traitées

Les données traitées sont stockées dans S3 et subissent des étapes de traitement qui valident, enrichissent et transforment les messages. Les listes d'objets et la vélocité sont des exemples de données traitées.

Les données transférées vers S3 sont stockées sous forme d'objets représentant des enregistrements mis en mémoire tampon pendant une période d'environ 10 minutes. Par défaut, AWS l'IoT FleetWise ajoute un préfixe d'heure UTC au format `year=YYYY/month=MM/date=DD/hour=HH` avant d'écrire des objets dans S3. Ce préfixe crée une hiérarchie logique dans le compartiment où chaque barre oblique (/) crée un niveau dans la hiérarchie. Les données traitées contiennent également l'URI de l'objet S3 pour les données non structurées.

L'URI de l'objet S3 des données traitées utilise le format suivant :

```
s3://bucket-name/prefix/processed-data/year=YYYY/month=MM/day=DD/hour=HH/
part-0000-random-ID.gz.parquet
```

Données brutes

Les données brutes, également appelées données primaires, sont des données collectées à partir de fichiers Amazon Ion. Vous pouvez utiliser les données brutes pour résoudre les problèmes ou pour identifier les causes d'erreurs.

L'URI de l'objet S3 de données brutes utilise le format suivant :

```
s3://bucket-name/prefix/raw-data/vehicle-name/eventID-timestamp.10n
```

Analyser les données du véhicule stockées dans Amazon S3

Une fois les données de votre véhicule transférées vers S3, vous pouvez utiliser les AWS services suivants pour surveiller, analyser et partager vos données.

Extrayez et analysez les données à l'aide d'Amazon SageMaker AI pour les flux de travail d'étiquetage et d'apprentissage automatique (ML) en aval.

Pour plus d'informations, consultez les rubriques suivantes dans le manuel Amazon SageMaker AI Developer Guide :

- Données de processus
- Formez des modèles d'apprentissage automatique
- Images d'étiquettes

Cataloguez vos données en les utilisant AWS Glue crawler et analysez-les dans Amazon Athena. Par défaut, les objets écrits dans S3 possèdent des partitions temporelles de style Apache Hive, avec des chemins de données contenant des paires clé-valeur reliées par des signes égaux.

Pour plus d'informations, consultez les rubriques suivantes dans le guide de l'utilisateur d'Amazon Athena :

- Partitionnement de données dans Athena
- Utilisation AWS Glue pour se connecter à des sources de données dans Amazon S3
- Bonnes pratiques lors de l'utilisation d'Athena avec AWS Glue

Visualisez les données QuickSight en lisant directement votre table Athena ou votre compartiment S3.

> **Tip**
>
> Si vous lisez directement depuis S3, vérifiez que les données de votre véhicule sont au format JSON, car elles QuickSight ne sont pas compatibles avec le format Apache Parquet.

Pour plus d'informations, consultez les rubriques suivantes du guide de QuickSight l'utilisateur Amazon :

- Sources de données prises en charge
- Création d'une source de données

Commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.
> Cette documentation décrit comment utiliser la fonctionnalité de commandes pour AWS l'IoT FleetWise. Pour plus d'informations sur l'utilisation de la fonctionnalité de commandes dans AWS IoT Device Management, consultez la section commandes.
> Vous êtes seul responsable du déploiement des commandes d'une manière sûre et conforme aux lois applicables. Pour plus d'informations sur vos responsabilités, veuillez consulter les conditions AWS de service relatives AWS IoT aux services.

Utilisez la fonction de commandes pour exécuter des commandes sur un véhicule depuis le cloud. Les commandes ciblent un appareil à la fois et peuvent être utilisées pour des applications à faible latence et à haut débit, par exemple pour récupérer les journaux côté appareil ou pour initier un changement d'état de l'appareil.

La commande est une ressource gérée par AWS IoT Device Management. Il contient des configurations réutilisables qui sont appliquées lors de l'envoi d'une exécution de commande au véhicule. Vous pouvez prédéfinir un ensemble de commandes pour des cas d'utilisation spécifiques ou les utiliser pour créer des configurations réutilisables pour des cas d'utilisation récurrents. Par exemple, vous pouvez configurer des commandes qui peuvent être utilisées par une application pour verrouiller la portière d'un véhicule ou pour modifier la température à distance.

À l'aide de la fonction de AWS IoT commandes, vous pouvez :

- Créez une ressource de commande et réutilisez la configuration pour envoyer plusieurs commandes à votre appareil cible, puis exécutez-les sur le périphérique.
- Contrôlez la granularité avec laquelle vous souhaitez que chaque commande soit exécutée sur l'appareil. Par exemple, vous pouvez approvisionner un véhicule en tant qu' AWS IoT objet, puis envoyer une commande pour verrouiller ou déverrouiller les portières du véhicule.
- Exécutez plusieurs commandes simultanément sur l'appareil cible sans attendre que la précédente soit terminée.

- Choisissez d'activer les notifications pour les événements liés aux commandes et de récupérer les informations d'état et de résultat sur l'appareil lors de l'exécution de la commande et une fois celle-ci terminée.

Les rubriques suivantes expliquent comment créer, envoyer, recevoir et gérer des commandes.

Rubriques

- Concepts de commandes
- Véhicules et commandes
- Création et gestion de commandes
- Démarrez et surveillez les exécutions de commandes
- Exemple : utilisation de commandes pour contrôler le mode de direction d'un véhicule (AWS CLI)
- Scénarios d'utilisation des commandes

Concepts de commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les commandes sont des instructions envoyées depuis le cloud à votre appareil cible. L'équipement cible peut être un véhicule et il doit être enregistré en tant qu'AWS IoT objet dans le registre des objets. La commande peut contenir des paramètres qui définissent une action que les actionneurs du véhicule doivent effectuer. Le véhicule analyse ensuite la commande et ses paramètres, puis les traite pour effectuer l'action correspondante. Il répond ensuite à l'application cloud avec l'état de l'exécution de la commande.

Pour le flux de travail détaillé, voirVéhicules et commandes.

Rubriques

- Concepts clés des commandes
- État de l'exécution de la commande

Concepts clés des commandes

Vous trouverez ci-dessous quelques concepts clés relatifs à l'utilisation de la fonctionnalité de commandes et son fonctionnement avec les modèles d'état du dernier état connu (LKS).

Commande

Une commande est une entité que vous pouvez utiliser pour envoyer des instructions à un véhicule physique afin qu'il exécute des actions telles que la mise en marche du moteur ou la modification de la position des vitres. Vous pouvez prédéfinir un ensemble de commandes pour des cas d'utilisation spécifiques ou les utiliser pour créer des configurations réutilisables pour des cas d'utilisation récurrents. Par exemple, vous pouvez configurer des commandes qui peuvent être utilisées par une application pour verrouiller la portière d'un véhicule ou pour modifier la température à distance.

Namespace

Lorsque vous utilisez la fonctionnalité des commandes, vous devez spécifier l'espace de noms de la commande. Lorsque vous créez une commande dans AWS l'IoT FleetWise, vous devez la choisir `AWS-IoT-FleetWise` comme espace de noms. Lorsque vous utilisez cet espace de noms, vous devez fournir les paramètres qui seront utilisés pour exécuter la commande sur le véhicule. Si vous souhaitez créer une commande à la AWS IoT Device Management place, vous devez utiliser l'espace de `AWS-IoT` noms à la place. Pour plus d'informations, consultez les commandes du guide du AWS IoT Device Management développeur.

États de commande

Les commandes que vous créez seront disponibles, ce qui signifie qu'elles peuvent être utilisées pour démarrer une exécution de commande sur le véhicule. Si une commande devient obsolète, vous pouvez la déprécier. Pour une commande à l'état obsolète, les exécutions de commandes existantes seront exécutées jusqu'à la fin. Vous ne pouvez pas mettre à jour la commande ni exécuter de nouvelles exécutions. Pour envoyer de nouvelles exécutions, vous devez restaurer la commande afin qu'elle soit disponible.

Vous pouvez également supprimer une commande si elle n'est plus nécessaire. Lorsque vous marquez une commande pour suppression, si la commande est devenue obsolète pendant une durée supérieure au délai maximum de 24 heures, elle est immédiatement supprimée. Si la commande n'est pas obsolète, ou si elle l'a été pendant une durée inférieure au délai maximum, elle sera en attente de suppression. La commande sera automatiquement supprimée de votre compte au bout de 24 heures.

Paramètres

Lorsque vous créez une commande, vous pouvez éventuellement spécifier les paramètres que vous souhaitez que le véhicule cible exécute lors de l'exécution de la commande. La commande que vous créez est une configuration réutilisable qui peut être utilisée pour envoyer plusieurs exécutions de commandes à votre véhicule et les exécuter simultanément. Vous pouvez également spécifier les paramètres uniquement lors de l'exécution et choisir d'effectuer une opération unique consistant à créer une commande et à l'envoyer à votre véhicule.

Véhicule cible

Lorsque vous souhaitez exécuter la commande, vous devez spécifier un véhicule cible qui recevra la commande et exécutera des actions spécifiques. Le véhicule cible doit déjà avoir été enregistré en tant qu'objet auprès de AWS IoT. Une fois que vous avez envoyé la commande au véhicule, celui-ci commence à exécuter une instance de la commande en fonction des paramètres et des valeurs que vous avez spécifiés.

Actionneurs

Lorsque vous souhaitez exécuter la commande, vous devez spécifier les actionneurs du véhicule qui recevront la commande et leurs valeurs qui déterminent les actions à effectuer. Vous pouvez éventuellement configurer des valeurs par défaut pour les actionneurs afin d'éviter d'envoyer des commandes inexactes. Par exemple, vous pouvez utiliser la valeur par défaut d'`LockDoor`un actionneur de verrouillage de porte afin que la commande ne déverrouille pas accidentellement les portes. Pour des informations générales sur les actionneurs, voirConcepts clés.

Support des types de données

Les types de données suivants sont pris en charge pour les actionneurs utilisés pour la fonction de commandes.

> **Note**
>
> Les tableaux ne sont pas pris en charge pour les données télématiques, les commandes ou le dernier état connu (LKS). Vous ne pouvez utiliser le type de données matriciel que pour les données des systèmes de vision.

- Types à virgule flottante. Les types suivants sont pris en charge.
  - Float (32 bits)
  - Double (64 bits)

- Entier (signé et non signé). Les types d'entiers suivants sont pris en charge.
  - int8 et uint8
  - int16 et uint16
  - int32 et uint32
- Longue. Les types longs suivants sont pris en charge.
  - Longue (int64)
  - Long non signé (uint64)
- Chaîne
- Booléen

Exécution de commandes

Une exécution de commande est une instance d'une commande exécutée sur un équipement cible. Le véhicule exécute la commande en utilisant les paramètres que vous avez spécifiés lorsque vous avez créé la commande ou lorsque vous l'avez lancée. Le véhicule exécute ensuite les opérations spécifiées et renvoie le statut de l'exécution.

> **Note**
>
> Pour un véhicule donné, vous pouvez exécuter plusieurs commandes simultanément. Pour plus d'informations sur le nombre maximal d'exécutions simultanées que vous pouvez exécuter pour chaque véhicule, consultez la section quotas de AWS IoT Device Management commandes.

Modèles d'état du dernier état connu (LKS)

Les modèles d'état fournissent un mécanisme permettant aux propriétaires de véhicules de suivre l'état de leur véhicule. Pour surveiller le dernier état connu (LKS) de vos véhicules en temps quasi réel, vous pouvez créer des modèles d'état et les associer à vos véhicules.

À l'aide de la fonction de commandes, vous pouvez effectuer des opérations « à la demande » qui peuvent être utilisées pour la collecte et le traitement des données d'État. Par exemple, vous pouvez demander l'état actuel du véhicule une seule fois (récupérer), ou activer ou désactiver les modèles d'état LKS précédemment déployés pour commencer ou arrêter de communiquer les données du véhicule. Pour des exemples illustrant l'utilisation de commandes avec des modèles d'état, consultezScénarios d'utilisation des commandes.

État de l'exécution de la commande

Une fois que vous avez commencé l'exécution de la commande, votre véhicule peut publier le statut de l'exécution et fournir les raisons de ce statut sous forme d'informations supplémentaires sur l'exécution. Les sections suivantes décrivent les différents statuts d'exécution des commandes, ainsi que les codes d'état.

Rubriques

- État d'exécution de la commande, code de raison et description
- État d'exécution des commandes et codes d'état
- État du délai d'exécution des commandes

État d'exécution de la commande, code de raison et description

Pour signaler les mises à jour de l'état d'exécution des commandes, vos véhicules peuvent utiliser l'`UpdateCommandExecution`API pour publier les informations d'état mises à jour dans le cloud, en utilisant les rubriques réservées aux commandes décrites dans le guide du AWS IoT Core développeur. Lorsque vous signalez les informations d'état, vos appareils peuvent fournir un contexte supplémentaire sur l'état de chaque exécution de commande à l'aide de l'`StatusReason`objet, `reasonCode` ainsi `reasonDescription` que sur les champs contenus dans l'objet.

État d'exécution des commandes et codes d'état

Le tableau suivant indique les différents codes d'état d'exécution de commandes et les statuts autorisés vers lesquels une exécution de commande peut passer. Il indique également si l'exécution d'une commande est « terminale » (c'est-à-dire qu'aucune autre mise à jour de statut n'est prévue), si le changement est initié par le véhicule ou par le cloud, ainsi que les différents codes d'état prédéfinis et leur correspondance avec les statuts signalés par le cloud.

- Pour plus d'informations sur l' AWS IoT FleetWise utilisation des codes d'état prédéfinis et de l'`statusReason`objet, consultez la section État des commandes dans la documentation du FleetWise logiciel Edge Agent for AWS IoT.
- Pour plus d'informations sur les exécutions terminales et non terminales, ainsi que sur les transitions entre les statuts, consultez la section État d'exécution des commandes dans le guide du AWS IoT Core développeur.

État et source de l'exécution des commandes

État de l'exécution de la commande	Description	Initié par un appareil/le cloud ?	Exécution du terminal ?	Transitions de statut autorisées	Codes d'état prédéfinis
`CREATED`	Lorsque la demande d'API pour démarrer l'exécution de la commande (`StartCommandExecution` API) aboutit, le statut d'exécution de la commande passe à`CREATED`.	Cloud	Non	• `IN_PROGRESS` • `SUCCEEDED` • `FAILED` • `REJECTED` • `TIMED_OUT`	Aucun
`IN_PROGRESS`	Lorsque le véhicule commence à exécuter la commande, il peut publier un message dans le sujet de réponse pour mettre à jour le statut`IN_PROGRESS` .	Appareil	Non	• EN_COURS • RÉUSSI • ÉCHEC • REFUSÉE • TIMED_OUT	`COMMAND_STATUS_COMMAND_IN_PROGRESS`

État de l'exécution de la commande	Description	Initié par un appareil/le cloud ?	Exécution du terminal ?	Transitions de statut autorisées	Codes d'état prédéfinis
`SUCCEEDED`	Lorsque le véhicule a traité avec succès la commande et terminé son exécution, il peut publier un message dans le sujet de réponse pour mettre à jour le statut`SUCCEEDED` .	Appareil	Oui	Ne s'applique pas	`COMMAND_STATUS_SUCCEEDED`
`FAILED`	Lorsque le véhicule n'exécute pas la commande, il peut publier un message dans le sujet de réponse pour mettre à jour le statut`FAILED`.	Appareil	Oui	Ne s'applique pas	`COMMAND_STATUS_EXECUTION_FAILED`

État de l'exécution de la commande	Description	Initié par un appareil/le cloud ?	Exécution du terminal ?	Transitions de statut autorisées	Codes d'état prédéfinis
`REJECTED`	Si le véhicule n'accepte pas la commande, il peut publier un message dans le sujet de réponse pour mettre à jour le statut`REJECTEI`	Appareil	Oui	Ne s’applique pas	Aucun

État de l'exécution de la commande	Description	Initié par un appareil/le cloud ?	Exécution du terminal ?	Transitions de statut autorisées	Codes d'état prédéfinis
`TIMED_OUT`	L'état d'exécution de la commande peut changer `TIMED_OUT` pour l'une des raisons suivantes. • Le résultat de l'exécution de la commande n'a pas été reçu et le cloud signale automatiquement un `TIMED_OUT` état. • Le véhicule indique qu'un délai s'est écoulé lorsqu'il a tenté d'exécuter la commande. Dans	Appareil et cloud	Non	• RÉUSSI • ÉCHEC • REFUSÉE • TIMED_OUT	`COMMAND_STATUS_EXECUTION_TIMEOUT`

État de l'exécution de la commande	Description	Initié par un appareil/le cloud ?	Exécution du terminal ?	Transitions de statut autorisées	Codes d'état prédéfinis
	ce cas, l'exécution de la commande devient terminale. Pour plus d'informations sur ce statut, consultez État du délai d'exécution des commandes.				

État du délai d'exécution des commandes

Un délai d'exécution d'une commande peut être signalé à la fois par le cloud et par l'appareil. Une fois la commande envoyée à l'appareil, une minuterie démarre. Si aucune réponse n'a été reçue de l'appareil dans le délai spécifié, le cloud signale un `TIMED_OUT` état. Dans ce cas, l'exécution de la commande dans le `TIMED_OUT` statut n'est pas terminale.

L'appareil peut remplacer cet état par un statut de terminal, tel que `SUCCEEDED``FAILED`, ou`REJECTED`. Il peut également signaler qu'un délai d'attente s'est produit lors de l'exécution de la commande. Dans ce cas, l'état d'exécution de la commande reste `TIMED_OUT` inchangé, mais les champs de l'`StatusReason`objet sont mis à jour en fonction des informations communiquées par l'appareil. L'exécution de la commande dans le `TIMED_OUT` statut devient désormais terminale.

Pour plus d'informations, consultez la section Considérations relatives au délai d'exécution des commandes dans le guide du AWS IoT Core développeur.

Véhicules et commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.
> Vous êtes seul responsable du déploiement des commandes d'une manière sûre et conforme aux lois applicables.

Pour utiliser la fonction de commandes :

1. Créez d'abord une ressource de commande. Spécifiez éventuellement les paramètres contenant les informations requises pour exécuter la commande.
2. Spécifiez le véhicule cible qui recevra la commande et exécutera les actions spécifiées.
3. Vous pouvez désormais exécuter la commande sur l'appareil cible et vérifier les détails de l'exécution de la commande pour récupérer l'état et utiliser CloudWatch les journaux pour résoudre les problèmes éventuels.

Les sections suivantes présentent le flux de travail entre les véhicules et les commandes.

Rubriques

- Présentation du flux de travail
- Flux de travail du véhicule
- Flux de travail des commandes
- (Facultatif) Notifications de commandes

Présentation du flux de travail

Les étapes suivantes fournissent une vue d'ensemble du flux de travail des commandes entre vos véhicules et les commandes. Lorsque vous utilisez l'une des commandes des opérations de l'API HTTP, la demande est signée à l'aide des informations d'identification Sigv4.

> **Note**
>
> À l'exception de l'opération `StartCommandExecution` API, toutes les opérations effectuées via le protocole HTTP utilisent le point de terminaison du plan de contrôle.

1. Établissez une connexion MQTT et abonnez-vous aux rubriques de commandes

   Pour préparer le flux de travail des commandes, les appareils doivent établir une connexion MQTT avec le `iot:Data-ATS` point de terminaison et s'abonner à la rubrique de demande de commandes mentionnée ci-dessus. En option, vos appareils peuvent également s'abonner aux sujets de réponse acceptés et rejetés pour les commandes.

2. Création d'un modèle de véhicule et d'une ressource de commande

   Vous pouvez désormais créer un véhicule et une ressource de commande à l'aide des opérations de l'API `CreateVehicle` et du plan de `CreateCommand` contrôle. La ressource de commande contient les configurations à appliquer lorsque la commande est exécutée sur le véhicule.

3. Lancer l'exécution de la commande sur le périphérique cible

   Démarrez l'exécution de la commande sur le véhicule à l'aide de l'API du plan de `StartCommandExecution` données avec le point de terminaison spécifique à votre compte`iot:Jobs`. L'API publie un message de charge utile codé en protobuf dans la rubrique de demande de commandes.

4. Mettre à jour le résultat de l'exécution de la commande

   Le véhicule traite la commande et la charge utile reçues, puis publie le résultat de l'exécution de la commande dans le sujet de réponse à l'aide de l'`UpdateCommandExecution`API. Si votre véhicule a souscrit aux commandes « sujets de réponse acceptés » et « rejetés », il recevra un message indiquant si la réponse a été acceptée ou rejetée par le service cloud.

5. (Facultatif) Récupère le résultat de l'exécution de la commande

   Pour récupérer le résultat de l'exécution de la commande, vous pouvez utiliser l'opération API du plan de `GetCommandExecution` contrôle. Une fois que votre véhicule a publié le résultat de l'exécution de la commande dans le sujet de réponse, cette API renvoie les informations mises à jour.

6. (Facultatif) Abonnement et gestion des événements liés aux commandes

   Pour recevoir des notifications concernant les mises à jour du statut d'exécution des commandes, vous pouvez vous abonner à la rubrique des événements relatifs aux commandes. Vous pouvez ensuite utiliser l'API du plan de `CreateTopicRule` contrôle pour acheminer les données des commandes et des événements vers d'autres applications, telles que AWS Lambda Functions ou Amazon SQS, et créer des applications à partir de cette API.

Flux de travail du véhicule

Les étapes suivantes décrivent en détail le flux de travail du véhicule lors de l'utilisation de la fonction de commandes.

> **Note**
>
> Les opérations décrites dans cette section utilisent le protocole MQTT.

1. Établir une connexion MQTT

   Pour préparer vos véhicules à utiliser la fonction de commandes, ils doivent d'abord se connecter au courtier de AWS IoT Core messages. Votre véhicule doit être autorisé à effectuer l'`iot:Connect`action de connexion AWS IoT Core et d'établissement d'une connexion MQTT avec le courtier de messages. Pour trouver le point de terminaison du plan de données qui vous convient Compte AWS, utilisez l'`DescribeEndpoint`API ou la commande `describe-endpoint` CLI, comme indiqué ci-dessous.

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

   L'exécution de cette commande renvoie le point de terminaison du plan de données spécifique au compte, comme indiqué ci-dessous.

   ```
   account-specific-prefix.iot.region.amazonaws.com
   ```

2. S'abonner à la rubrique de demande de commandes

   Une fois la connexion établie, vos appareils peuvent s'abonner à la rubrique de demande MQTT des AWS IoT commandes. Lorsque vous créez une commande et que vous lancez son

exécution sur votre appareil cible, un message de charge utile codé en protobuf est publié dans le sujet de la demande par le courtier de messages. Votre appareil peut ensuite recevoir le message de charge utile et traiter la commande. Dans cet exemple, remplacez-le *<DeviceID>* par l'identifiant unique de votre véhicule cible. Cet identifiant peut être l'identifiant unique de votre véhicule ou un nom d'objet

> **Note**
>
> Le message de charge utile envoyé à l'appareil doit utiliser le format protobuf.

```
$aws/commands/things/<DeviceID>/executions/+/request/protobuf
```

3. (Facultatif) Abonnez-vous aux rubriques de réponse aux commandes

   Vous pouvez éventuellement vous abonner à ces rubriques de réponse aux commandes pour recevoir un message indiquant si le service cloud a accepté ou rejeté la réponse de l'appareil.

   > **Note**
   >
   > Il est facultatif pour vos véhicules de s'abonner aux rubriques `/accepted` et `/rejected` aux réponses. Vos véhicules recevront automatiquement ces messages de réponse même s'ils ne se sont pas explicitement abonnés à ces sujets.

   ```
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/accepted
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/rejected
   ```

4. Mettre à jour le résultat de l'exécution d'une commande

   Le véhicule cible traite ensuite la commande. Il utilise ensuite l'`UpdateCommandExecution`API pour publier le résultat de l'exécution dans la rubrique de réponse MQTT suivante.

   > **Note**
   >
   > Pour un véhicule et une exécution de commande donnés, le champ *<DeviceID>* doit correspondre au champ correspondant dans le sujet de demande auquel l'appareil s'est abonné.

```
$aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf
```

L'`UpdateCommandExecution`API est une opération d'API de plan de données sur MQTT authentifiée par TLS.

- Si le service cloud a traité avec succès le résultat de l'exécution de la commande, un message est publié dans la rubrique MQTT acceptée. Le sujet accepté utilise le format suivant.

  ```
  $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
  accepted
  ```

- Si le service cloud ne parvient pas à traiter le résultat de l'exécution de la commande, une réponse est publiée dans le sujet MQTT rejeté. Le sujet rejeté utilise le format suivant.

  ```
  $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
  rejected
  ```

Pour plus d'informations sur cette API et un exemple, consultezMettre à jour le résultat d'exécution des commandes.

Flux de travail des commandes

Les étapes suivantes décrivent le flux de travail des commandes en détail.

> **Note**
>
> Les opérations décrites dans cette section utilisent le protocole HTTP.

1. Enregistrez votre véhicule

   Maintenant que vous avez préparé votre véhicule pour utiliser la fonction de commandes, vous pouvez préparer votre demande en enregistrant votre véhicule, puis en créant une commande qui sera envoyée au véhicule. Pour enregistrer le véhicule, créez une instance d'un modèle de véhicule (manifeste du modèle) à l'aide de l'opération API du plan de `CreateVehicle`contrôle. Pour plus d'informations et des exemples, voir Création d'un véhicule.

2. Création d'une commande

   Utilisez l'opération de l'API du plan de contrôle `CreateCommand`HTTP pour modéliser les commandes applicables au véhicule que vous ciblez. Spécifiez les paramètres et les valeurs par défaut à utiliser lors de l'exécution de la commande, et assurez-vous qu'elle utilise l'espace de `AWS-IoT-FleetWise` noms. Pour plus d'informations et des exemples d'utilisation de cette API, consultezCréation d'une ressource de commande.

3. Lancer l'exécution de la commande

   Vous pouvez désormais exécuter la commande que vous avez créée sur le véhicule à l'aide de l'opération API du plan de `StartCommandExecution`données. AWS IoT Device Management récupère la commande et les paramètres de commande, et valide la demande entrante. Il invoque ensuite AWS IoT FleetWise l'API avec les paramètres requis pour générer la charge utile spécifique au véhicule. La charge utile est ensuite envoyée à l'appareil via MQTT AWS IoT Device Management sur le sujet de demande de commande auquel votre appareil s'est abonné. Pour plus d'informations et des exemples d'utilisation de cette API, consultezEnvoyer une commande (AWS CLI).

   ```
   $aws/commands/things/<DeviceID>/executions/+/request/protobuf
   ```

   > **Note**
   >
   > Si l'appareil était hors ligne lorsque la commande a été envoyée depuis le cloud et que des sessions persistantes MQTT sont en cours d'utilisation, la commande attend le courtier de messages. Si l'appareil revient en ligne avant la fin du délai imparti et s'il s'est abonné à la rubrique de demande de commandes, il peut alors traiter la commande et publier le résultat dans la rubrique de réponse. Si l'appareil ne revient pas en ligne avant le délai imparti, l'exécution de la commande expirera et le message de charge utile expirera.

4. Récupérez l'exécution de la commande

   Après avoir exécuté la commande sur l'appareil, utilisez l'opération API du plan de `GetCommandExecution`contrôle pour récupérer et surveiller le résultat de l'exécution de la commande. Vous pouvez également utiliser l'API pour obtenir des informations supplémentaires sur les données d'exécution, telles que la date de dernière mise à jour, la date de fin de l'exécution et les paramètres spécifiés.

> **Note**
>
> Pour récupérer les dernières informations d'état, votre appareil doit avoir publié le résultat de l'exécution de la commande dans la rubrique de réponse.

Pour plus d'informations et des exemples d'utilisation de cette API, consultezObtenir l'exécution d'une commande.

(Facultatif) Notifications de commandes

Vous pouvez vous abonner aux événements de commandes pour recevoir des notifications lorsque le statut de l'exécution d'une commande change. Les étapes suivantes vous montrent comment vous abonner à des événements de commandes, puis comment les traiter.

1. Créer une règle de rubrique

   Vous pouvez vous abonner à la rubrique des événements relatifs aux commandes et recevoir des notifications lorsque le statut de l'exécution d'une commande change. Vous pouvez également créer une règle thématique pour acheminer les données traitées par le véhicule vers d'autres applications, telles que des AWS Lambda fonctions. Vous pouvez créer une règle de sujet à l'aide de la AWS IoT console ou à l'aide de l'API du plan de `CreateTopicRule` AWS IoT Core contrôle. Pour plus d'informations, consultez la section Création et AWS IoT règle.

   Dans cet exemple, remplacez-le *<CommandID>* par l'identifiant de la commande pour laquelle vous souhaitez recevoir des notifications et *<CommandExecutionStatus>* par le statut de l'exécution de la commande.

   ```
   $aws/events/commandExecution/<CommandID>/<CommandExecutionStatus>
   ```

   > **Note**
   >
   > Pour recevoir des notifications concernant toutes les commandes et les statuts d'exécution des commandes, vous pouvez utiliser des caractères génériques et vous abonner à la rubrique suivante.

```
$aws/events/commandExecution/+/#
```

2. Réception et traitement des événements liés aux commandes

   Si vous avez créé une règle de sujet à l'étape précédente pour vous abonner aux événements liés aux commandes, vous pouvez gérer les notifications push de commandes que vous recevez. Vous pouvez également éventuellement créer des applications dessus, telles que with, Amazon SQS AWS Lambda, Amazon SNS ou Step AWS Functions en utilisant la règle de rubrique que vous avez créée.

Le code suivant montre un exemple de charge utile pour les notifications d'événements de commande que vous recevrez.

```
{
    "executionId": "2bd65c51-4cfd-49e4-9310-d5cbfdbc8554",
    "status":"FAILED",
    "statusReason": {
         "reasonCode": "4",
         "reasonDescription": ""
    },
    "eventType": "COMMAND_EXECUTION",
    "commandArn":"arn:aws:iot:us-east-1:123456789012:command/0b9d9ddf-
e873-43a9-8e2c-9fe004a90086",
    "targetArn":"arn:aws:iot:us-east-1:123456789012:thing/5006c3fc-
de96-4def-8427-7eee36c6f2bd",
    "timestamp":1717708862107
}
```

Création et gestion de commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez configurer des actions à distance réutilisables ou envoyer des instructions immédiates et ponctuelles à vos appareils. Lorsque vous utilisez cette fonctionnalité, vous pouvez spécifier les instructions que vos appareils peuvent exécuter quasiment en temps réel. Une commande vous permet de configurer des actions à distance réutilisables pour votre véhicule cible. Après avoir créé une commande, vous pouvez lancer une exécution de commande qui cible un véhicule spécifique.

Cette rubrique explique comment créer et gérer une ressource de commande à l'aide de l' AWS IoT Core API ou du AWS CLI. Il explique comment effectuer les actions suivantes sur une ressource de commande.

Rubriques

- Création d'une ressource de commande
- Récupérer des informations sur une commande
- Répertorier les commandes de votre compte
- Mettre à jour ou rendre obsolète une ressource de commande
- Supprimer une ressource de commande

Création d'une ressource de commande

Vous pouvez utiliser le fonctionnement de l'API du plan de `CreateCommand` AWS IoT Core contrôle ou la FleetWise console AWS IoT pour créer une commande.

Création d'une commande (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour créer une commande.

Pour créer une commande

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, choisissez Commands.
3. Choisissez Créer une commande.
4. Spécifiez un identifiant de commande unique pour vous aider à identifier la commande que vous souhaitez exécuter sur le véhicule.
5. (Facultatif) Spécifiez un nom d'affichage et une description facultatifs.
6. (Facultatif) Sélectionnez l'actionneur et la valeur du paramètre par défaut. Les paramètres spécifient les actions que le véhicule cible peut effectuer lors de la réception de la commande. Si vous n'ajoutez pas de paramètres, vous devrez les fournir lors de l'exécution de la commande.

7. Choisissez un rôle IAM qui accorde les autorisations nécessaires pour générer la charge utile des commandes. Consultez la section Contrôle de l'accès.
8. Choisissez Créer une commande.

Création d'une commande (AWS CLI)

L'exemple suivant montre comment créer une commande avec un paramètre.

Considérations relatives à la création d'une commande

Lorsque vous créez une commande dans AWS IoT FleetWise :

- Vous devez spécifier celui `roleArn` qui autorise la création et l'exécution de commandes sur votre véhicule. Pour plus d'informations et sur les exemples de politiques, notamment lorsque les clés KMS sont activées, consultezAccorder AWS IoT Device Management l'autorisation de générer la charge utile pour les commandes avec AWS IoT FleetWise.
- Vous devez le spécifier `AWS-IoT-FleetWise` comme espace de noms.
- Vous pouvez ignorer le `mandatory-parameters` champ et les spécifier au moment de l'exécution. Vous pouvez également créer une commande avec des paramètres et éventuellement spécifier des valeurs par défaut pour ceux-ci. Si vous avez spécifié des valeurs par défaut, vous pouvez utiliser ces valeurs lors de l'exécution ou les remplacer en spécifiant vos propres valeurs. Pour ces exemples supplémentaires, voirScénarios d'utilisation des commandes.
- Vous pouvez spécifier jusqu'à trois paires nom-valeur pour le `mandatory-parameters` champ. Toutefois, lors de l'exécution de la commande sur le véhicule, une seule paire nom-valeur est acceptée, et le `name` champ doit utiliser le nom complet avec le `$actuatorPath.` préfixe.

- Remplacez *`command-id`* par un identifiant unique pour la commande. Vous pouvez utiliser l'UUID, les caractères alphanumériques, « - » et « _ ».
- *`role-arn`*Remplacez-le par le rôle IAM qui vous autorise à créer et à exécuter des commandes, par exemple,`"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"`.
- (Facultatif) *`display-name`* Remplacez-le par un nom convivial pour la commande et *`description`* par une description significative de la commande.
- Remplacez *`name`* et *`value`* de l'`mandatory-parameters`objet par les informations requises pour la commande en cours de création. Le `name` champ est le nom complet tel que défini dans le catalogue de signaux avec `$actuatorPath.` comme préfixe. Par exemple, il `name` peut être

$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode et `value` peut être un booléen indiquant un état du mode de direction tel que. *{"B": false}*

```
aws iot create-command --command-id command-id \
    --role-arn role-arn \
    --description description \
    --display-name display-name \
    --namespace "AWS-IoT-FleetWise" \
    --mandatory-parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

L'opération d'`CreateCommand`API renvoie une réponse contenant l'ID et l'ARN (Amazon Resource Name) de la commande.

```
{
    "commandId": "HandsOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"
}
```

Récupérer des informations sur une commande

Vous pouvez utiliser l'opération API du plan de `GetCommand` AWS IoT Core contrôle pour récupérer des informations sur une ressource de commande.

Pour obtenir des informations sur une ressource de commande, exécutez la commande suivante. Remplacez *command-id* par l'identifiant utilisé lors de la création de la commande.

```
aws iot get-command --command-id command-id
```

L'opération `GetCommand` API renvoie une réponse contenant les informations suivantes.

- L'ID et l'ARN (Amazon Resource Name) de la commande.
- Date et heure de création et de dernière mise à jour de la commande.
- État de la commande qui indique si elle est disponible pour fonctionner sur le véhicule.

- Tous les paramètres que vous avez spécifiés lors de la création de la commande.

```
{
    "commandId": "HandsOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode",
            "value": {"B": false }
        }
    ],
    "createdAt": "2024-03-23T11:24:14.919000-07:00",
    "lastUpdatedAt": "2024-03-23T11:24:14.919000-07:00",
    "deprecated": false,
    "pendingDeletion": false
}
```

Répertorier les commandes de votre compte

Vous pouvez utiliser l'opération API du plan de `ListCommands` AWS IoT Core contrôle pour répertorier toutes les commandes que vous avez créées dans votre compte.

Pour répertorier les commandes de votre compte, exécutez la commande suivante. Par défaut, l'API renvoie les commandes créées pour les deux espaces de noms. Pour filtrer la liste afin d'afficher uniquement les commandes créées pour AWS IoT FleetWise, exécutez la commande suivante.

> **Note**
>
> Vous pouvez également trier la liste par ordre croissant ou décroissant, ou filtrer la liste pour n'afficher que les commandes ayant un nom de paramètre de commande spécifique.

```
aws iot list-commands --namespace "AWS-IoT-FleetWise"
```

L'opération `ListCommands` API renvoie une réponse contenant les informations suivantes.

- L'ID et l'ARN (Amazon Resource Name) des commandes.

- Date et heure de création et de dernière mise à jour de la commande.
- État des commandes qui indique si les commandes sont disponibles pour être exécutées sur le véhicule.

Mettre à jour ou rendre obsolète une ressource de commande

Vous pouvez utiliser le fonctionnement de l'API du plan de `UpdateCommand` AWS IoT Core contrôle ou FleetWise la console AWS IoT pour mettre à jour une ressource de commande. Vous pouvez mettre à jour le nom d'affichage et la description d'une commande. Vous pouvez également déprécier une commande si elle n'est pas actuellement utilisée.

> **Note**
>
> Vous ne pouvez pas modifier les informations de l'espace de noms ou les paramètres à utiliser lors de l'exécution de la commande.

Mettre à jour une commande (console)

Mettre à jour une commande

Pour mettre à jour une commande depuis la console, rendez-vous sur la page Commandes de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez la commande que vous souhaitez mettre à jour, puis sélectionnez Modifier.
2. Modifiez les détails de la commande, puis choisissez Enregistrer les modifications.

Déprécier une commande

Pour désactiver une commande depuis la console, rendez-vous sur la page Commandes de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez la commande que vous souhaitez déprécier, puis choisissez Déprécier.
2. Confirmez la dépréciation, puis choisissez Déprécier.

Mettre à jour une commande (AWS CLI)

Mettre à jour une commande

Pour mettre à jour une ressource de commande, exécutez la commande suivante. *command-id*Remplacez-le par l'identifiant de la commande que vous souhaitez mettre à jour et fournissez le *display-name* et mis à jour*description*.

```
aws iot update-command \
    --command-id command-id \
    --display-name display-name \
    --description description
```

L'opération `UpdateCommand` API renvoie la réponse suivante.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": false,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Déprécier une commande

Vous désapprouvez une commande lorsque vous avez l'intention de ne plus continuer à l'utiliser sur votre appareil ou lorsqu'elle est obsolète. L'exemple suivant montre comment déprécier une commande.

```
aws iot update-command \
    --command-id command-id \
    --deprecated
```

L'opération d'`UpdateCommand`API renvoie une réponse contenant l'ID et l'ARN (Amazon Resource Name) de la commande.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": true,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Une fois qu'une commande est devenue obsolète, les exécutions de commandes existantes continuent de s'exécuter sur le véhicule jusqu'à ce qu'elles deviennent un terminal. Pour exécuter de nouvelles commandes, vous devez utiliser l'`UpdateCommand`API pour restaurer la commande

afin qu'elle soit disponible. Pour plus d'informations sur la dépréciation et la restauration d'une commande, ainsi que sur les considérations associées, consultez la section Dépréciation d'une ressource de commande dans le guide du développeur.AWS IoT Core

Supprimer une ressource de commande

Vous pouvez utiliser le fonctionnement de l'API du plan de `DeleteCommand` AWS IoT Core contrôle ou FleetWise la console AWS IoT pour supprimer une ressource de commande.

> **Note**
>
> Les actions de suppression sont permanentes et ne peuvent être annulées. La commande sera définitivement supprimée de votre compte.

Supprimer une commande (console)

Pour supprimer une commande de la console, rendez-vous sur la page Commandes de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez la commande que vous souhaitez supprimer, puis sélectionnez Supprimer.
2. Confirmez que vous souhaitez supprimer la commande, puis choisissez Supprimer.

Supprimer une commande (AWS CLI)

Pour supprimer une ressource de commande, exécutez la commande suivante. Remplacez *`command-id`* par l'identifiant de la commande que vous souhaitez supprimer. L'exemple suivant montre comment supprimer une ressource de commande.

```
aws iot delete-command --command-id command-id
```

Si la demande de suppression aboutit :

- Si la commande est devenue obsolète pendant une durée supérieure au délai maximum de 24 heures, elle sera immédiatement supprimée et vous verrez un HTTP `statusCode` de 204.
- Si la commande n'est pas obsolète, ou si elle l'a été pendant une durée inférieure au délai maximum, la commande sera dans un `pending deletion` état et vous verrez un HTTP de 202.

`statusCode` La commande sera automatiquement supprimée de votre compte après le délai maximum de 24 heures.

Démarrez et surveillez les exécutions de commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Après avoir créé une ressource de commande, vous pouvez démarrer une exécution de commande sur le véhicule cible. Une fois que le véhicule commence à exécuter la commande, il peut commencer à mettre à jour le résultat de l'exécution de la commande et publier des mises à jour de statut et des informations sur les résultats dans les rubriques réservées au MQTT. Vous pouvez ensuite récupérer le statut de l'exécution de la commande et surveiller le statut des exécutions dans votre compte.

Cette rubrique explique comment envoyer une commande à votre véhicule à l'aide de la FleetWise console AWS CLI ou AWS IoT. Il vous montre également comment surveiller et mettre à jour le statut de l'exécution de la commande.

Rubriques

- Mettre à jour le résultat d'exécution des commandes
- Obtenir l'exécution d'une commande
- Répertorier les exécutions de commandes dans votre compte
- Supprimer l'exécution d'une commande

Envoyer une commande (console)

Pour envoyer une commande depuis la console, rendez-vous sur la page Véhicules de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez le véhicule auquel vous souhaitez envoyer une commande.
2. Sélectionnez Run Command (Exécuter la commande).
3. Sélectionnez l'ID de commande.

4. Spécifiez le délai d'exécution de la commande, puis choisissez Exécuter la commande.

Envoyer une commande (AWS CLI)

Vous pouvez utiliser le fonctionnement de l'API du plan de `StartCommandExecution` AWS IoT données pour envoyer une commande à un véhicule. Le véhicule transmet ensuite la commande à un service intergiciel automobile (comme SOME/IP (Scalable Service-Oriented Middleware over IP)) ou la publie sur un réseau du véhicule (comme une interface de périphérique CAN). L'exemple suivant repose sur AWS CLI.

Rubriques

- Considérations relatives à l'envoi d'une commande
- Obtenir le point de terminaison du plan de données spécifique au compte
- Envoyer un exemple de commande

Considérations relatives à l'envoi d'une commande

Lorsque vous lancez l'exécution d'une commande dans AWS IoT FleetWise :

- Vous devez prévoir AWS IoT quoi que ce soit pour le véhicule. Pour de plus amples informations, veuillez consulter Fournir AWS des FleetWise véhicules IoT.
- Vous devez déjà avoir créé une commande avec `AWS-IoT-FleetWise` comme espace de noms et en avoir fourni un `role-Arn` qui vous autorise à créer et exécuter des commandes dans AWS l'IoT FleetWise. Pour de plus amples informations, veuillez consulter Création d'une ressource de commande.
- Vous pouvez ignorer le `parameters` champ si vous choisissez d'utiliser les valeurs par défaut spécifiées pour les paramètres lors de la création de la commande. Si aucune valeur `mandatory-parameters` n'a été spécifiée au moment de la création, ou si vous souhaitez remplacer les valeurs par défaut en spécifiant vos propres valeurs pour les paramètres, vous devez spécifier le `parameters` champ. Pour ces exemples supplémentaires, voirScénarios d'utilisation des commandes.
- Vous pouvez spécifier jusqu'à trois paires nom-valeur pour le `mandatory-parameters` champ. Toutefois, lors de l'exécution de la commande sur le véhicule, une seule paire nom-valeur est acceptée, et le `name` champ doit utiliser le nom complet avec le `$actuatorPath.` préfixe.

Obtenir le point de terminaison du plan de données spécifique au compte

Avant d'exécuter la commande API, vous devez obtenir l'URL du point de terminaison spécifique au compte pour le `iot:Jobs` point de terminaison. Par exemple, si vous exécutez la commande suivante :

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

Il renverra l'URL du point de terminaison spécifique au compte, comme indiqué dans l'exemple de réponse ci-dessous.

```
{
    "endpointAddress": "<account-specific-prefix>.jobs.iot.<region>.amazonaws.com"
}
```

Envoyer un exemple de commande

Pour envoyer une commande à un véhicule, exécutez la commande suivante.

- *`command-arn`*Remplacez-le par l'ARN de la commande que vous souhaitez exécuter. Vous pouvez obtenir ces informations à partir de la réponse de la commande `create-command` CLI.
- *`target-arn`*Remplacez-le par l'ARN de l'appareil cible, ou de l' AWS IoT objet, pour lequel vous souhaitez exécuter la commande.

  > **Note**
  >
  > Vous pouvez spécifier l'ARN cible d'un AWS IoT objet (FleetWise véhicule AWS IoT). Les groupes d'objets et les flottes ne sont actuellement pas pris en charge.

- *`endpoint-url`*Remplacez-le par le point de terminaison spécifique au compte dans lequel vous l'avez obtenuObtenir le point de terminaison du plan de données spécifique au compte, préfixé par`https://`, par exemple,. `https://`*`123456789012abcd`*`.jobs.iot.`*`ap-south-1`*`.amazonaws.com`
- Remplacez *`name`* et *`value`* par le `mandatory-parameters` champ que vous avez spécifié lorsque vous avez créé la commande à l'aide de la `create-command` CLI.

  Le `name` champ est le nom complet tel que défini dans le catalogue de signaux avec `$actuatorPath.` comme préfixe. Par exemple, il `name` peut être

$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode et `value` peut être un booléen indiquant un état du mode de direction tel que. *{"B": false}*

- (Facultatif) Vous pouvez également spécifier un paramètre supplémentaire,`executionTimeoutSeconds`. Ce champ facultatif indique le délai en secondes pendant lequel le périphérique doit répondre avec le résultat de l'exécution. Vous pouvez configurer le délai d'attente à une valeur maximale de 24 heures.

  Lorsque l'exécution de la commande a été créée, un temporisateur démarre. Avant l'expiration du délai, si l'état d'exécution de la commande ne passe pas à un état qui la rend terminale, tel que `SUCCEEDED` ou`FAILED`, alors le statut passe automatiquement à`TIMED_OUT`.

  > **Note**
  >
  > L'appareil peut également signaler un `TIMED_OUT` état ou remplacer cet état par un état tel que`SUCCEEDED`, ou `FAILEDREJECTED`, et l'exécution de la commande deviendra terminale. Pour de plus amples informations, veuillez consulter État du délai d'exécution des commandes.

```
aws iot-jobs-data start-command-execution \
    --command-arn command-arn \
    --target-arn target-arn \
    --execution-timeout-seconds 30 \
    --endpoint-url endpoint-url \
    --parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

L'opération `StartCommandExecution` API renvoie un ID d'exécution de commande. Vous pouvez utiliser cet ID pour demander le statut, les détails et l'historique de l'exécution des commandes.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
 }
```

Après avoir exécuté la commande, vos appareils recevront une notification contenant les informations suivantes. Le `issued_timestamp_ms` champ correspond à l'heure à laquelle l'`StartCommandExecution`API a été appelée. Le `timeout_ms` correspond à la valeur du délai d'expiration configurée à l'aide du `executionTimeoutSeconds` paramètre lors de l'appel de l'`StartCommandExecution`API.

```
timeout_ms: 9000000
issued_timestamp_ms: 1723847831317
```

Mettre à jour le résultat d'exécution des commandes

Pour mettre à jour le statut de l'exécution de la commande, votre appareil doit avoir établi une connexion MQTT et s'être abonné à la rubrique de demande de commandes suivante.

Dans cet exemple, remplacez-le *<device-id>* par l'identifiant unique de votre équipement cible, qui peut être le `VehicleId` nom de l'objet, et *<execution-id>* par l'identifiant de l'exécution de la commande.

> **Note**
>
> - La charge utile doit utiliser le format protobuf.
> - Il est facultatif pour vos appareils de s'abonner aux rubriques `/accepted` et `/rejected` aux réponses. Vos appareils recevront ces messages de réponse même s'ils ne s'y sont pas explicitement abonnés.

```
// Request topic
$aws/devices/<DeviceID>/command_executions/+/request/protobuf

// Response topics (Optional)
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/accepted/protobuf
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/rejected/protobuf
```

Votre appareil peut publier un message dans la rubrique de réponse aux commandes. Après avoir traité la commande, elle envoie une réponse codée en protobuf à cette rubrique. Le *<DeviceID>* champ doit correspondre au champ correspondant dans le sujet de la demande.

```
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/<PayloadFormat>
```

Une fois que votre appareil a publié une réponse à cette rubrique, vous pouvez récupérer les informations d'état mises à jour à l'aide de l'`GetCommandExecution`API. Le statut de l'exécution d'une commande peut être l'un de ceux listés ici.

- `IN_PROGRESS`
- `SUCCEEDED`
- `FAILED`
- `REJECTED`
- `TIMED_OUT`

Notez qu'une exécution de commande dans l'un des statuts `SUCCEEDEDFAILED`, et `REJECTED` est terminale, et que l'état est signalé par le périphérique. Lorsqu'une commande est exécutée en mode terminal, cela signifie qu'aucune autre mise à jour ne sera apportée à son statut ou aux champs associés. Un `TIMED_OUT` état peut être signalé par l'appareil ou par le cloud. En cas de signalement par le cloud, une mise à jour du champ de motif du statut peut être effectuée ultérieurement par l'appareil.

Par exemple, ce qui suit montre un exemple de message MQTT publié par le périphérique.

> **Note**
>
> En ce qui concerne l'état d'exécution de la commande, si vos appareils utilisent l'`statusReason`objet pour publier les informations d'état, vous devez vous assurer que :
>
> - `reasonCode`Utilise le modèle`[A-Z0-9_-]+`, et sa longueur ne dépasse pas 64 caractères.
> - La longueur `reasonDescription` ne dépasse pas 1 024 caractères. Il peut utiliser n'importe quel caractère à l'exception des caractères de contrôle tels que les nouvelles lignes.

```
{
    "deviceId": "",
    "executionId": "",
    "status": "CREATED",
    "statusReason": {
        "reasonCode": "",
```

```
        "reasonDescription": ""
    }
}
```

Pour un exemple montrant comment vous pouvez utiliser le client de test AWS IoT Core MQTT pour vous abonner aux rubriques et voir les messages d'exécution des commandes, consultez la section Affichage des mises à jour des commandes à l'aide du client de test MQTT dans le guide du AWS IoT Core développeur.

Obtenir l'exécution d'une commande

Vous pouvez utiliser l'opération API du plan de `GetCommandExecution` AWS IoT contrôle pour récupérer des informations sur l'exécution d'une commande. Vous devez déjà avoir exécuté cette commande à l'aide de l'opération `StartCommandExecution` API.

Pour récupérer les métadonnées d'une commande exécutée, exécutez la commande suivante.

- Remplacez *`execution-id`* par l'ID de la commande. Vous pouvez obtenir ces informations à partir de la réponse de la commande `start-command-execution` CLI.
- *`target-arn`*Remplacez-le par l'ARN du véhicule ou de l' AWS IoT objet cible pour lequel vous souhaitez exécuter la commande.

```
aws iot get-command-execution --execution-id execution-id \
    --target-arn target-arn
```

L'opération `GetCommandExecution` API renvoie une réponse contenant des informations sur l'ARN de l'exécution de la commande, le statut de l'exécution, ainsi que l'heure à laquelle la commande a commencé à s'exécuter et à laquelle elle s'est terminée. Le code suivant montre un exemple de réponse à la demande d'API.

Pour fournir un contexte supplémentaire sur l'état de chaque exécution de commande, la fonctionnalité des commandes fournit un `statusReason` objet. L'objet contient deux champs, `reasonCode` et`reasonDescription`. À l'aide de ces champs, vos appareils peuvent fournir des informations supplémentaires sur le statut de l'exécution d'une commande. Ces informations remplaceront toute valeur par défaut `reasonCode` `reasonDescription` signalée depuis le cloud.

Pour signaler ces informations, vos appareils peuvent publier les informations d'état mises à jour sur le cloud. Ensuite, lorsque vous récupérerez l'état d'exécution de la commande à l'aide de l'`GetCommandExecution`API, vous verrez les derniers codes d'état.

> **Note**
>
> Le `completedAt` champ de la réponse d'exécution correspond au moment où l'appareil signale l'état du terminal au cloud. En cas d'`TIMED_OUT`état, ce champ ne sera défini que lorsque l'appareil signalera un délai d'expiration. Lorsque le `TIMED_OUT` statut est défini par le cloud, il n'est pas mis à jour. `TIMED_OUT` Pour plus d'informations sur le comportement en cas de temporisation, consultezÉtat du délai d'exécution des commandes.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/LockDoor",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myFrontDoor",
    "status": "SUCCEEDED",
    "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "Parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode":

         { "B": true }
    }'
}
```

Répertorier les exécutions de commandes dans votre compte

Utilisez l'opération d'API HTTP du plan de `ListCommandExecutions` AWS IoT Core contrôle pour répertorier toutes les exécutions de commandes dans votre compte. L'exemple utilise le AWS CLI.

Rubriques

- Considérations relatives à la liste des exécutions de commandes
- Exemple d'exécution de commandes de liste

Considérations relatives à la liste des exécutions de commandes

Voici quelques points à prendre en compte lors de l'utilisation de l'`ListCommandExecutions`API.

- Vous devez spécifier au moins le `targetArn` ou le `commandArn` selon que vous souhaitez répertorier les exécutions pour une commande particulière ou pour un véhicule cible. La demande d'API ne peut pas être vide et ne peut pas contenir les deux champs dans la même demande.
- Vous ne devez fournir que les informations `startedTimeFilter` ou les `completedTimeFilter` informations. La demande d'API ne peut pas être vide et ne peut pas contenir les deux champs dans la même demande. Vous pouvez utiliser les `after` champs `before` et de l'objet pour répertorier les exécutions de commandes créées ou achevées au cours d'une période donnée.
- Les `after` champs `before` et ne doivent pas être supérieurs à l'heure actuelle. Par défaut, si vous ne spécifiez aucune valeur, le `before` champ correspond à l'heure actuelle et le `after` champ à l'heure actuelle, soit 6 mois. En d'autres termes, en fonction du filtre que vous utilisez, l'API répertorie toutes les exécutions créées ou achevées au cours des six derniers mois.
- Vous pouvez utiliser le `sort-order` paramètre pour indiquer si vous souhaitez répertorier les exécutions dans l'ordre croissant. Par défaut, les exécutions seront répertoriées par ordre décroissant si vous ne spécifiez pas ce champ.
- Vous ne pouvez pas filtrer les exécutions de commandes en fonction de leur statut lorsque vous listez les exécutions de commandes pour un ARN de commande.

Exemple d'exécution de commandes de liste

L'exemple suivant vous montre comment répertorier les exécutions de commandes dans votre Compte AWS.

Lorsque vous exécutez la commande, vous devez spécifier s'il faut filtrer la liste pour afficher uniquement les exécutions de commandes créées pour un appareil particulier à l'aide du`targetArn`, ou les exécutions pour une commande spécifique spécifiée à l'aide du`commandArn`.

Dans cet exemple, remplacez :

- *<target-arn>*avec le numéro de ressource Amazon (ARN) de l'appareil pour lequel vous ciblez l'exécution, tel que`arn:aws:iot:`*us-east-1*`:`*123456789012*`:thing/`*b8e4157c98f332cffb37627f*.
- *<target-arn>*avec le numéro de ressource Amazon (ARN) de l'appareil pour lequel vous ciblez l'exécution, tel que`arn:aws:iot:`*us-east-1*`:`*123456789012*`:thing/`*b8e4157c98f332cffb37627f*.

- *<after>*avec le délai après lequel vous souhaitez répertorier les exécutions créées, par exemple,`2024-11-01T03:00`.

```
aws iot list-command-executions \
--target-arn <target-arn> \
--started-time-filter '{after=<after>}' \
--sort-order "ASCENDING"
```

L'exécution de cette commande génère une réponse qui contient une liste des exécutions de commandes que vous avez créées, ainsi que l'heure à laquelle les exécutions ont commencé à s'exécuter et à laquelle elles se sont terminées. Il fournit également des informations sur le statut, ainsi que l'`statusReason`objet qui contient des informations supplémentaires sur le statut.

```
{
    "commandExecutions": [
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "b2b654ca-1a71-427f-9669-e74ae9d92d24",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "TIMED_OUT",
            "createdAt": "2024-11-24T14:39:25.791000-08:00",
            "startedAt": "2024-11-24T14:39:25.791000-08:00"
        },
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "34bf015f-ef0f-4453-acd0-9cca2d42a48f",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "IN_PROGRESS",
            "createdAt": "2024-11-24T14:05:36.021000-08:00",
            "startedAt": "2024-11-24T14:05:36.021000-08:00"
        }
    ]
}
```

Supprimer l'exécution d'une commande

Si vous ne souhaitez plus utiliser l'exécution d'une commande, vous pouvez la supprimer définitivement de votre compte.

> **Note**
>
> L'exécution d'une commande ne peut être supprimée que si elle est entrée dans un statut de terminalSUCCEEDED, tel queFAILED, ouREJECTED.

L'exemple suivant montre comment supprimer une exécution de commande à l'aide de cette `delete-command-execution` AWS CLI commande. *<execution-id>*Remplacez-le par l'identifiant de l'exécution de la commande que vous supprimez.

```
aws iot delete-command-execution --execution-id <execution-id>
```

Si la demande d'API aboutit, l'exécution de la commande génère un code d'état de 200. Vous pouvez utiliser l'`GetCommandExecution`API pour vérifier que l'exécution de la commande n'existe plus dans votre compte.

Exemple : utilisation de commandes pour contrôler le mode de direction d'un véhicule (AWS CLI)

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

L'exemple suivant montre comment utiliser la fonctionnalité de commandes à l'aide du AWS CLI. Cet exemple utilise un AWS IoT FleetWise véhicule comme équipement cible pour montrer comment envoyer une commande pour contrôler à distance le mode de direction.

Rubriques

- Vue d'ensemble de l'exemple du mode de direction du véhicule
- Prérequis
- Politique IAM pour l'utilisation des commandes à distance
- Exécuter AWS IoT des commandes (AWS CLI)
- Nettoyage

Vue d'ensemble de l'exemple du mode de direction du véhicule

Dans cet exemple, vous allez :

1. Créez une ressource de commande pour l'opération en utilisant le `create-command` AWS CLI pour modifier le mode de direction du véhicule.
2. Récupérez des informations sur la commande, telles que l'heure à laquelle elle a été créée ou mise à jour pour la dernière fois à l'aide du `get-command` AWS CLI.
3. Envoyez la commande au véhicule en utilisant le `start-command-execution` AWS CLI mode de direction comme paramètre obligatoire, qui sera ensuite exécutée sur l'appareil.
4. Obtenez le résultat de l'exécution de la commande à l'aide du `get-command-execution` AWS CLI. Vous pouvez vérifier quand l'exécution est terminée et récupérer des détails supplémentaires tels que le résultat de l'exécution et le temps nécessaire pour terminer l'exécution de la commande.
5. Effectuez des activités de nettoyage en supprimant les commandes et les exécutions de commandes que vous ne souhaitez plus utiliser.

Prérequis

Avant d'exécuter cet exemple :

- Provisionnez votre AWS IoT FleetWise véhicule en tant qu' AWS IoT objet inscrit dans le AWS IoT registre. Vous devez également ajouter un certificat à votre objet, l'activer et associer une politique à votre objet. Votre appareil peut ensuite se connecter au cloud et exécuter les commandes. Pour plus d'informations, voir Provisionner des véhicules.
- Créez un utilisateur IAM et une politique IAM qui vous autorise à effectuer les opérations d'API pour utiliser des commandes, comme indiqué dans. Politique IAM pour l'utilisation des commandes à distance

Politique IAM pour l'utilisation des commandes à distance

Le tableau suivant présente un exemple de politique IAM qui accorde l'accès à toutes les opérations d'API du plan de contrôle et du plan de données pour la fonctionnalité de commandes. L'utilisateur de l'application sera autorisé à effectuer toutes les opérations de l'API de commande à distance, comme indiqué dans le tableau.

Opération API

Action d'API	Plan de contrôle/de données	Protocole	Description	Ressource
`CreateCommand`	Plan de contrôle	HTTP	Crée une ressource de commande	• command
`GetCommand`	Plan de contrôle	HTTP	Récupère les informations relatives à une commande	• command
`UpdateCommand`	Plan de contrôle	HTTP	Met à jour les informations relatives à une commande ou pour la rendre obsolète	• command
`ListCommands`	Plan de contrôle	HTTP	Répertorie les commandes de votre compte	• command
`DeleteCommand`	Plan de contrôle	HTTP	Supprime une commande	• command
`StartCommandExecution`	Plan de données	HTTP	Lance l'exécution d'une commande	• command • thing
`UpdateCommandExecution`	Plan de données	MQTT	Mettre à jour l'exécution d'une commande	• command • thing
`GetCommandExecution`	Plan de contrôle	HTTP	Récupère les informations relatives à l'exécution d'une commande	• command • thing
`ListCommandExecutions`	Plan de contrôle	HTTP	Répertorie les exécutions de commandes dans votre compte	• command • thing

Action d'API	Plan de contrôle/de données	Protocole	Description	Ressource
`DeleteCommandExecution`	Plan de contrôle	HTTP	Supprime l'exécution d'une commande	• command • thing

Dans cet exemple, remplacez :

- *region*avec votre Région AWS, par exemple`ap-south-1`.
- *account-id*avec votre Compte AWS numéro, par exemple`57EXAMPLE833`.
- *command-idcommand-id1*, et *command-id2* avec votre identifiant de commande unique, tel que `LockDoor` ou`TurnOffAC`.
- *thing-name*avec le nom de votre AWS IoT objet, tel que`my_car`.

```
{
    "Version": "2012-10-17",
    "Statement": [
     {
        "Action": [
            "iot:CreateCommand",
            "iot:GetCommand",
            "iot:ListCommands",
            "iot:UpdateCommand",
            "iot:DeleteCommand"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id1",
            "arn:aws:iot:<region>:<account-id>:command/command-id2",
        ]
     },
     {
        "Action": [
            "iot:GetCommandExecution",
            "iot:ListCommandExecutions",
            "iot:DeleteCommandExecution"
```

```
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
        ]
     },
     {
        "Action": "iot:StartCommandExecution",
        "Effect": "Allow",
        "Resource": [
            "arn:aws:iot:<region>:<account-id>:command/command-id",
            "arn:aws:iot:<region>:<account-id>:thing/thing-name",
        ]
     }
   ]
}
```

Exécuter AWS IoT des commandes (AWS CLI)

Ce qui suit montre comment vous pouvez utiliser le AWS CLI pour exécuter des commandes et modifier le mode de direction du véhicule.

1. Création d'une ressource de commande pour le fonctionnement du mode de direction

   Créez la commande que vous souhaitez envoyer à votre appareil à l'aide de la `create-command` CLI. Dans cet exemple, spécifiez :

   - `command-id` comme *`TurnOffSteeringMode`*
   - `role-arn`car "`arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*" le rôle IAM `role-arn` doit être fourni, car c'est le rôle IAM qui accorde les autorisations nécessaires pour créer et exécuter des commandes sur votre véhicule. Pour de plus amples informations, veuillez consulter Accorder AWS IoT Device Management l'autorisation de générer la charge utile pour les commandes avec AWS IoT FleetWise.
   - `display-name`sous la forme de « *`Turn off steering mode`* »
   - `namespace`doit être `AWS-IoT-FleetWise`
   - `mandatory-parameters`sous forme de paire nom-valeur, avec « *`$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode`* » et DefaultValue `name` comme *`{ "S": "true" }`*

> **Note**
>
> Vous pouvez également créer une commande sans spécifier de paramètres obligatoires. Vous devez ensuite spécifier les paramètres à utiliser lors de l'exécution de la commande à l'aide de la `start-command-execution` CLI. Pour obtenir un exemple, consultez Scénarios d'utilisation des commandes.

> **Important**
>
> Lorsque vous utilisez l'`AWS-IoT-FleetWise`espace de noms, vous devez vous assurer que le `Name` champ spécifié dans le cadre de celui-ci `mandatory-parameters` utilise le `$actuatorPath.` préfixe et que le `Value` champ doit utiliser le type de données chaîne.

```
aws iot create-command \
    --command-id TurnOffSteeringMode \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --display-name "Turn off steering mode" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
        "name": "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
        "defaultValue": { "S": "true" }
      }
    ]'
```

La sortie suivante montre un exemple de réponse de la CLI, où `ap-south-1` et `123456789012` sont des exemples de l' Compte AWS ID Région AWS and.

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode"
}
```

Pour d'autres exemples d'utilisation de cette commande, consultezCréation d'une ressource de commande.

2. Récupérez les informations relatives à la commande

Exécutez la commande suivante pour récupérer des informations sur la commande, où se `command-id` trouve l'ID de commande dans le résultat de l'`create-command`opération ci-dessus.

> **Note**
>
> Si vous créez plusieurs commandes, vous pouvez utiliser l'`ListCommands`API pour répertorier toutes les commandes de votre compte, puis utiliser l'`GetCommand`API pour obtenir des informations supplémentaires sur une commande spécifique. Pour de plus amples informations, veuillez consulter Répertorier les commandes de votre compte.

```
aws iot get-command --command-id TurnOffSteeringMode
```

L'exécution de cette commande génère la réponse suivante. Vous verrez l'heure à laquelle la commande a été créée et quand elle a été mise à jour pour la dernière fois, les paramètres que vous avez spécifiés et si la commande est disponible pour être exécutée sur l'appareil.

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
            "defaultValue": {"S": "true" }
        }
    ],
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "lastUpdatedAt": "2024-03-23T00:50:10.095000-07:00",
    "deprecated": false
}
```

Pour d'autres exemples d'utilisation de cette commande, consultezRécupérer des informations sur une commande.

3. Lancer l'exécution de la commande

   Exécutez la commande suivante pour commencer à exécuter la commande, où se `command-arn` trouve l'ARN de la commande dans le résultat de l'`get-command`opération ci-dessus. `target-arn`Il s'agit de l'ARN de l'appareil cible pour lequel vous exécutez la commande, par exemple*myVehicle*.

   Dans cet exemple, puisque vous avez fourni des valeurs par défaut pour les paramètres lors de la création de la commande, la `start-command-execution` CLI peut utiliser ces valeurs lors de l'exécution de la commande. Vous pouvez également choisir de remplacer la valeur par défaut en spécifiant une valeur différente pour les paramètres lorsque vous utilisez la CLI.

   ```
   aws iot-data start-command-execution \
       --command-arn arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode \
       --target-arn arn:aws:iot:ap-south-1:123456789012:thing/myVehicle
   ```

   L'exécution de cette commande renvoie un ID d'exécution de commande. Vous pouvez utiliser cet ID pour demander le statut, les détails et l'historique de l'exécution des commandes.

   ```
   {
       "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
   }
   ```

   Pour des exemples supplémentaires d'utilisation de la CLI, consultezEnvoyer une commande (AWS CLI).

4. Récupère les informations relatives à l'exécution de la commande

   Exécutez la commande suivante pour récupérer les informations relatives à la commande que vous avez exécutée sur le périphérique cible. Spécifiez le`execution-id`, que vous avez obtenu comme résultat de l'`start-command-execution`opération ci-dessus, et le`target-arn`, qui est l'ARN de l'appareil que vous ciblez.

   > **Note**
   >
   > - Pour obtenir les dernières informations d'état, vos appareils doivent avoir publié les informations d'état mises à jour dans la rubrique de réponse réservée MQTT pour les commandes utilisant l'API `UpdateCommandExecution` MQTT. Pour de plus amples informations, veuillez consulter Mettre à jour le résultat d'exécution des commandes.

- Si vous lancez plusieurs exécutions de commandes, vous pouvez utiliser l'`ListCommandExecutions`API pour répertorier toutes les exécutions de commandes de votre compte, puis utiliser l'`GetCommandExecution`API pour obtenir des informations supplémentaires sur une exécution spécifique. Pour de plus amples informations, veuillez consulter Répertorier les exécutions de commandes dans votre compte.

```
aws iot get-command-execution \
    --execution-id <"07e4b780-7eca-4ffd-b772-b76358da5542"> \
    --target-arn arn:aws:iot:<region>:<account>:thing/myVehicle
```

L'exécution de cette commande renvoie des informations sur l'exécution de la commande, son état d'exécution, l'heure à laquelle elle a commencé à s'exécuter et l'heure à laquelle elle s'est terminée. Par exemple, la réponse suivante indique que l'exécution de la commande a réussi sur l'équipement cible et que le mode de direction a été désactivé.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myVehicle",
    "result": "SUCCEEDED",
     "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "result": {
        "KeyName": {
            "S": "",
            "B": true,
            "BIN": null
        }
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode":
         { "S": "true" }
    }'
```

```
}
```

Nettoyage

Maintenant que vous avez créé une commande et que vous l'avez exécutée sur votre appareil, vous pouvez la supprimer si vous n'avez plus l'intention de l'utiliser. Toutes les exécutions de commandes en cours continueront de s'exécuter sans être affectées par la demande de suppression.

> **Note**
>
> Vous pouvez également déprécier une commande si elle est obsolète et vous devrez peut-être l'utiliser ultérieurement pour l'exécuter sur le périphérique cible.

1. (Facultatif) Déprécier la ressource de commande

   Exécutez la commande suivante pour la rendre obsolète, où se `command-id` trouve l'ID de commande dans le résultat de l'`get-command`opération ci-dessus.

   ```
   aws iot update-command \
      --command-id TurnOffSteeringMode \
      --deprecated
   ```

   L'exécution de cette commande renvoie un résultat indiquant que la commande est obsolète. Vous pouvez également utiliser la CLI pour restaurer la commande.

   > **Note**
   >
   > Vous pouvez également utiliser la `update-command` CLI pour mettre à jour le nom d'affichage et la description d'une commande. Pour plus d’informations, consultez Mettre à jour ou rendre obsolète une ressource de commande.

   ```
   {
       "commandId": "TurnOffSteeringMode",
       "deprecated": true,
       "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
   ```

```
}
```

2. Supprimer la commande

   Exécutez la commande suivante pour supprimer la commande, spécifiée par le`command-id`.

   > **Note**
   >
   > L'action de suppression est permanente et ne peut pas être annulée.

   ```
   aws iot delete-command --command-id TurnOffSteeringMode
   ```

   Si la demande de suppression aboutit, vous verrez un HTTP `statusCode` de 202 ou 204 selon que vous avez marqué la commande comme obsolète ou non et selon la date à laquelle elle a été abandonnée. Pour plus d'informations et pour voir un exemple, consultez Supprimer une ressource de commande.

   Vous pouvez utiliser la `get-command` CLI pour vérifier que la commande a été supprimée de votre compte.

3. (Facultatif) Supprimez les exécutions de commandes

   Par défaut, toutes les exécutions de commandes seront supprimées dans les six mois suivant leur création. Vous pouvez consulter ces informations à l'aide du `timeToLive` paramètre de l'`GetCommandExecution`API.

   Sinon, si l'exécution de votre commande est devenue terminale, par exemple si votre statut d'exécution est l'un des `SUCCEEDED` suivants :`FAILED`, ou`REJECTED`, vous pouvez supprimer l'exécution de la commande. Exécutez la commande suivante pour supprimer l'exécution, où se `execution-id` trouve l'ID d'exécution dans le résultat de l'`get-command-execution`opération ci-dessus.

   ```
   aws iot delete-command-execution \
               --execution-id "07e4b780-7eca-4ffd-b772-b76358da5542"
   ```

   Vous pouvez utiliser la `get-command-execution` CLI pour vérifier que l'exécution de la commande a été supprimée de votre compte.

Scénarios d'utilisation des commandes

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Lorsque vous utilisez la fonctionnalité de commandes, vous pouvez créer et exécuter des commandes dans les scénarios suivants :

- Vous pouvez omettre les paramètres lors de la création et spécifier uniquement l'ID de commande. Dans ce cas, vous devez spécifier les paramètres à utiliser lors de l'exécution de la commande sur le périphérique cible.
- Vous pouvez spécifier un ou plusieurs paramètres et configurer leurs valeurs par défaut lors de la création de la commande. En fournissant des valeurs par défaut, vous éviterez d'envoyer des commandes inexactes.
- Vous pouvez spécifier un ou plusieurs paramètres et configurer leurs valeurs lors de la création de la commande. Plusieurs paramètres peuvent être fournis, mais un seul d'entre eux sera exécuté, et le `Name` champ de ce paramètre doit utiliser le `$actuatorPath` préfixe.

Cette section fournit des scénarios d'utilisation pour l'`CreateCommand`API `StartCommandExecution` et l'utilisation des paramètres. Il présente également quelques exemples d'utilisation de commandes avec des modèles d'état.

Rubriques

- Création d'une commande sans paramètres
- Création d'une commande avec des valeurs par défaut pour les paramètres
- Création d'une commande avec des valeurs de paramètres
- Utilisation de commandes avec des modèles d'état

Création d'une commande sans paramètres

Le cas d'utilisation suivant montre comment vous pouvez utiliser l'`CreateCommand`API ou la `create-command` CLI pour créer une commande sans paramètres. Lorsque vous créez une commande, il vous suffit de fournir un ID de commande et un ARN de rôle.

Ce cas d'utilisation est particulièrement utile dans les cas d'utilisation récurrents, par exemple lorsque vous souhaitez envoyer la même commande plusieurs fois à un véhicule. Dans ce cas, la commande n'est pas liée à un actionneur spécifique et vous donne la flexibilité d'exécuter la commande sur n'importe quel actionneur. Vous devez plutôt spécifier les paramètres au moment de l'exécution lorsque vous exécutez la commande à l'aide de l'`StartCommandExecution`API ou de la `start-command-execution` CLI, qui inclut les actionneurs et les valeurs des signaux physiques.

Création d'une commande sans **`mandatory-parameters`** saisie

Ce cas d'utilisation montre comment créer une commande sans saisie de paramètres obligatoires.

```
aws iot create-command \
    --command-id "UserJourney1" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --description "UserJourney1 - No mandatory parameters" \
    --namespace "AWS-IoT-FleetWise"
```

Exécution d'une commande créée sans **`mandatory-parameters`** saisie

Dans ce premier exemple, la commande créée ci-dessus vous permet d'exécuter une commande sur n'importe quel actionneur sans restrictions. Pour `actuator1` définir une valeur de 10, exécutez :

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

De même, vous pouvez exécuter une commande dont `actuator3` la valeur est définie sur`true`.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney1 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
```

```
        "$actuatorPath.Vehicle.actuator3": {"S": "true"}
    }'
```

Création d'une commande avec des valeurs par défaut pour les paramètres

Cette commande vous permet uniquement d'exécuter une commande sur l'actionneur spécifié. En fournissant des valeurs par défaut, vous éviterez d'envoyer des commandes inexactes. Par exemple, une `LockDoor` commande qui verrouille et déverrouille des portes peut être configurée avec une valeur par défaut afin d'éviter que la commande ne déverrouille accidentellement des portes.

Ce cas d'utilisation est particulièrement utile lorsque vous souhaitez envoyer la même commande plusieurs fois et effectuer différentes actions sur le même actionneur, telles que le verrouillage et le déverrouillage des portières d'un véhicule. Si vous souhaitez régler l'actionneur sur la valeur par défaut, vous n'avez pas besoin d'en transmettre `parameters` à la `start-command-execution` CLI. Si vous spécifiez une valeur différente pour le `parameters` dans la `start-command-execution` CLI, elle remplacera la valeur par défaut.

Création d'une commande avec des valeurs par défaut pour **`mandatory-parameters`**

La commande suivante montre comment fournir une valeur par défaut pour actuator1.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1",
            "defaultValue": {"S": "0"}
        }
    ]'
```

Exécution d'une commande créée avec des valeurs par défaut pour **`mandatory-parameters`**

La commande vous `UserJourney2` permet d'exécuter une commande sans avoir à transmettre de valeur d'entrée pendant l'exécution. Dans ce cas, l'exécution au moment de l'exécution utilisera les valeurs par défaut spécifiées lors de la création.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
```

```
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle
```

Vous pouvez également transmettre une valeur différente pour le même actionneur, actuator1, pendant l'exécution, qui remplacera la valeur par défaut.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney3 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "139"}
    }'
```

Création d'une commande avec des valeurs de paramètres

Cette commande vous permet uniquement d'exécuter une commande sur l'actionneur spécifié. Cela vous oblige également à définir une valeur pour l'actionneur pendant l'exécution.

Ce cas d'utilisation est particulièrement utile lorsque vous souhaitez que l'utilisateur final n'effectue que certaines actions spécifiées sur certains actionneurs lorsqu'il les fait fonctionner sur le véhicule.

> **Note**
>
> Vous pouvez avoir plus que des paires nom-valeur pour l'`mandatory-parameters`entrée, avec des valeurs par défaut pour certaines ou toutes. Au moment de l'exécution, vous pouvez ensuite déterminer le paramètre que vous souhaitez utiliser lors de l'exécution de l'actionneur, à condition que le nom de l'actionneur utilise le nom complet avec le `$actuatorPath.` préfixe.

Création d'une commande sans valeurs par défaut pour **`mandatory-parameters`**

Cette commande vous permet uniquement d'exécuter une commande sur l'actionneur spécifié. Cela vous oblige également à définir une valeur pour l'actionneur pendant l'exécution.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
```

```
            "name": "$actuatorPath.Vehicle.actuator1"
        }
    ]'
```

Exécution d'une commande créée sans valeurs par défaut pour **`mandatory-parameters`**

Lors de l'exécution de la commande, dans ce cas, vous devez spécifier une valeur pour actionneur1. L'exécution de la commande illustrée ci-dessous définira avec succès la valeur de `actuator1` to`10`.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/UserJourney2 \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

Utilisation de commandes avec des modèles d'état

Vous pouvez également utiliser les opérations de l'API de commandes pour la collecte et le traitement des données d'état. Par exemple, vous pouvez récupérer un instantané d'état unique ou activer ou désactiver des modèles d'état pour démarrer ou arrêter la collecte de données sur l'état du véhicule. Les exemples suivants montrent comment utiliser la fonctionnalité de commandes avec des modèles d'état. Pour de plus amples informations, consultez Opérations relatives aux modèles d'état pour la collecte et le traitement des données.

> **Note**
>
> Le champ Nom spécifié dans le cadre de la `mandatory-parameters` saisie doit utiliser le `$stateTemplate` préfixe.

Exemple 1 : création de commandes pour des modèles d'état avec des valeurs par défaut

Cet exemple montre comment utiliser la `create-command` CLI pour activer des modèles d'état.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
```

```
      {
          "name": "$stateTemplate.name"
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      }
    ]'
```

De même, la commande suivante montre un exemple de la manière dont vous pouvez utiliser la `start-command-execution` CLI pour les modèles d'état.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
       "$stateTemplate.name": {"S": "ST345"}
    }'
```

Exemple 2 : Création de commandes pour des modèles d'états sans valeurs par défaut

La commande suivante crée plusieurs modèles d'états sans valeurs par défaut pour aucun des paramètres. Cela vous oblige à exécuter la commande avec ces paramètres et leurs valeurs.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name",
          "defaultValue": {"S": "ST123"}
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      },
      {
          "name": "$stateTemplate.deactivateAfterSeconds",
          "defaultValue": {"L": "120"}
      }
    ]'
```

La commande suivante montre comment utiliser la `start-command-execution` CLI dans l'exemple ci-dessus.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME> \
    --parameters '{
        "$stateTemplate.name": {"S": "ST345"},
        "$stateTemplate.operation": {"S": "activate"},
        "$stateTemplate.deactivateAfterSeconds" : {"L": "120"}
```

Surveillez le dernier état connu de vos véhicules

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez surveiller le dernier état connu de vos véhicules en temps quasi réel en créant des modèles d'état et en les associant à vos véhicules. Les véhicules associés aux modèles d'état diffusent des données de télémétrie avec une stratégie `onChange` ou une stratégie de `periodic` mise à jour. Dans le cadre d'une stratégie de mise à jour en cas de modification, les véhicules associés diffusent des données de télémétrie en cas de modification. Dans le cadre d'une stratégie de mise à jour périodique, les véhicules associés diffusent des données de télémétrie pendant une période spécifiée.

Avec les opérations à la demande, vous pouvez demander l'état actuel du véhicule en une seule fois (fetch). Vous pouvez également activer ou désactiver les modèles d'état précédemment déployés pour commencer ou arrêter de communiquer les données sur l'état du véhicule. Les dernières opérations sur l'état connu sont effectuées à l'aide de la AWS IoT commande APIs.

Chaque modèle d'état contient les informations suivantes.

`name`

Alias unique du modèle d'état.

`signalCatalogArn`

Le nom de ressource Amazon (ARN) du catalogue de signaux associé au modèle d'état.

`stateTemplateProperties`

Liste des signaux à partir desquels les données sont collectées. Les propriétés du modèle d'état déterminent les mises à jour de signal spécifiques que le véhicule envoie au cloud.

`dataExtraDimensions`

Une liste d'attributs du véhicule à inclure dans les tampons de protocole (Protobuf) a codé les données traitées.

`metadataExtraDimensions`

Une liste des attributs du véhicule à publier avec les données traitées en tant que propriété utilisateur MQTT 5.

`id`

Identifiant unique généré par le service.

Pour connaître les méthodes de collecte des données envoyées par un véhicule utilisant le FleetWise logiciel Edge Agent for AWS IoT, voirTraitez les dernières données connues du véhicule à l'aide de la messagerie MQTT. Pour plus d'informations sur la façon d'associer un modèle d'état à un véhicule, consultezCréez un FleetWise véhicule AWS IoT.

Rubriques

- Création d'un modèle d' FleetWise état de l' AWS IoT
- Mettre à jour un modèle d' FleetWise état de l' AWS IoT
- Supprimer un modèle FleetWise d'état AWS IoT
- Obtenir des informations sur le modèle FleetWise d'état de l' AWS IoT
- Opérations relatives aux modèles d'état pour la collecte et le traitement des données

Création d'un modèle d' FleetWise état de l' AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l' FleetWise API ou la console AWS IoT pour créer un modèle d'état. Les modèles d'état fournissent un mécanisme permettant de suivre l'état de vos véhicules. Le FleetWise logiciel Edge Agent pour AWS l'IoT qui s'exécute sur le véhicule collecte et envoie des mises à jour des signaux au cloud.

Création d'un modèle d'état (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour créer un modèle d'état.

Pour créer un modèle d'état

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, choisissez State templates.
3. Sur la page Modèles d'état, choisissez Créer un modèle d'état.
4. Dans Détails du modèle d'état, entrez un nom pour le modèle d'état et entrez éventuellement une description.
5. Dans Choisir les signaux, ajoutez les signaux à partir desquels vous souhaitez obtenir des informations sur l'état du véhicule.
6. Choisissez Créer un modèle d'état.

Une fois que vous avez créé un modèle d'état avec succès, vous le verrez répertorié sur la page des modèles d'état. Vous pouvez désormais l'associer à un véhicule.

Créer un modèle d'état (AWS CLI)

Vous pouvez utiliser l'opération CreateStateTemplateAPI pour créer un modèle d'état. L'exemple suivant repose sur AWS CLI.

Pour créer un modèle d'état, exécutez la commande suivante.

*create-state-template*Remplacez-le par le nom du fichier .json qui contient la configuration du modèle d'état.

```
aws iotfleetwise create-state-template \
    --cli-input-json file://create-state-template.json
```

Example configuration du modèle d'état

`stateTemplateProperties`doit contenir les noms complets des signaux.

`dataExtraDimensions`set `metadataExtraDimensions` doit contenir les noms complets des attributs du véhicule. Les dimensions spécifiées remplacent toutes les valeurs de dimension existantes dans le modèle d'état.

```
{
    "name": "state-template-name",
    "signalCatalogArn": "arn:aws:iotfleetwise:region:account:signal-catalog/catalog-
name",
```

```
    "stateTemplateProperties": [
        "Vehicle.Signal.One",
        "Vehicle.Signal.Two"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Associer un modèle FleetWise d'état AWS IoT à un véhicule

Associer un modèle d'état à un véhicule (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour ajouter des modèles d'état associés à un véhicule.

Pour associer un modèle d'état

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Choisissez un véhicule dans la liste pour ouvrir sa page de détails.
4. Dans l'onglet Modèles d'état, choisissez Gérer les modèles d'état.
5. Choisissez Ajouter un modèle d'état.
6. Sélectionnez un modèle d'état et choisissez sa méthode de reporting.
    a. En cas de modification — Le modèle d'état signalera les modifications apportées à l'état du véhicule.
    b. Périodique — Le modèle d'état signalera les mises à jour selon l'intervalle de temps spécifié.
7. Sélectionnez Enregistrer les modifications.

Associer un modèle FleetWise d'état AWS IoT à un véhicule (AWS CLI)

Associez le modèle d'état créé à un véhicule pour permettre la collecte des mises à jour d'état du véhicule vers le cloud. Pour ce faire, utilisez :

- Lorsque vous créez un véhicule, utilisez le `stateTemplates` champ de `create-vehicle` commande. Pour de plus amples informations, veuillez consulter Créez un FleetWise véhicule AWS IoT.
- Lors de la mise à jour d'un véhicule, utilisez les `stateTemplatesToRemove` champs `stateTemplatesToAdd` ou de la `update-vehicle` commande. Pour de plus amples informations, veuillez consulter Mettre à jour un FleetWise véhicule AWS IoT.

Mettre à jour un modèle d' FleetWise état de l' AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser le fonctionnement de l'UpdateStateTemplateAPI ou la FleetWise console AWS IoT pour mettre à jour un modèle d'état existant.

Mettre à jour un modèle d'état (console)

Pour mettre à jour un modèle d'état depuis la console, rendez-vous sur la page des modèles d'état de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez le modèle d'état que vous souhaitez mettre à jour, puis choisissez Modifier.
2. Modifiez les détails du modèle d'état, puis choisissez Enregistrer les modifications.

Mettre à jour un modèle d'état (AWS CLI)

Pour mettre à jour un modèle d'état, exécutez la commande suivante.

*`update-state-template`*Remplacez-le par le nom du fichier .json qui contient la configuration du modèle d'état.

```
aws iotfleetwise update-state-template \
    --cli-input-json file://update-state-template.json
```

Example configuration du modèle d'état

Le `stateTemplateProperties` doit contenir les noms complets des signaux.

Le `dataExtraDimensions` et `metadataExtraDimensions` doit contenir les noms complets des attributs du véhicule.

```
{
    "identifier": "state-template-name",
    "stateTemplatePropertiesToAdd": [
        "Vehicle.Signal.Three"
    ],
    "stateTemplatePropertiesToRemove": [
        "Vehicle.Signal.One"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Supprimer un modèle FleetWise d'état AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l'opération DeleteStateTemplateAPI ou la FleetWise console AWS IoT pour supprimer un modèle d'état.

Supprimer un modèle d'état (console)

Pour supprimer un modèle d'état de la console, rendez-vous sur la page des modèles d'état de la FleetWise console AWS IoT et effectuez les étapes suivantes.

1. Choisissez le modèle d'état que vous souhaitez supprimer, puis choisissez Supprimer.
2. Confirmez que vous souhaitez supprimer le modèle d'état, puis choisissez Supprimer.

Supprimer un modèle d'état (AWS CLI)

Pour supprimer un modèle d'état, exécutez la commande suivante.

Remplacez *identifier* par le nom ou l'ID du modèle d'état.

```
aws iotfleetwise delete-state-template \
    --identifier idenitfier
```

Obtenir des informations sur le modèle FleetWise d'état de l' AWS IoT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez utiliser l'opération GetStateTemplateAPI pour récupérer des informations sur un modèle d'état. L'exemple suivant repose sur AWS CLI.

Remplacez *identifier* par le nom du modèle d'état.

```
aws iotfleetwise get-state-template \
    --identifier idenitfier
```

Vous pouvez utiliser l'opération ListStateTemplatesAPI pour récupérer la liste des modèles d'état que vous avez créés. L'exemple suivant repose sur AWS CLI.

```
aws iotfleetwise list-state-templates
```

Si vous avez activé le chiffrement à l'aide d'une AWS KMS clé gérée par le client, incluez la déclaration de politique suivante afin que votre rôle puisse invoquer les opérations `ListStateTemplates` d'API `GetStateTemplate` ou.

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

    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:kms:KMS_KEY_REGION:KMS_KEY_ACCOUNT_ID:key/KMS_KEY_ID"
      ]
    },
  ]
}
```

Opérations relatives aux modèles d'état pour la collecte et le traitement des données

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les sections suivantes décrivent comment utiliser les modèles d'état pour activer et désactiver la collecte de données, effectuer une opération de récupération et traiter les données d'état de vos véhicules.

Rubriques

- Activer et désactiver la collecte de données d'état à l'aide de modèles d'état
- Récupérez un instantané de l'état du véhicule à l'aide de modèles d'état
- Traitez les dernières données connues du véhicule à l'aide de la messagerie MQTT

Activer et désactiver la collecte de données d'état à l'aide de modèles d'état

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les sections suivantes décrivent comment activer et désactiver l'ingestion de données à l'aide de modèles d'état à l'aide du AWS CLI.

> **Important**
>
> Avant de commencer, assurez-vous que vous avez déjà créé un modèle d'état et que vous l'avez associé, ainsi que sa stratégie de mise à jour, à un véhicule.

Vous devez activer un modèle d'état pour que l'agent Edge puisse envoyer des mises à jour des signaux au cloud.

Pour effectuer ces opérations avec des modèles d'état, créez d'abord une ressource de commande, puis lancez l'exécution des commandes sur le véhicule. La section suivante décrit comment utiliser cette API et comment activer et désactiver l'ingestion de données.

Rubriques

- Utilisation de l'API CreateCommand
- Exemple : activation d'un modèle d'état
- Exemple : désactiver un modèle d'état

Utilisation de l'API `CreateCommand`

Créez une ressource de commande dans l'espace de noms `AWS-IoTFleetwise` « » et utilisez les paramètres suivants lorsque vous créez ou envoyez une ressource de commande pour un modèle d'état :

- `$stateTemplate.name`— Nom du modèle d'état sur lequel effectuer l'opération. Le modèle d'état doit être appliqué au véhicule pour que vous puissiez effectuer une opération. Pour de plus amples informations, veuillez consulter Associer un modèle FleetWise d'état AWS IoT à un véhicule.
- `$stateTemplate.operation`— Opération à effectuer sur le modèle d'état. Utilisez l'une des valeurs suivantes pour ce paramètre :
  - `activate`— L'agent Edge commence à envoyer des mises à jour des signaux au cloud en fonction de ce `stateTemplateUpdateStrategy` que vous avez spécifié (en cas de modification ou périodique) lorsque vous avez appliqué le modèle d'état au véhicule. Pour de plus amples informations, veuillez consulter Associer un modèle FleetWise d'état AWS IoT à un véhicule.

    Vous pouvez également définir un délai de désactivation automatique du modèle d'état pour arrêter les mises à jour après une période spécifiée. Si aucune heure de désactivation automatique n'est fournie, les modèles d'état continueront à envoyer des mises à jour jusqu'à ce qu'un appel de désactivation soit émis.

    Dès réception de la `activate` commande, l'appareil doit envoyer les signaux spécifiés dans le modèle d'état conformément à la stratégie de mise à jour. AWS L'IoT FleetWise recommande que lorsqu'une commande d'activation est reçue par l'appareil, le premier message qu'il envoie contienne un instantané de tous les signaux du modèle d'état. Les messages suivants doivent être envoyés conformément à la stratégie de mise à jour.
  - `deactivate`— L'agent Edge arrête d'envoyer des mises à jour des signaux au cloud.
  - `fetchSnapshot`— L'agent Edge envoie un instantané unique des signaux définis dans le modèle d'état, indépendamment de ceux que `stateTemplateUpdateStrategy` vous avez spécifiés lorsque vous avez appliqué le modèle d'état au véhicule.
- (Facultatif) `$stateTemplate.deactivateAfterSeconds` — Le modèle d'état est automatiquement désactivé après le délai spécifié. Ce paramètre ne peut être utilisé que lorsque la valeur du `$stateTemplate.operation` paramètre est « activer ». Si ce paramètre n'est pas spécifié, ou si la valeur de ce paramètre est 0, l'agent Edge continue d'envoyer des mises à jour des signaux au cloud jusqu'à ce qu'une opération de « désactivation » soit reçue pour le modèle d'état. Le modèle d'état n'est jamais automatiquement désactivé.

  Valeur minimale : 0, valeur maximale : 4294967295.

Note

- L'API indique le succès en réponse à une demande d'activation pour un modèle déjà actif.
- L'API indique le succès en réponse à une demande de désactivation pour un modèle déjà en état de désactivation.
- La demande la plus récente que vous faites sur un modèle d'état est celle qui prend effet. Par exemple, si vous demandez la désactivation d'un modèle d'état dans une heure, puis que vous faites une deuxième demande pour que ce même modèle soit désactivé dans quatre heures, la désactivation de quatre heures prend effet car il s'agit de la demande la plus récente.

Important

Une exception de validation peut se produire dans l'un des scénarios suivants :

- Un modèle d'état qui n'est pas fourni `ASSOCIATED` avec un véhicule est fourni.
- Une demande est faite pour activer un modèle d'état mais celui-ci ne figurait pas `DEPLOYED` sur un véhicule.
- Une demande est envoyée à un modèle d'État mais elle se trouve `DELETED` sur un véhicule.

Exemple : activation d'un modèle d'état

Pour activer un modèle d'état, créez d'abord une ressource de commande. Vous pouvez ensuite envoyer la commande suivante au véhicule sur lequel vous souhaitez activer le modèle d'état. Cet exemple montre comment vous pouvez spécifier des valeurs par défaut pour les paramètres lors de la création d'une commande. Ces paramètres et leurs valeurs sont utilisés lors du démarrage de l'exécution de la commande pour activer le modèle d'état.

1. Création d'une ressource de commande

   Avant de pouvoir envoyer une commande au véhicule, vous devez créer une ressource de commande. Vous pouvez spécifier des valeurs alternatives pour les paramètres obligatoires lorsque vous envoyez la commande au véhicule. Pour de plus amples informations, veuillez consulter Création d'une ressource de commande.

> **Important**
>
> `$stateTemplate.name`et `$stateTemplate.operation` les paramètres doivent être fournis sous forme de chaîne de données. Si un autre type de données est fourni, ou si l'un de ces deux paramètres est absent, l'exécution de la commande échoue avec une exception de validation. Le `$stateTemplate.deactivateAfterSeconds` paramètre doit être fourni sous forme de type de `Long` données.

```
aws iot create-command \
    --description "This command activates a state template on a vehicle"
    --command-id ActivateStateTemplate \
    --display-name "Activate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "activate"}
    },
    {
        "name": "$stateTemplate.deactivateAfterSeconds",
        "defaultValue": {"L": "120"}
    }
]'
```

2. Lancer l'exécution de la commande sur le véhicule

   Une fois la commande créée, envoyez-la au véhicule. Si vous n'avez pas spécifié de valeurs pour les paramètres obligatoires lors de la création de la ressource de commande, vous devez les spécifier maintenant. Pour de plus amples informations, veuillez consulter Envoyer une commande (AWS CLI).

> **Important**
>
> Assurez-vous d'utiliser le point de terminaison de l'API du plan de données des AWS IoT tâches spécifique au compte pour le fonctionnement de l'API.

```
aws iot-jobs-data start-command-execution \
    --endpoint-url <endpoint-url> \
    --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/ActivateStateTemplate \
    --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
```

3. Récupère le statut de l'opération du modèle d'état

   Après avoir lancé l'exécution de la commande, vous pouvez utiliser l'`GetCommandExecution`API pour récupérer le modèle d'état.

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

Exemple : désactiver un modèle d'état

Pour désactiver un modèle d'état, créez d'abord une ressource de commande. Vous pouvez ensuite envoyer la commande suivante au véhicule sur lequel vous souhaitez désactiver le modèle d'état. Cet exemple montre comment vous pouvez spécifier des valeurs par défaut pour les paramètres lors de la création d'une commande. Ces paramètres et leurs valeurs sont utilisés lors du démarrage de l'exécution de la commande pour désactiver le modèle d'état.

1. Création d'une ressource de commande

   Avant de pouvoir envoyer une commande au véhicule, vous devez créer une ressource de commande. Vous pouvez spécifier des valeurs alternatives pour les paramètres obligatoires lorsque vous envoyez la commande au véhicule. Pour de plus amples informations, veuillez consulter Création d'une ressource de commande.

```
aws iot create-command \
    --description "This command deactivates a state template on a vehicle"
    --command-id DeactivateStateTemplate \
    --display-name "Deactivate State Template" \
    --namespace AWS-IoTFleetWise \
```

```
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "deactivate"}
    }
]'
```

2. Lancer l'exécution de la commande sur le véhicule

   Une fois la commande créée, envoyez-la au véhicule. Si vous n'avez pas spécifié de valeurs pour les paramètres obligatoires lors de la création de la ressource de commande, vous devez les spécifier maintenant. Pour de plus amples informations, veuillez consulter Envoyer une commande (AWS CLI).

   ```
   aws iot-jobs-data start-command-execution \
       --endpoint-url <endpoint-url> \
       --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/DeactivateStateTemplate
    \
       --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
   ```

3. Récupère le statut de l'opération du modèle d'état

   Après avoir lancé l'exécution de la commande, vous pouvez utiliser l'`GetCommandExecution`API pour récupérer le modèle d'état.

   ```
   aws iot get-command-execution  --execution-id <EXECUTION_ID>
   ```

Récupérez un instantané de l'état du véhicule à l'aide de modèles d'état

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Vous pouvez récupérer le dernier état connu d'un véhicule à l'aide de l'API du plan de `CreateCommand` AWS IoT Core contrôle ou de la FleetWise console AWS IoT.

> **Important**
>
> Une exception de validation peut se produire dans l'un des scénarios suivants :
>
> - Un modèle d'état qui n'est pas fourni `ASSOCIATED` avec un véhicule est fourni.
> - Une demande est faite pour activer un modèle d'état mais celui-ci ne figurait pas `DEPLOYED` sur un véhicule.
> - Une demande est envoyée à un modèle d'État mais elle se trouve `DELETED` sur un véhicule.

Récupérer un instantané de l'état du véhicule (console)

Vous pouvez utiliser la FleetWise console AWS IoT pour récupérer le dernier état connu d'un véhicule. AWS L'IoT FleetWise créera une commande vous permettant de récupérer des données.

Pour récupérer l'état d'un véhicule

1. Ouvrez la FleetWise console AWS IoT.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Choisissez un véhicule dans la liste pour ouvrir sa page de détails.
4. Dans l'onglet Modèles d'état, choisissez Fetch data.
5. Sélectionnez le rôle IAM qui autorise AWS l'IoT à FleetWise envoyer une commande et à récupérer des données. Consultez la section Contrôle de l'accès.
6. Choisissez Fetch state.

Récupère un instantané de l'état du véhicule ()AWS CLI

Pour récupérer un instantané d'état, créez d'abord une ressource de commande. Vous pouvez ensuite envoyer la commande suivante au véhicule dont vous souhaitez récupérer le cliché d'état. Pour plus d'informations sur l'utilisation de l'`CreateCommand`API et de ses paramètres, consultezUtilisation de l'API `CreateCommand`.

1. Création d'une ressource de commande

   L'exemple suivant montre comment créer la ressource de commande pour effectuer l'opération d'extraction. Vous pouvez spécifier des valeurs alternatives pour les paramètres obligatoires lorsque vous envoyez la commande au véhicule. Pour de plus amples informations, veuillez consulter Création d'une ressource de commande.

   ```
   aws iot create-command \
       --command-id <COMMAND_ID> \
       --display-name "FetchSnapshot State Template" \
       --namespace AWS-IoTFleetWise \
       --mandatory-parameters '[
         {
             "name": "$stateTemplate.name",
             "defaultValue": {"S": "ST123"}
         },
         {
             "name": "$stateTemplate.operation",
             "defaultValue": {"S": "fetchSnapshot"}
         }
       ]'
   ```

   Réponse :

   ```
   {
       "commandId": "<COMMAND_ID>",
       "commandArn": "arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID>"
   }
   ```

2. Lancer l'exécution de la commande pour récupérer un instantané d'état

   Une fois la commande créée, envoyez-la au véhicule. Si vous n'avez pas spécifié de valeurs pour les paramètres obligatoires lors de la création de la ressource de commande, vous devez les spécifier maintenant. Pour de plus amples informations, veuillez consulter Envoyer une commande (AWS CLI).

   ```
   aws iot-jobs-data start-command-execution \
       --command-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:command/<COMMAND_ID> \
       --target-arn arn:aws:iot:<REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>
   ```

   Réponse :

```
{
    "executionId": "<UNIQUE_UUID>"
}
```

3. Récupère le statut de l'opération du modèle d'état

   Après avoir lancé l'exécution de la commande, vous pouvez utiliser l'`GetCommandExecution`API pour récupérer le modèle d'état.

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

Traitez les dernières données connues du véhicule à l'aide de la messagerie MQTT

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Pour recevoir des mises à jour de votre véhicule et traiter ses données, abonnez-vous à la rubrique MQTT suivante. Pour plus d'informations, consultez les rubriques relatives au MQTT dans le Guide du AWS IoT Core développeur.

```
$aws/iotfleetwise/vehicles/$vehicle_name/last_known_state/$state_template_name/data
```

Les derniers messages connus de mise à jour du signal d'état peuvent être reçus hors service, car MQTT ne garantit pas la commande. Tous les clients qui utilisent MQTT pour recevoir et traiter les données du véhicule doivent s'en occuper. Les derniers messages de mise à jour du signal d'état connus suivent le protocole de messagerie MQTT 5.

L'en-tête de message de chaque message MQTT possède les propriétés utilisateur suivantes :

- VehicleName — Identifiant unique des véhicules.
- stateTemplateName— Identifiant unique du dernier modèle d'état connu.

En outre, vous pouvez spécifier les attributs du véhicule à inclure dans l'en-tête du message MQTT en spécifiant le paramètre de `metadataExtraDimensions` demande lors de la mise à jour ou de la création d'un modèle d'état. (Voir Modèles d'états.)

Les propriétés utilisateur de l'en-tête du message MQTT sont utiles pour acheminer des messages vers différentes destinations sans inspecter la charge utile.

La charge utile des messages MQTT contient des données collectées auprès des véhicules. Vous pouvez spécifier les attributs du véhicule à inclure dans la charge utile du message MQTT en spécifiant le paramètre de `extraDimensions` demande lors de la création ou de la mise à jour d'un modèle d'état (voirCréation d'un modèle d' FleetWise état de l' AWS IoT). Les dimensions supplémentaires enrichissent les données collectées auprès des véhicules en leur associant des dimensions supplémentaires.

La charge utile du message MQTT est codée en mémoire tampon (Protobuf), et l'en-tête du message MQTT contient un indicateur de type de contenu défini comme application/octet-stream. Le schéma de codage Protobuf est le suivant :

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

syntax = "proto3";

option java_package = "com.amazonaws.iot.autobahn.schemas.lastknownstate";
package Aws.IoTFleetWise.Schemas.CustomerMessage;

message LastKnownState {

  /*
   * The absolute timestamp in milliseconds since Unix Epoch of when the event was
 triggered in vehicle.
   */
  uint64 time_ms = 1;

  /*
   * This field is deprecated, use signals instead
   */
  repeated Signal signal = 2 [ deprecated = true ];

  repeated Signal signals = 3;

  repeated ExtraDimension extra_dimensions = 4;
```

```
}

message Signal {

  /*
   * The Fully Qualified Name of the signal is the path to the signal plus the signal's
 name.
   * For example, Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  /*
   * The FWE reported signal value can be one of the following data types.
   */
  oneof SignalValue {
    double double_value = 2;

    bool boolean_value = 3;

    sint32 int8_value = 4;

    uint32 uint8_value = 5;

    sint32 int16_value = 6;

    uint32 uint16_value = 7;

    sint32 int32_value = 8;

    uint32 uint32_value = 9;

    sint64 int64_value = 10;

    uint64 uint64_value = 11;

    float float_value = 12;
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 13;
  }
}
```

```
message ExtraDimension {
  /*
   * The Fully Qualified Name of the attribute is the path to the attribute plus the
 attribute's name.
   * For example, Vehicle.Model.Color
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  oneof ExtraDimensionValue {
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 2;
  }
}
```

Où :

- `time_ms`:

  Horodatage absolu (en millisecondes depuis l'époque Unix) du moment où l'événement a été déclenché dans le véhicule. Le logiciel Edge Agent utilise l'horloge du véhicule pour cet horodatage.
- `signal`:

  Un tableau de `Signal` s contenant les informations du signal : `name` (chaîne) et `signalValue` qui prend en charge les types de données suivants : `double` `boolint8`,`uint8`,,`int16`,`uint16`,`int32`,`uint32`,`int64`,`uint64`,`float`,`string`.
- `extra_dimensions`:

  Un tableau `ExtraDimensions` contenant des informations sur les attributs du véhicule : `name` (chaîne) et `extraDimensionValue` qui ne prend actuellement en charge que le type de `string` données.

Tutoriel : Configuration de la collecte de données indépendante du réseau à l'aide d'une interface de décodage personnalisée

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Introduction

Ce didacticiel explique comment configurer l' AWS IoT FleetWise pour collecter des données et exécuter des commandes à l'aide d'une collecte de données indépendante du réseau, qui utilise une interface de décodage personnalisée. Grâce à la collecte de données indépendante du réseau, vous pouvez utiliser vos propres méthodes pour décoder les signaux avant de les envoyer à la destination de données que vous avez spécifiée. Cela permet de gagner du temps, car il n'est pas nécessaire de créer des décodeurs de signaux spécifiques pour l'IoT AWS . FleetWise Vous pouvez décoder un sous-ensemble de signaux à l'aide de votre propre implémentation, ou vous pouvez l'utiliser `defaultForUnmappedSignals` lorsque vous créez ou mettez à jour un manifeste de décodeur. Cela permet également de collecter des signaux et des déclencheurs via un large éventail de sources au sein du véhicule.

Ce didacticiel est destiné aux signaux de véhicules qui ne se trouvent pas sur une interface de réseau de contrôle (bus CAN) standard. Par exemple, les données sont codées dans un format ou un schéma personnalisé intégré au véhicule.

Configuration de l’environnement

Ce didacticiel part du principe que vous avez suivi les étapes de configuration de vos environnements pour accéder au FleetWise cloud AWS IoT, ainsi qu'à l'implémentation APIs et à la base de code d'Edge.

Modèles de données

La section suivante montre comment modéliser les propriétés d'un véhicule à l'aide d'une interface de décodage personnalisée. Cela s'applique à la collecte de données ainsi qu'aux cas d'utilisation des commandes. Elle s'applique également à toute modélisation de source de données sous-jacente utilisée dans le véhicule, par exemple IDLs.

Dans l'exemple, il existe deux propriétés du véhicule : un capteur du véhicule (position actuelle du véhicule) à collecter et un actionneur du véhicule (climatiseur) à contrôler à distance. Les deux sont définis dans ce schéma :

```
// Vehicle WGS84 Coordinates
double Latitude;
double Longitude;

// Vehicle AC
Boolean ActivateAC;
```

L'étape suivante consiste à importer ces définitions dans FleetWise l' AWS IoT à l'aide de l'interface APIs de décodage personnalisée.

mises à jour du catalogue de signaux

Importez ces définitions dans votre catalogue de signaux. Si vous disposez FleetWise déjà d'un catalogue de signaux dans AWS l'IoT, utilisez directement l'API de mise à jour. Si vous n'en avez pas, créez d'abord un catalogue de signaux, puis appelez l'API de mise à jour.

Vous devez d'abord créer la représentation VSS de ces signaux du véhicule. Le VSS est utilisé comme taxonomie pour représenter les données des véhicules dans l'IoT AWS . FleetWise Créez un fichier json appelé « vehicle-signals.json » avec le contenu suivant :

```
// vehicle-signals.json
// Verify that branches and nodes are unique in terms of fully qualified name
// in the signal catalog.
[
 {
    "branch": {
      "fullyQualifiedName": "Vehicle",
      "description": "Vehicle Branch"
    }
  },
```

```
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.CurrentLocation",
      "description": "CurrentLocation"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "description": "Latitude"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "description": "Longitude"
    }
  },
  {
    "actuator": {
      "fullyQualifiedName": "Vehicle.ActivateAC",
      "description": "AC Controller",
      "dataType": "BOOLEAN"
    }
  }
]
```

Si vous n'avez pas de catalogue de signaux en place, vous devez invoquer `create-signal-catalog` :

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise create-signal-catalog \
        --name my-signal-catalog \
        --nodes "${VEHICLE_NODES}"
```

Si vous disposez déjà d'un catalogue de signaux, vous pouvez ajouter ces signaux à l'aide de l'`update-signal-catalog`API :

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise update-signal-catalog \
```

```
        --name my-signal-catalog \
        --nodes-to-add "${VEHICLE_NODES}"
```

Modèle de véhicule et décodeur

Après avoir inséré les signaux dans le catalogue de signaux, l'étape suivante consiste à créer un modèle de véhicule et à instancier ces signaux. Pour cela, vous utilisez le `create-model-manifest` et `create-decoder-manifest` APIs.

Commencez par formater les noms des signaux que vous souhaitez insérer dans le modèle du véhicule :

```
# Prepare the signals for insertion into the vehicle model.
VEHICLE_NODES=`cat vehicle-signals.json`
VEHICLE_NODES=`echo ${VEHICLE_NODES} | jq -r ".[] | .actuator,.sensor
 | .fullyQualifiedName" | grep Vehicle\\.`
VEHICLE_NODES=`echo "${VEHICLE_NODES}" | jq -Rn [inputs]`
# This is how the vehicle model input looks.
echo $VEHICLE_NODES
# [ "Vehicle.CurrentLocation.Latitude",
#   "Vehicle.CurrentLocation.Longitude",
#   "Vehicle.ActivateAC" ]
# Create the vehicle model with those signals.
aws iotfleetwise create-model-manifest \
    --name my-model-manifest \
    --signal-catalog-arn arn:xxxx:signal-catalog/my-signal-catalog \
    --nodes "${VEHICLE_NODES}"

# Activate the vehicle model.
 aws iotfleetwise update-model-manifest \
    --name my-model-manifest --status ACTIVE
```

Utilisez maintenant l'interface de décodage personnalisée pour créer un manifeste de décodeur.

> **Note**
>
> Vous n'avez besoin de créer des interfaces réseau et des signaux que si vous souhaitez spécifier une option personnalisée IDs, ce qui ne fait pas partie de cet exemple.

Pour plus d'informations sur le mappage des informations de décodage lorsque le nom complet (FQN) diffère de l'ID du signal de décodage personnalisé, consultez le Guide du développeur de l'agent Edge.

```
// Create a network interface that is of type : CUSTOM_DECODING_INTERFACE
// custom-interface.json
[
  {
    "interfaceId": "NAMED_SIGNAL",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  },
  {
    "interfaceId": "AC_ACTUATORS",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
// custom-decoders.json
// Refer to the fully qualified names of the signals, make them of
// type CUSTOM_DECODING_SIGNAL, and specify them as part of the same interface ID
// that was defined above.
[
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Longitude"
      }
    },
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Latitude"
```

```
      }
    },
    {
        "fullyQualifiedName": "Vehicle.ActivateAC",
        "interfaceId": "AC_ACTUATORS",
        "type": "CUSTOM_DECODING_SIGNAL",
        "customDecodingSignal": {
          "id": "Vehicle.ActivateAC"
        }
    }
]
# Create the decoder manifest.
 CUSTOM_INTERFACE=`cat custom-interface.json`
 CUSTOM_DECODERS=`cat custom-decoders.json`

aws iotfleetwise create-decoder-manifest \
   --name my-decoder-manifest \
   --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
   --network-interfaces "${CUSTOM_INTERFACE}" \
   --signal-decoders "${CUSTOM_DECODERS}"

 # Activate the decoder manifest.
 aws iotfleetwise update-decoder-manifest \
    --name my-decoder-manifest \
    --status ACTIVE
```

À ce stade, vous avez entièrement modélisé ces signaux dans AWS l'IoT FleetWise. Ensuite, vous créez le véhicule et vous l'associez au modèle que vous avez créé. Vous utilisez l'`create-vehicle`API pour cela :

```
aws iotfleetwise create-vehicle \
    --decoder-manifest-arn arn:xxx:decoder-manifest/my-decoder-manifest \
    --association-behavior ValidateIotThingExists \
    --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
    --vehicle-name "my-vehicle"
```

L'étape suivante consiste à se concentrer sur la base de code AWS IoT FleetWise Edge et à écrire l'extension de code nécessaire.

> **Note**
>
> Pour plus d'informations sur l'implémentation d'Edge, consultez le Guide du développeur de l'agent Edge.

Envoyer une commande

Maintenant, compilez le logiciel (assurez-vous d'ajouter vos en-têtes et vos fichiers C++ au CMake fichier), puis retournez dans le cloud APIs pour tester une commande sur cet actionneur :

```
// Create a command targeting your vehicle.
aws iot create-command --command-id activateAC \
    --namespace "AWS-IoT-Fleetwise" \
    --endpoint-url endpoint-url \
    --role-arn ${SERVICE_ROLE_ARN} \
    --mandatory-parameters '[ { "name": "$actuatorPath.Vehicle.ActivateAC",
 "defaultValue": {"B": "false"} } ]' \
// You will receive the command ARN.

{
    "commandId": "activateAC",
    "commandArn": "arn:aws:iot:xxx:command/activateAC"
}

// You can send the command to activate the AC targeting your vehicle.

JOBS_ENDPOINT_URL=`aws iot describe-endpoint --endpoint-type iot:Jobs | jq -
j .endpointAddress`
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:xxx:command/activateAC \
    --target-arn arn:xxx:vehicle/my-vehicle \
    --parameters '{ "$actuatorPath.Vehicle.ActivateAC" : {"B": "true"}}' \
    --endpoint-url https://${JOBS_ENDPOINT_URL}
// You will receive the corresponding execution ID.
{
    "executionId": "01HSK4ZH6ME7D43RB2BV8JC51D"
}

// If you have the AWS IoT FleetWise Edge Agent running, you can see the logs.
[AcCommandDispatcher.cpp:26] [setActuatorValue()]:
```

```
[Actuator Vehicle.ActivateAC executed successfully for command ID
 01HSK4ZH6ME7D43RB2BV8JC51D]
```

Utilisation AWS CLI et AWS SDKs avec AWS l'IoT FleetWise

Cette section fournit des informations sur les requêtes FleetWise d'API AWS IoT. Pour plus d'informations sur les AWS FleetWise opérations et les types de données liés à l'AWS IoT, consultez la référence des FleetWise API IoT.

Pour utiliser AWS l'IoT FleetWise avec différents langages de programmation, utilisez le AWS SDKs, qui contient les fonctionnalités automatiques suivantes :

- Signature cryptographique des requêtes de service
- Nouvelles tentatives de requête
- Gestion des réponses d'erreur

Pour accéder à la ligne de commande, utilisez AWS l'IoT FleetWise avec le AWS CLI. Vous pouvez contrôler AWS l'IoT FleetWise et vos autres services depuis la ligne de commande et les automatiser par le biais de scripts.

Résolution des problèmes liés à AWS l'IoT FleetWise

Utilisez les informations de dépannage et les solutions de cette section pour résoudre les problèmes liés à AWS l'IoT FleetWise.

Les informations suivantes peuvent vous aider à résoudre les problèmes courants liés à AWS l'IoT FleetWise.

Rubriques

- AWS Problèmes manifestes du FleetWise décodeur IoT
- Problèmes liés au FleetWise logiciel Edge Agent pour AWS l'IoT
- Stocker et transférer les numéros

AWS Problèmes manifestes du FleetWise décodeur IoT

Résolvez les problèmes liés au manifeste du décodeur.

Diagnostic des appels d'API du manifeste du décodeur

Erreur	Consignes pour la résolution des problèmes
`UpdateOperationFailure.ConflictingDecoderUpdate`	Le même manifeste de décodeur contient plusieurs demandes de mise à jour. Attendez et réessayez.
`UpdateOperationFailure.InternalFailure`	InternalFailure est lancé en tant qu'exception encapsulée. Le problème lui-même dépend de l'exception encapsulée.
`UpdateOperationFailure.ActiveDecoderUpdate`	Le manifeste du décodeur est dans un `Active` état et ne peut pas être mis à jour. Modifiez l'état du manifeste du décodeur sur`DRAFT`, puis réessayez.
`UpdateOperationFailure.ConflictingModelUpdate`	AWS L'IoT FleetWise essaie de valider par rapport à un modèle de véhicule (manifest

Erreur	Consignes pour la résolution des problèmes
	e du modèle) modifié par quelqu'un d'autre. Attendez et réessayez.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND`	Aucun signal n'est associé au modèle du véhicule. Ajoutez des signaux au modèle de véhicule et vérifiez qu'ils se trouvent dans le catalogue de signaux associé.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE`	Mettez à jour le modèle du véhicule pour qu'il soit en bon `ACTIVE` état, puis réessayez.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_FOUND`	AWS L'IoT FleetWise ne trouve pas le modèle de véhicule associé au manifeste du décodeur. Vérifiez l'Amazon Resource Name (ARN) du modèle de véhicule et réessayez.
`UpdateOperationFailure.ModelManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE`	La validation du modèle de véhicule a échoué car les noms des signaux du modèle de véhicule n'ont pas été trouvés dans le catalogue de signaux. Vérifiez que les signaux du modèle de véhicule sont tous inclus dans le catalogue de signaux associé.
`UpdateOperationFailure.ValidationFailure`	Des signaux ou des interfaces réseau non valides ont été trouvés dans la demande de mise à jour du manifeste du décodeur. Vérifiez que tous les signaux et interfaces réseau renvoyés par l'exception existent, que tous les signaux utilisés sont associés à une interface disponible et que vous ne supprimerez pas une interface associée à des signaux.

Erreur	Consignes pour la résolution des problèmes
`UpdateOperationFailure.KmsKeyAccessDenied`	Il y a un problème d'autorisation sur la clé AWS Key Management Service (AWS KMS) utilisée pour l'opération. Vérifiez que vous utilisez un rôle ayant accès à la clé et réessayez.
`UpdateOperationFailure.DecoderDoesNotExist`	Le manifeste du décodeur n'existe pas. Vérifiez le nom du manifeste du décodeur et réessayez.

Les messages d'erreur relatifs aux données du système de vision indiquant le `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` motif incluront dans la réponse un indice expliquant pourquoi la demande a échoué. Vous pouvez utiliser cet indice pour déterminer les consignes de dépannage à suivre.

> **Note**
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Diagnostic, décodeur, manifeste, vision, validation des données du système

Erreur	Consignes pour la résolution des problèmes
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)`	AWS L'IoT FleetWise n'a pas trouvé la structure de signal racine utilisée dans le décodeur de signaux à l'aide du catalogue de signaux. Vérifiez que le signal racine de la structure est correctement défini dans le catalogue de signaux.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	Aucun message primitif dans le catalogue de signaux n'a été défini avec le même type de données dans la demande de mise à jour du manifeste du décodeur. Vérifiez que les messages primitifs définis dans la demande

Erreur	Consignes pour la résolution des problèmes
	correspondent à leur définition de catalogue de signaux correspondante.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)`	Le nombre de propriétés définies dans une structure du catalogue de signaux ne correspond pas au nombre de propriétés que vous essayez de décoder dans le manifeste du décodeur. Vérifiez que vous disposez du nombre correct de signaux à décoder en le comparant aux signaux définis dans le catalogue de signaux.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	AWS L'IoT FleetWise a trouvé un signal défini comme une STRUCTURE dans le catalogue de signaux sans qu'il soit structuredMessageDefinition défini dans la demande de manifeste du décodeur. Assurez-vous que chaque structure est définie comme une structure dMessageDefinition dans la demande de mise à jour du manifeste du décodeur.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	Le signal racine de la structure utilisée dans le manifeste du décodeur n'est pas correctement défini en tant que structure dans le catalogue de signaux. La structure du signal racine utilisée dans le manifeste du décodeur doit avoir son champ structFullyQualified Nom défini. Il a également besoin d'un nœud STRUCT avec cela fullyQualifiedName.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	L'un des messages leaf utilisés dans la demande de manifeste du décodeur n'est pas défini comme un message primitif. Vérifiez que tous les objets feuille de la demande sont définis comme des messages primitifs.

Erreur	Consignes pour la résolution des problèmes
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	Un objet de tableau dans le catalogue de signaux n'a pas été défini en tant que structure dMessageList définition dans la demande de mise à jour du manifeste du décodeur. Vérifiez que toutes les propriétés du tableau sont définies en tant que structuredMessageList définition dans la demande de mise à jour du manifeste du décodeur.

Problèmes liés au FleetWise logiciel Edge Agent pour AWS l'IoT

Résolvez les problèmes liés au logiciel Edge Agent.

Problèmes

- Problème : le logiciel Edge Agent ne démarre pas.
- Problème : [ERREUR] [Io TFleet WiseEngine : :connect] : [Impossible d'initialiser la bibliothèque de persistance]
- Problème : Le logiciel Edge Agent ne collecte pas les diagnostics embarqués (OBD) II PIDs et les codes de diagnostic (DTCs).
- Problème : Le FleetWise logiciel Edge Agent for AWS IoT ne collecte pas de données sur le réseau ou n'est pas en mesure d'appliquer les règles d'inspection des données.
- Problème : [ERROR] [AwsIotConnectivityModule: :connect] : [Échec de la connexion avec erreur] ou [WARN] [AwsIotChannel: :send] : [Aucune connexion MQTT active.]

Problème : le logiciel Edge Agent ne démarre pas.

Les erreurs suivantes peuvent s'afficher lorsque le logiciel Edge Agent ne démarre pas.

- ```
  Error from reader: * Line 1, Column 1
  Syntax error: value, object or array expected.
  ```

  Solution : assurez-vous que le fichier de configuration du FleetWise logiciel Edge Agent for AWS IoT utilise un format JSON valide. Par exemple, assurez-vous que les virgules sont utilisées
  ```

correctement. Pour plus d'informations sur le fichier de configuration, procédez comme suit pour télécharger le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT.

1. Ouvrez la FleetWise console AWS IoT.
2. Sur la page d'accueil du service, dans la FleetWise section Commencer avec AWS l'IoT, choisissez Explore Edge Agent.

- ```
  [ERROR] [SocketCANBusChannel::connect]: [ SocketCan with name xxx is not accessible]
  [ERROR] [IoTFleetWiseEngine::connect]: [ Failed to Bind Consumers to Producers ]
  ```

  Solution : cette erreur peut s'afficher lorsque le logiciel Edge Agent ne parvient pas à établir une communication par socket avec les interfaces réseau définies dans le fichier de configuration.

  Pour vérifier que chaque interface réseau définie dans la configuration est disponible, exécutez la commande suivante.

  ```
  ip link show
  ```

  Pour mettre en ligne une interface réseau, exécutez la commande suivante. Remplacez *network-interface-id* par l'ID de l'interface réseau.

  ```
  sudo ip link set network-interface-id up
  ```

- ```
  [ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error]
  [WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]
  # or
  [WARN] [AwsIotChannel::send]: [aws-c-common: AWS_ERROR_FILE_INVALID_PATH]
  ```

  Solution : cette erreur peut s'afficher lorsque le logiciel Edge Agent ne parvient pas à établir une connexion MQTT avec. AWS IoT Core Vérifiez que les éléments suivants sont correctement configurés et redémarrez le logiciel Edge Agent.

  - `mqttConnection::endpointUrl`— point de terminaison de l'appareil IoT du AWS compte.
  - `mqttConnection::clientID`— L'ID du véhicule dans lequel le logiciel Edge Agent est exécuté.
  - `mqttConnection::certificateFilename`— Le chemin d'accès au fichier de certificat du véhicule.

- `mqttConnection::privateKeyFilename`— Le chemin d'accès au fichier de clé privée du véhicule.
- Vous avez l'habitude AWS IoT Core de ravitailler le véhicule. Pour de plus amples informations, veuillez consulter Fournir AWS des FleetWise véhicules IoT.

Pour plus d'informations sur le dépannage, consultez Kit SDK des appareils AWS IoT pour C++ les questions fréquemment posées.

Problème : [ERREUR] [Io TFleet WiseEngine : :connect] : [Impossible d'initialiser la bibliothèque de persistance]

Solution : cette erreur peut s'afficher lorsque le logiciel Edge Agent ne parvient pas à localiser le stockage persistant. Vérifiez que les éléments suivants sont correctement configurés et redémarrez le logiciel Edge Agent.

`persistency:persistencyPath`— Un chemin local utilisé pour conserver les schémas de collecte, les manifestes du décodeur et les instantanés de données.

Problème : Le logiciel Edge Agent ne collecte pas les diagnostics embarqués (OBD) II PIDs et les codes de diagnostic (DTCs).

Solution : cette erreur peut s'afficher si `obdInterface:pidRequestIntervalSeconds` ou `obdInterface:dtcRequestIntervalSeconds` est configuré sur 0.

Si le logiciel Edge Agent est exécuté dans un véhicule à transmission automatique, assurez-vous qu'`obdInterface:hasTransmissionEcu`il est configuré pour`true`.

Si votre véhicule prend en charge l'arbitrage étendu via le Controller Area Network (bus CAN) IDs, assurez-vous qu'`obdInterface:useExtendedIds`il est configuré pour`true`.

Problème : Le FleetWise logiciel Edge Agent for AWS IoT ne collecte pas de données sur le réseau ou n'est pas en mesure d'appliquer les règles d'inspection des données.

Solution : cette erreur peut s'afficher lorsque les quotas par défaut sont dépassés.

Ressource	Quota	Ajustable	Remarque
Valeur de l'ID du signal	L'ID du signal doit être inférieur ou égal à 50 000	Oui	Le logiciel Edge Agent ne collecte pas de données à partir de signaux dont l'identif iant est supérieur à 50 000. Nous vous recommandons de vérifier le nombre de signaux contenus dans le catalogue de signaux avant de modifier ce quota.
Nombre de systèmes de collecte de données actifs par véhicule	256	Oui	Nous vous recommandons de vérifier le nombre de campagnes que vous avez créées dans le cloud et le nombre de schémas que contient chaque campagne avant de modifier ce quota.
Taille de la mémoire tampon de l'historique du signal	20 Mo	Oui	Si le quota est dépassé, le logiciel Edge Agent arrête de collecter de nouvelles données.

Problème : [ERROR] [AwsIotConnectivityModule: :connect] : [Échec de la connexion avec erreur] ou [WARN] [AwsIotChannel: :send] : [Aucune connexion MQTT active.]

Solution : cette erreur peut s'afficher lorsque le logiciel Edge Agent n'est pas connecté au cloud. Par défaut, le logiciel Edge Agent envoie une demande ping AWS IoT Core toutes les minutes et attend trois minutes. En l'absence de réponse, le logiciel Edge Agent rétablit automatiquement la connexion au cloud.

Stocker et transférer les numéros

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Problème : réception d'un message **`AccessDeniedException`** avec toutes les autorisations IAM requises

Solution : le lancement anticipé de la fonctionnalité Store and Forward pour le partitionnement des données dans les campagnes nécessite Allowlisting. Contactez l'équipe de service pour vous assurer que vos ressources disposent des autorisations appropriées via allowlisting.

Problème : Les données téléchargées vers AWS IoT Jobs ne tiennent pas compte des **`endTime`**

Solution : Vous avez indiqué une valeur non valide `endtime` dans le document de travail. Par exemple, le `endtime` ne suit pas le format ISO 8601 (UTC). Dans les journaux de AWS IoT FleetWise l'agent, il peut y avoir une déclaration de niveau d'avertissement indiquant :. `Malformed IoT Job endTime: customer configured endTime. Not setting endTime`

Problème : le téléchargement des données vers AWS IoT Jobs a un statut **`REJECTED`** d'exécution.

Solution : Vous avez indiqué une valeur non valide `campaignArn` dans le document de travail. Par exemple, si vous spécifiez un ARN pour une campagne qui n'est pas exécutée sur un véhicule, il se peut qu'une déclaration indiquant le niveau d'erreur indique «, `CampaignArn value in the received job document does not match the ARN of a Store and Forward campaign` dans les journaux de l' AWS IoT FleetWise agent.

La sécurité dans AWS l'IoT FleetWise

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud : AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité applicables à AWS l'IoT FleetWise, consultez la section Services AWS concernés par programme de conformité .
- Sécurité dans le cloud — Votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris la sensibilité de vos données, les exigences de votre entreprise et la législation et la réglementation applicables.

Cette documentation vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de l' AWS IoT FleetWise. Il vous montre comment configurer l' AWS IoT FleetWise pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos FleetWise ressources AWS IoT.

Table des matières

- Protection des données dans AWS l'IoT FleetWise
- Contrôler l'accès avec AWS IoT FleetWise
- Identity and Access Management pour AWS l'IoT FleetWise
- Validation de conformité pour AWS l'IoT FleetWise
- Résilience dans AWS l'IoT FleetWise
- La sécurité des infrastructures dans AWS l'IoT FleetWise
- Analyse de configuration et de vulnérabilité dans AWS l'IoT FleetWise
- Bonnes pratiques de sécurité pour AWS l'IoT FleetWise

Protection des données dans AWS l'IoT FleetWise

Le modèle de responsabilité AWS partagée s'applique à la protection des données dans AWS l'IoT FleetWise. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez Questions fréquentes (FAQ) sur la confidentialité des données. Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog Modèle de responsabilité partagée AWS et RGPD (Règlement général sur la protection des données) sur le Blog de sécuritéAWS .

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) avec chaque compte.
- SSL/TLS À utiliser pour communiquer avec AWS les ressources. Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail. Pour plus d'informations sur l'utilisation des CloudTrail sentiers pour capturer AWS des activités, consultez la section Utilisation des CloudTrail sentiers dans le guide de AWS CloudTrail l'utilisateur.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de modules cryptographiques validés par la norme FIPS 140-3 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS disponibles, consultez Norme FIPS (Federal Information Processing Standard) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Nom. Cela inclut lorsque vous travaillez avec AWS l'IoT FleetWise ou autre Services AWS

à l'aide de la console, de l'API ou AWS SDKs. AWS CLI Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

AWS L'IoT FleetWise est destiné à être utilisé avec un agent Edge que vous développez et installez sur le matériel automobile compatible afin de transmettre les données du véhicule au AWS cloud. L'extraction de données à partir de véhicules peut être soumise aux réglementations relatives à la confidentialité des données dans certaines juridictions. Avant d'utiliser l' AWS IoT FleetWise et d'installer votre agent Edge, nous vous recommandons vivement d'évaluer vos obligations de conformité conformément à la législation en vigueur. Cela inclut toutes les exigences légales applicables visant à fournir des avis de confidentialité juridiquement adéquats et à obtenir les consentements nécessaires pour extraire les données du véhicule.

Le chiffrement au repos dans AWS l'IoT FleetWise

Les données collectées à partir d'un véhicule sont transmises au cloud par le biais d'un AWS IoT Core message utilisant le protocole de message MQTT. AWS L'IoT FleetWise fournit les données à votre base de données Amazon Timestream. Dans Timestream, vos données sont cryptées. Toutes Services AWS cryptent les données au repos par défaut. Pour plus d'informations, consultez Protection des données par chiffrement dans le guide de l'utilisateur Amazon S3 et Protection des données dans Timestream pour. LiveAnalytics

Encryption at rest s'intègre à AWS Key Management Service (AWS KMS) pour gérer la clé de chiffrement utilisée pour chiffrer vos données. Vous pouvez choisir d'utiliser une clé gérée par le client pour chiffrer les données collectées par AWS l'IoT FleetWise. Vous pouvez créer, gérer et consulter votre clé de chiffrement via AWS KMS. Pour plus d'informations, voir Qu'est-ce que c'est AWS Key Management Service ? dans le Guide AWS Key Management Service du développeur.

Chiffrement en transit

Toutes les données échangées avec les AWS IoT services sont cryptées en transit à l'aide du protocole TLS (Transport Layer Security). Pour de plus amples informations, veuillez consulter Sécurité du transport dans le Manuel du développeur AWS IoT .

Prend également en AWS IoT Core charge l'authentification et l'autorisation pour contrôler en toute sécurité l'accès aux FleetWise ressources de AWS l'IoT. Les véhicules peuvent utiliser les certificats X.509 pour s'authentifier (se connecter) afin d'utiliser l' AWS IoT FleetWise et utiliser des AWS

IoT Core politiques pour obtenir l'autorisation (disposer d'autorisations) pour effectuer des actions spécifiques. Pour de plus amples informations, veuillez consulter the section called “Véhicules de ravitaillement”.

Chiffrement des données dans AWS l'IoT FleetWise

Le chiffrement des données fait référence à la protection des données en transit (lorsqu'elles sont acheminées vers et depuis AWS l'IoT FleetWise, et entre les passerelles et les serveurs) et au repos (lorsqu'elles sont stockées sur des appareils locaux ou à Services AWS l'intérieur). Vous pouvez protéger les données au repos à l'aide du chiffrement côté client.

> **Note**
>
> AWS Le traitement de FleetWise pointe de l'IoT expose APIs des objets hébergés sur des FleetWise passerelles AWS IoT et accessibles via le réseau local. Ils APIs sont exposés via une connexion TLS soutenue par un certificat de serveur appartenant au connecteur AWS IoT FleetWise Edge. Pour l'authentification du client, ceux-ci APIs utilisent un mot de passe de contrôle d'accès. La clé privée du certificat de serveur et le mot de passe de contrôle d'accès sont tous deux stockés sur disque. AWS Le traitement de FleetWise pointe de l'IoT repose sur le chiffrement du système de fichiers pour la sécurité de ces informations d'identification au repos.

Pour plus d’informations sur le chiffrement côté serveur et le chiffrement côté client, consultez les rubriques suivantes.

Table des matières

- Le chiffrement au repos dans AWS l'IoT FleetWise
- Gestion des clés dans AWS l'IoT FleetWise

Le chiffrement au repos dans AWS l'IoT FleetWise

AWS L'IoT FleetWise stocke vos données dans le AWS cloud et sur des passerelles.

Données au repos dans le AWS cloud

AWS L'IoT FleetWise stocke les données dans d'autres Services AWS systèmes qui chiffrent les données au repos par défaut. Encryption at rest s'intègre à AWS Key Management Service (AWS

KMS) pour gérer la clé de chiffrement utilisée pour chiffrer les valeurs des propriétés de vos actifs et les valeurs agrégées dans AWS l'IoT FleetWise. Vous pouvez choisir d'utiliser une clé gérée par le client pour chiffrer les valeurs des propriétés des actifs et les valeurs agrégées dans AWS l'IoT FleetWise. Vous pouvez créer, gérer et consulter votre clé de chiffrement via AWS KMS.

Vous pouvez choisir une clé gérée par le client Clé détenue par AWS ou une clé gérée par le client pour chiffrer vos données.

Comment ça marche

Le chiffrement au repos s'intègre AWS KMS à la gestion de la clé de chiffrement utilisée pour chiffrer vos données.

- Clé détenue par AWS — Clé de chiffrement par défaut. AWS C' FleetWise est l'IoT qui détient cette clé. Vous ne pouvez pas afficher, gérer ou utiliser cette clé dans votre Compte AWS. Vous ne pouvez pas non plus voir les opérations effectuées sur la clé dans AWS CloudTrail les journaux. Vous pouvez utiliser cette clé sans frais supplémentaires.
- Clé gérée par le client — La clé est stockée dans votre compte, que vous créez, détenez et gérez. Vous avez le contrôle total de la clé KMS. Des AWS KMS frais supplémentaires s'appliquent.

Clés détenues par AWS

Clés détenues par AWS ne sont pas enregistrés dans votre compte. Elles font partie d'un ensemble de clés KMS qui AWS possède et gère pour une utilisation multiple Comptes AWS. Services AWS peut être utilisé Clés détenues par AWS pour protéger vos données.

Vous ne pouvez ni afficher, ni gérer, ni utiliser Clés détenues par AWS, ni auditer leur utilisation. Cependant, il n'est pas nécessaire de prendre des mesures ou de modifier des programmes pour protéger les clés qui chiffrent vos données.

Aucuns frais ne vous seront facturés si vous l'utilisez Clés détenues par AWS, et ils ne sont pas pris en compte dans les AWS KMS quotas de votre compte.

Clés gérées par le client

Les clés gérées par le client sont des clés KMS de votre compte que vous créez, possédez et gérez. Vous avez le contrôle total de ces clés KMS, telles que les suivantes :

- Établir et maintenir leurs politiques clés, leurs politiques IAM et leurs subventions

- Les activer et les désactiver
- Rotation de leur matériel cryptographique
- Ajout de balises
- Création d'alias qui y font référence
- Planifier leur suppression

Vous pouvez également utiliser CloudTrail Amazon CloudWatch Logs pour suivre les demandes que AWS l'IoT FleetWise envoie AWS KMS en votre nom.

Si vous utilisez des clés gérées par le client, vous devez autoriser AWS l'IoT à FleetWise accéder à la clé KMS enregistrée dans votre compte. AWS L'IoT FleetWise utilise le chiffrement des enveloppes et la hiérarchie des clés pour chiffrer les données. Votre clé de chiffrement AWS KMS est utilisée pour chiffrer la clé racine de cette hiérarchie de clés. Pour plus d'informations, consultez Chiffrement d'enveloppe dans le Guide du développeur AWS Key Management Service .

L'exemple de politique suivant accorde à AWS l'IoT l' FleetWise autorisation d'utiliser votre AWS KMS clé.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Important**
>
> Lorsque vous ajoutez les nouvelles sections à votre politique de clé KMS, ne modifiez aucune section existante de la politique. AWS L'IoT ne FleetWise peut pas effectuer d'opérations sur vos données si le chiffrement est activé pour AWS l'IoT FleetWise et si l'une des conditions suivantes est vraie :
>
> - La clé KMS est désactivée ou supprimée.
> - La politique des clés KMS n'est pas correctement configurée pour le service.

Utilisation des données du système de vision avec chiffrement au repos

> **Note**
>
> Les données du système de vision sont en version préliminaire et sont susceptibles d'être modifiées.

Si vous avez un chiffrement géré par le client avec des AWS KMS clés activées sur votre FleetWise compte AWS IoT et que vous souhaitez utiliser les données du système de vision, réinitialisez vos paramètres de chiffrement pour qu'ils soient compatibles avec les types de données complexes. Cela permet FleetWise à AWS l'IoT d'établir les autorisations supplémentaires nécessaires pour les données du système de vision.

> **Note**
>
> Le manifeste de votre décodeur est peut-être bloqué dans un état de validation si vous n'avez pas réinitialisé les paramètres de chiffrement des données du système de vision.

1. Utilisez l'opération GetEncryptionConfigurationAPI pour vérifier si AWS KMS le chiffrement est activé. Aucune autre action n'est nécessaire si le type de cryptage est le cas`FLEETWISE_DEFAULT_ENCRYPTION`.
2. Si le type de chiffrement est`KMS_BASED_ENCRYPTION`, utilisez l'opération PutEncryptionConfigurationAPI pour réinitialiser le type de chiffrement à`FLEETWISE_DEFAULT_ENCRYPTION`.

```
aws iotfleetwise put-encryption-configuration \
      --encryption-type FLEETWISE_DEFAULT_ENCRYPTION
```

3. Utilisez l'opération PutEncryptionConfigurationAPI pour réactiver le type de chiffrement surKMS_BASED_ENCRYPTION.

```
aws iotfleetwise put-encryption-configuration \
        --encryption-type KMS_BASED_ENCRYPTION \
        --kms-key-id kms_key_id
```

Pour plus d'informations sur l'activation du chiffrement, consultezGestion des clés dans AWS l'IoT FleetWise.

Gestion des clés dans AWS l'IoT FleetWise

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

AWS Gestion des clés FleetWise dans le cloud IoT

Par défaut, AWS l' FleetWise IoT Clés gérées par AWS protège vos données dans le AWS Cloud. Vous pouvez mettre à jour vos paramètres pour utiliser une clé gérée par le client afin de chiffrer les données dans AWS l'IoT FleetWise. Vous pouvez créer, gérer et consulter votre clé de chiffrement via AWS Key Management Service (AWS KMS).

AWS L'IoT FleetWise prend en charge le chiffrement côté serveur en stockant des clés gérées par le client AWS KMS pour chiffrer les données des ressources suivantes.

AWS FleetWise Ressource IoT	Type de données	Champs chiffrés au repos à l'aide de clés gérées par le client
Catalogue de signaux		description

AWS FleetWise Ressource IoT	Type de données	Champs chiffrés au repos à l'aide de clés gérées par le client
	Attribut	description, AllowedValues, DefaultValue, min, max
	Actuator	description, AllowedValues, min, max
	Sensor	description, AllowedValues, min, max
Modèle de véhicule (manifeste du modèle)		description
Manifeste du décodeur		description
	CanInterface	Nom du protocole, version du protocole
	ObdInterface	requestMessageId, dtcReques tInterval secondes hasTransm issionEcu, norme OBD, secondes, pidRequestInterval useExtendedIds
	CanSignal	facteur, isBigEndian IsSigned, longueur, MessageId, offset, StartBit
	ObdSignal	ByteLength, offset, pid, mise à l'échelle, pidResponseLength ServiceMode, StartByte, bitMaskLe ngth bitRightShift
Véhicule		attributs
Campagne		description

AWS FleetWise Ressource IoT	Type de données	Champs chiffrés au repos à l'aide de clés gérées par le client
	conditionBasedCollectionSchéma	expression, minimumTriggerInterval Ms conditionLanguageVersion, TriggerMode
	TimeBasedCollectionScheme	Période DMS
Modèle d'état		description

> **Note**
>
> Les autres données et ressources sont cryptées à l'aide du chiffrement par défaut avec des clés gérées par AWS l'IoT FleetWise. Cette clé est créée et stockée dans le FleetWise compte AWS IoT.

Pour plus d'informations, voir Qu'est-ce que c'est AWS Key Management Service ? dans le Guide AWS Key Management Service du développeur.

Activer le chiffrement à l'aide des clés KMS (console)

Pour utiliser des clés gérées par le client avec AWS l'IoT FleetWise, vous devez mettre à jour vos FleetWise paramètres AWS IoT.

Pour activer le chiffrement à l'aide des clés KMS (console)

1. Ouvrez la FleetWise console AWS IoT.
2. Accédez aux paramètres.
3. Dans Chiffrement, choisissez Modifier pour ouvrir la page Modifier le chiffrement.
4. Pour le type de clé de chiffrement, choisissez Choisir une autre AWS KMS clé. Cela permet le chiffrement avec les clés gérées par le client stockées dans AWS KMS.

> **Note**
>
> Vous ne pouvez utiliser le chiffrement par clé géré par le client que pour les FleetWise ressources AWS IoT. Cela inclut le catalogue de signaux, le modèle de véhicule (manifeste du modèle), le manifeste du décodeur, le véhicule, la flotte et la campagne.

5. Choisissez votre clé KMS avec l'une des options suivantes :

   - Pour utiliser une clé KMS existante : choisissez l'alias de votre clé KMS dans la liste.
   - Pour créer une nouvelle clé KMS, choisissez Create an AWS KMS key.

   > **Note**
   >
   > Cela ouvre la AWS KMS console. Pour plus d'informations sur la création d'une clé KMS, consultez la section Création de clés dans le Guide du AWS Key Management Service développeur.

6. Choisissez Enregistrer pour mettre à jour vos paramètres.

Activer le chiffrement à l'aide de clés KMS (AWS CLI)

Vous pouvez utiliser l'opération PutEncryptionConfigurationAPI pour activer le chiffrement de votre FleetWise compte AWS IoT. L'exemple suivant utilise AWS CLI.

Pour activer le chiffrement, exécutez la commande suivante.

- Remplacez *kms_key_id* par l'ID de la clé KMS.

```
aws iotfleetwise put-encryption-configuration \
      --encryption-type KMS_BASED_ENCRYPTION \
      --kms-key-id kms_key_id
```

Example réponse

```
{
 "kmsKeyId": "customer_kms_key_id",
 "encryptionStatus": "PENDING",
 "encryptionType": "KMS_BASED_ENCRYPTION"
```

```
}
```

Politique de clé KMS

Après avoir créé une clé KMS, vous devez au minimum ajouter la déclaration suivante à votre politique de clé KMS pour qu'elle fonctionne avec AWS l'IoT FleetWise. Le principe du FleetWise service AWS IoT indiqué `iotfleetwise.amazonaws.com` dans la déclaration de politique clé du KMS permet FleetWise à AWS l'IoT d'accéder à la clé KMS.

```
{
  "Sid": "Allow FleetWise to encrypt and decrypt data when customer managed KMS key
 based encryption is enabled",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*"
}
```

La meilleure pratique en matière de sécurité consiste à ajouter `aws:SourceArn` et à `aws:SourceAccount` conditionner des clés à la politique des clés KMS. La clé de condition globale IAM `aws:SourceArn` permet de garantir que l' AWS IoT FleetWise utilise la clé KMS uniquement pour les ressources Amazon Resource Names () spécifiques au service. ARNs

Si vous définissez la valeur de`aws:SourceArn`, elle doit toujours être`arn:aws:iotfleetwise:us-east-1:account_id:*`. Cela permet à la clé KMS d'accéder à toutes les FleetWise ressources AWS IoT à cet effet Compte AWS. AWS L'IoT FleetWise prend en charge une clé KMS par compte pour toutes les ressources qu'il contient Région AWS. L'utilisation d'une autre valeur pour le champ de ressource ARN`SourceArn`, ou le fait de ne pas utiliser le caractère générique (*) pour le champ de ressource ARN, FleetWise empêche AWS l'IoT d'accéder à la clé KMS.

La valeur de `aws:SourceAccount` est votre identifiant de compte, qui est utilisé pour restreindre davantage la clé KMS afin qu'elle ne puisse être utilisée que pour votre compte spécifique. Si vous ajoutez `aws:SourceAccount` et `aws:SourceArn` conditionnez des clés à la clé KMS, assurez-vous que celle-ci n'est utilisée par aucun autre service ou compte. Cela permet d'éviter les défaillances.

La politique suivante inclut un principal de service (un identifiant pour un service), ainsi `aws:SourceAccount` qu'une `aws:SourceArn` configuration pour une utilisation en fonction de l'identifiant de votre compte Région AWS et de votre identifiant de compte.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:SourceAccount": "AWS-account-ID"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iotfleetwise:region:AWS-account-ID:*"
    }
  }
}
```

Pour plus d'informations sur la modification d'une politique de clé KMS pour une utilisation avec AWS l'IoT FleetWise, consultez la section Modification d'une politique clé dans le guide du AWS Key Management Service développeur.

> **Important**
>
> Lorsque vous ajoutez les nouvelles sections à votre politique de clé KMS, ne modifiez aucune section existante de la politique. AWS L'IoT ne FleetWise peut pas effectuer d'opérations sur vos données si le chiffrement est activé pour AWS l'IoT FleetWise et si l'une des conditions suivantes est vraie :
>
> - La clé KMS est désactivée ou supprimée.
> - La politique des clés KMS n'est pas correctement configurée pour le service.

Autorisations pour le AWS KMS chiffrement

Si vous avez activé AWS KMS le chiffrement, vous devez spécifier les autorisations dans la politique de rôle afin de pouvoir appeler AWS l'IoT FleetWise APIs. La politique suivante permet d'accéder à toutes les FleetWise actions AWS IoT, ainsi qu'à AWS KMS des autorisations spécifiques.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:*",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

La déclaration de politique suivante est requise pour que votre rôle puisse invoquer le chiffrement APIs. Cette déclaration de politique autorise `PutEncryptionConfiguration` et prend des `GetEncryptionConfiguration` mesures à partir de AWS l'IoT FleetWise.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:GetEncryptionConfiguration",
        "iotfleetwise:PutEncryptionConfiguration",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Restauration après suppression de AWS KMS la clé

Si vous supprimez une AWS KMS clé après avoir activé le chiffrement avec AWS l'IoT FleetWise, vous devez réinitialiser votre compte en supprimant toutes les données avant de FleetWise réutiliser l' AWS IoT. Vous pouvez utiliser les opérations de liste et de suppression de l'API pour nettoyer les ressources de votre compte.

Pour nettoyer les ressources de votre compte

1. Utilisez la liste APIs avec le `listResponseScope` paramètre défini sur`METADATA_ONLY`. Cela fournit une liste de ressources, y compris les noms des ressources et d'autres métadonnées telles que ARNs les horodatages.
2. Utilisez la commande Supprimer APIs pour supprimer des ressources individuelles.

Vous devez nettoyer les ressources dans l'ordre suivant.

1. Campagnes

    a. Répertoriez toutes les campagnes dont le `listResponseScope` paramètre est défini sur`METADATA_ONLY`.

   b. Supprimez les campagnes.

2. Flottes et véhicules

   a. Répertoriez toutes les flottes dont le `listResponseScope` paramètre est défini sur. `METADATA_ONLY`

   b. Répertoriez tous les véhicules de chaque flotte avec le `listResponseScope` paramètre défini sur`METADATA_ONLY`.

   c. Dissociez tous les véhicules de chaque flotte.

   d. Supprimez les flottes.

   e. Supprimez les véhicules.

3. Manifestes du décodeur

   a. Répertoriez tous les manifestes du décodeur dont le `listResponseScope` paramètre est défini sur. `METADATA_ONLY`

   b. Supprimez tous les manifestes du décodeur.

4. Modèles de véhicules (manifestes des modèles)

   a. Répertoriez tous les modèles de véhicules dont le `listResponseScope` paramètre est réglé sur`METADATA_ONLY`.

   b. Supprimez tous les modèles de véhicules.

5. Modèles d'états

   a. Répertoriez tous les modèles d'états dont le `listResponseScope` paramètre est défini sur`METADATA_ONLY`.

   b. Supprimez tous les modèles d'états.

6. Catalogues de signaux

   a. Répertoriez tous les catalogues de signaux.

   b. Supprimez tous les catalogues de signaux.

Contrôler l'accès avec AWS IoT FleetWise

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les sections suivantes expliquent comment contrôler l'accès à vos AWS IoT FleetWise ressources et à partir de celles-ci. Les informations qu'ils couvrent incluent la manière d'accorder l'accès à votre application afin que AWS l'IoT FleetWise puisse transférer les données des véhicules pendant les campagnes. Ils décrivent également comment vous pouvez accorder l' AWS IoT FleetWise accès à votre compartiment Amazon S3 (S3) ou à la base de données et à la table Amazon Timestream pour stocker des données, ou aux messages MQTT utilisés pour envoyer des données depuis des véhicules.

La technologie permettant de gérer toutes ces formes d'accès est AWS Identity and Access Management (IAM). Pour plus d'informations sur IAM, consultez En quoi consiste IAM ?.

Table des matières

- AWS IoT FleetWise Autoriser l'envoi et la réception de données sur un sujet MQTT
- Accorder l' AWS IoT FleetWise accès à une destination Amazon S3
- Autoriser AWS IoT FleetWise l'accès à une destination Amazon Timestream
- Accorder AWS IoT Device Management l'autorisation de générer la charge utile pour les commandes avec AWS IoT FleetWise

AWS IoT FleetWise Autoriser l'envoi et la réception de données sur un sujet MQTT

Lorsque vous utilisez un sujet MQTT, vos véhicules envoient des données à l'aide du courtier de messages AWS IoT MQTT. Vous devez accorder AWS IoT FleetWise l'autorisation de vous abonner à la rubrique MQTT que vous spécifiez. Si vous utilisez également des AWS IoT règles pour agir ou acheminer des données vers d'autres destinations, vous devez associer des politiques à un rôle IAM afin de permettre le transfert de données AWS IoT FleetWise vers les règles IoT.

En outre, vos autres applications ou appareils peuvent s'abonner au sujet que vous spécifiez pour recevoir les données du véhicule en temps quasi réel, et ces applications ou appareils doivent bénéficier des autorisations et des accès nécessaires.

Pour plus d'informations sur l'utilisation de MQTT ainsi que sur les rôles et autorisations requis, consultez :

- Protocoles de communication de l'appareil
- Règles pour AWS IoT
- Accorder à une AWS IoT règle l'accès dont elle a besoin
- Transmettre les autorisations de rôle

Avant de commencer, vérifiez les points suivants :

> **Important**
>
> - Vous devez utiliser la même AWS région lorsque vous créez des ressources de campagne sur les véhicules pour AWS l'IoT FleetWise. Si vous changez de AWS région, il se peut que vous rencontriez des problèmes pour accéder aux ressources.
> - AWS L'IoT FleetWise est disponible dans l'est des États-Unis (Virginie du Nord) et en Europe (Francfort).

Vous pouvez utiliser le AWS CLI pour créer un rôle IAM avec une politique de confiance pour la messagerie MQTT. Pour créer un rôle IAM, exécutez la commande suivante.

Pour créer un rôle IAM avec une politique de confiance

- *IotTopicExecutionRole*Remplacez-le par le nom du rôle que vous créez.
- *trust-policy*Remplacez-le par le fichier JSON qui contient la politique de confiance.

```
aws iam create-role --role-name IotTopicExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "mqttTopicTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:iotfleetwise:region:account-id:campaign/campaign-name"
          ],
          "aws:SourceAccount": [
            "account-id"
          ]
        }
      }
    }
  ]
}
```

Créez une politique d'autorisation pour autoriser AWS l'IoT FleetWise à publier des messages sur le sujet MQTT que vous avez spécifié. Pour créer une politique d'autorisations, exécutez la commande suivante.

Pour créer une politique d'autorisations

- *AWSIoTFleetwiseAccessIotTopicPermissionsPolicy*Remplacez-le par le nom de la politique que vous créez.
- *permissions-policy*Remplacez-le par le nom du fichier JSON contenant la politique d'autorisations.

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessIotTopicPermissionsPolicy --
policy-document file://permissions-policy.json
```

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

```
      {
        "Effect": "Allow",
        "Action": [
          "iot:Publish"
        ],
        "Resource": [
          "topic-arn"
        ]
      }
    ]
}
```

Pour associer la politique d'autorisations à votre rôle IAM

1. À partir de la sortie, copiez le nom de ressource Amazon (ARN) de la politique d'autorisation.
2. Pour associer la politique d'autorisations IAM à votre rôle IAM, exécutez la commande suivante.

   - *permissions-policy-arn*Remplacez-le par l'ARN que vous avez copié à l'étape précédente.
   - *IotTopicExecutionRole*Remplacez-le par le nom du rôle IAM que vous avez créé.

   ```
   aws iam attach-role-policy --policy-arn permissions-policy-arn --role-
   name IotTopicExecutionRole
   ```

Pour plus d'informations, consultez la section Gestion de l'accès aux AWS ressources dans le Guide de l'utilisateur IAM.

Accorder l' AWS IoT FleetWise accès à une destination Amazon S3

Lorsque vous utilisez une destination Amazon S3, que vous AWS IoT FleetWise transmettez les données du véhicule à votre compartiment S3 et que vous pouvez éventuellement utiliser une AWS KMS clé que vous possédez pour le chiffrement des données. Si la journalisation des erreurs est activée, elle envoie AWS IoT FleetWise également des erreurs de livraison de données à votre groupe de CloudWatch journaux et à vos flux. Vous devez disposer d'un rôle IAM lors de la création d'un flux de diffusion.

AWS IoT FleetWise utilise une politique de compartiment avec le principal de service pour la destination S3. Pour plus d'informations sur l'ajout de politiques de compartiment, consultez la section

Ajouter une politique de compartiment à l'aide de la console Amazon S3 dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Utilisez la politique d'accès suivante pour permettre AWS IoT FleetWise l'accès à votre compartiment S3. Si vous n'êtes pas propriétaire du compartiment S3, ajoutez `s3:PutObjectAcl` à la liste des actions Amazon S3. Cela donne au propriétaire du compartiment un accès complet aux objets livrés par AWS IoT FleetWise. Pour plus d'informations sur la manière dont vous pouvez sécuriser l'accès aux objets de vos compartiments, consultez les exemples de politiques relatives aux compartiments dans le guide de l'utilisateur d'Amazon Simple Storage Service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": "campaign-arn",
          "aws:SourceAccount": "account-id"
        }
      }
```

```
    }
  ]
}
```

La politique de bucket suivante s'applique à toutes les campagnes associées à un compte dans une AWS région.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:iotfleetwise:region:account-id:campaign/*",
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
```

```
}
```

Si une clé KMS est attachée à votre compartiment S3, elle doit respecter les règles suivantes. Pour plus d'informations sur la gestion des clés, consultez la section Protection des données à l'aide du chiffrement côté serveur à l'aide de AWS Key Management Service clés (SSE-KMS) dans le guide de l'utilisateur d'Amazon Simple Storage Service.

```
{
  "Version": "2012-10-17",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
   ],
  "Resource": "key-arn"
}
```

> **Important**
>
> Lorsque vous créez un bucket, S3 crée une liste de contrôle d'accès (ACL) par défaut qui accorde au propriétaire de la ressource un contrôle total sur la ressource. Si AWS l'IoT ne FleetWise peut pas fournir de données à S3, assurez-vous de désactiver l'ACL sur le compartiment S3. Pour plus d'informations, consultez la section Désactivation de tous ACLs les nouveaux compartiments et renforcement de la propriété des objets dans le guide de l'utilisateur d'Amazon Simple Storage Service.

Autoriser AWS IoT FleetWise l'accès à une destination Amazon Timestream

Lorsque vous utilisez une destination Timestream, AWS IoT FleetWise fournit les données du véhicule dans une table Timestream. Vous devez associer les politiques au rôle IAM pour autoriser l'envoi de données AWS IoT FleetWise à Timestream.

Si vous utilisez la console pour créer une campagne, AWS l'IoT associe FleetWise automatiquement la politique requise au rôle.

> **Note**
>
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

Avant de commencer, vérifiez les points suivants :

> **Important**
>
> - Vous devez utiliser la même AWS région lorsque vous créez des ressources Timestream pour l'IoT AWS . FleetWise Si vous changez de AWS région, il se peut que vous rencontriez des problèmes pour accéder aux ressources Timestream.
> - AWS L'IoT FleetWise est disponible dans l'est des États-Unis (Virginie du Nord), en Europe (Francfort) et en Asie-Pacifique (Mumbai).
> - Pour la liste des régions prises en charge, consultez la section Points de terminaison et quotas Timestream dans le. Références générales AWS

- Vous devez disposer d'une base de données Timestream. Pour un didacticiel, consultez la section Création d'une base de données dans le manuel Amazon Timestream Developer Guide.
- Vous devez avoir créé une table dans la base de données Timestream spécifiée. Pour un didacticiel, consultez la section Création d'un tableau dans le manuel Amazon Timestream Developer Guide.

Vous pouvez utiliser le AWS CLI pour créer un rôle IAM avec une politique de confiance pour Timestream. Pour créer un rôle IAM, exécutez la commande suivante.

Pour créer un rôle IAM avec une politique de confiance

- *`TimestreamExecutionRole`*Remplacez-le par le nom du rôle que vous créez.
- *`trust-policy`*Remplacez-le par le fichier .json qui contient la politique de confiance.

```
aws iam create-role --role-name TimestreamExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "timestreamTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
           "aws:SourceArn": [
            "arn:aws:iotfleetwise:region:account-id:campaign/campaign-name"
           ],
           "aws:SourceAccount": [
            "account-id"
          ]
        }
      }
    }
  ]
}
```

Créez une politique d'autorisation pour autoriser AWS l'IoT FleetWise à écrire des données dans Timestream. Pour créer une politique d'autorisations, exécutez la commande suivante.

Pour créer une politique d'autorisations

- *AWSIoTFleetwiseAccessTimestreamPermissionsPolicy*Remplacez-le par le nom de la politique que vous créez.
- *permissions-policy*Remplacez-le par le nom du fichier JSON contenant la politique d'autorisations.

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessTimestreamPermissionsPolicy --
policy-document file://permissions-policy.json
```

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

```
  "Statement": [
    {
      "Sid": "timestreamIngestion",
      "Effect": "Allow",
      "Action": [
        "timestream:WriteRecords",
        "timestream:Select",
        "timestream:DescribeTable"
      ],
      "Resource": "table-arn"
    },
    {
      "Sid": "timestreamDescribeEndpoint",
      "Effect": "Allow",
      "Action": [
        "timestream:DescribeEndpoints"
      ],
      "Resource": "*"
    }
  ]
}
```

Pour associer la politique d'autorisations à votre rôle IAM

1. À partir de la sortie, copiez le nom de ressource Amazon (ARN) de la politique d'autorisation.
2. Pour associer la politique d'autorisations IAM à votre rôle IAM, exécutez la commande suivante.

   - *permissions-policy-arn*Remplacez-le par l'ARN que vous avez copié à l'étape précédente.
   - *TimestreamExecutionRole*Remplacez-le par le nom du rôle IAM que vous avez créé.

   ```
   aws iam attach-role-policy --policy-arn permissions-policy-arn --role-
   name TimestreamExecutionRole
   ```

Pour plus d'informations, consultez la section Gestion de l'accès aux AWS ressources dans le Guide de l'utilisateur IAM.

Accorder AWS IoT Device Management l'autorisation de générer la charge utile pour les commandes avec AWS IoT FleetWise

Lorsque vous utilisez la fonction de commandes pour démarrer l'exécution d'une commande, AWS IoT Device Management la commande et ses paramètres sont extraits de la demande entrante. Il a ensuite besoin d'autorisations pour accéder aux FleetWise ressources de AWS l'IoT afin de valider la demande et de générer la charge utile. La charge utile est ensuite envoyée au véhicule par MQTT AWS IoT Device Management sur le sujet de demande de commande auquel votre véhicule s'est abonné.

Vous devez d'abord créer un rôle IAM qui accorde AWS IoT Device Management les autorisations requises pour générer la charge utile. Fournissez ensuite l'ARN de ce rôle à l'`CreateCommand`API à l'aide du `roleArn` champ. Vous trouverez ci-dessous quelques exemples de politiques.

> **Important**
>
> Pour le rôle IAM, vous devez utiliser le même Région AWS que celui dans lequel vous avez créé le véhicule et les ressources de commande. Si vous changez Région AWS, il se peut que vous rencontriez des problèmes pour accéder aux ressources.

Le rôle IAM doit respecter la politique de confiance suivante.

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

Accorder des autorisations à tous les véhicules (objets IoT)

L'exemple suivant montre comment accorder des autorisations pour générer la charge utile pour tous les véhicules enregistrés en tant qu' AWS IoT objets.

> **Note**
>
> - Cette politique peut être trop permissive. Utilisez le principe du moindre privilège pour vous assurer que vous n'accordez que les autorisations nécessaires.
> - Pour refuser les autorisations, passez plutôt `"Effect": "Allow"` `"Effect": "Deny"` à la politique IAM.

Dans cet exemple, remplacez :

- *<AWS_REGION>*avec l' Région AWS endroit où vous utilisez les FleetWise ressources de l' AWS IoT.
- *<ACCOUNT_ID>*avec ton Compte AWS numéro.

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

Accorder une autorisation à un véhicule spécifique (objet IoT)

L'exemple suivant montre comment accorder des autorisations pour générer la charge utile pour un véhicule spécifique enregistré en tant qu' AWS IoT objet.

Dans cet exemple, remplacez :

- *<AWS_REGION>*avec l' Région AWS endroit où vous utilisez les FleetWise ressources de l' AWS IoT.

- *<ACCOUNT_ID>*avec ton Compte AWS numéro.
- *<VEHICLE_NAME>*avec le nom IoT de votre véhicule.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>"
        }
    ]
}
```

Accorder des autorisations à des véhicules et à des signaux spécifiques

L'exemple suivant montre comment accorder des autorisations pour générer la charge utile de l'actionneur d'un véhicule spécifique.

Dans cet exemple, remplacez :

- *<AWS_REGION>*avec l' Région AWS endroit où vous utilisez les FleetWise ressources de l' AWS IoT.
- *<ACCOUNT_ID>*avec ton Compte AWS numéro.
- *<VEHICLE_NAME>*avec le nom IoT de votre véhicule.
- *<SIGNAL_FQN>*avec le nom du signal, tel que*<Vehicle.actuator2>*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "iotfleetwise:Signals": ["<SIGNAL_FQN>"]
                }
```

```
            }
        }
    ]
}
```

Accorder des autorisations à des véhicules et à des modèles d'états spécifiques

L'exemple suivant montre comment accorder des autorisations pour générer la charge utile pour un modèle de véhicule et d'état spécifiques.

Dans cet exemple, remplacez :

- *<AWS_REGION>*est l' Région AWS endroit où vous utilisez les FleetWise ressources de l' AWS IoT.
- *<ACCOUNT_ID>*est ton Compte AWS numéro.
- *<VEHICLE_NAME>*est le nom IoT de votre véhicule.
- *<STATE_TEMPLATE_ID>*avec l'identifiant de votre modèle d'état.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": [
                "arn:aws:iot:<AWS_REGION>:<ACCOUNT_ID>:thing/<VEHICLE_NAME>",
                "arn:aws:iotfleetwise:<AWS_REGION>:<ACCOUNT_ID>:state-
template/<STATE_TEMPLATE_ID>"]
        }
    ]
}
```

Accorder des autorisations pour utiliser les clés KMS gérées par le client

Si vous avez activé les clés KMS gérées par le client pour AWS IoT FleetWise, l'exemple suivant montre comment accorder des autorisations pour générer la charge utile.

Dans cet exemple, remplacez :

- *<AWS_REGION>*avec l' Région AWS endroit où vous utilisez les FleetWise ressources de l' AWS IoT.

- *<ACCOUNT_ID>*avec ton Compte AWS numéro.
- *<KMS_KEY_ID>*avec l'ID de votre clé KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:<AWS_REGION>:<ACCOUNT_ID>:key/<KMS_KEY_ID>"
        }
    ]
}
```

Identity and Access Management pour AWS l'IoT FleetWise

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources AWS IoT FleetWise . IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

- Public ciblé
- Authentification par des identités
- Gestion des accès à l’aide de politiques
- Comment AWS l'IoT FleetWise fonctionne avec l'IAM
- Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise
- Résolution des problèmes FleetWise d'identité et d'accès à l' AWS IoT
- AWS Référence des actions et des ressources de l' FleetWise API IoT

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans le domaine de AWS l'IoT FleetWise.

Utilisateur du service : si vous utilisez le FleetWise service AWS IoT pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez de plus en plus de FleetWise fonctionnalités de l' AWS IoT pour effectuer votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne pouvez pas accéder à une fonctionnalité de AWS l'IoT FleetWise, consultezRésolution des problèmes FleetWise d'identité et d'accès à l' AWS IoT.

Administrateur de services — Si vous êtes responsable des FleetWise ressources AWS IoT de votre entreprise, vous avez probablement un accès complet à AWS l'IoT FleetWise. C'est à vous de déterminer les FleetWise fonctionnalités et les ressources de AWS l'IoT auxquelles les utilisateurs de vos services doivent accéder. Vous devez ensuite soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser l'IAM avec AWS l'IoT FleetWise, consultezComment AWS l'IoT FleetWise fonctionne avec l'IAM.

Administrateur IAM : si vous êtes administrateur IAM, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à l'IoT AWS . FleetWise Pour consulter des exemples de politiques FleetWise basées sur l'identité AWS IoT que vous pouvez utiliser dans IAM, consultez. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAM Identity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec

des rôles IAM. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer des demandes vous-même, consultez AWS Signature Version 4 pour les demandes d'API dans le Guide de l'utilisateur IAM.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour plus d'informations, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Authentification multifactorielle AWS dans IAM dans le Guide de l'utilisateur IAM.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui vous imposent de vous connecter en tant qu'utilisateur racine, consultez Tâches nécessitant des informations d'identification d'utilisateur racine dans le Guide de l'utilisateur IAM.

Identité fédérée

La meilleure pratique consiste à obliger les utilisateurs humains, y compris ceux qui ont besoin d'un accès administrateur, à utiliser la fédération avec un fournisseur d'identité pour accéder à l'aide Services AWS d'informations d'identification temporaires.

Une identité fédérée est un utilisateur de l'annuaire des utilisateurs de votre entreprise, d'un fournisseur d'identité Web AWS Directory Service, du répertoire Identity Center ou de tout utilisateur qui y accède à l'aide des informations d'identification fournies Services AWS par le biais d'une source d'identité. Lorsque des identités fédérées y accèdent Comptes AWS, elles assument des rôles, qui fournissent des informations d'identification temporaires.

Pour une gestion des accès centralisée, nous vous recommandons d'utiliser AWS IAM Identity Center. Vous pouvez créer des utilisateurs et des groupes dans IAM Identity Center, ou vous pouvez vous connecter et synchroniser avec un ensemble d'utilisateurs et de groupes dans votre propre source d'identité afin de les utiliser dans toutes vos applications Comptes AWS et applications. Pour obtenir des informations sur IAM Identity Center, consultez Qu'est-ce que IAM Identity Center ? dans le Guide de l'utilisateur AWS IAM Identity Center .

Utilisateurs et groupes IAM

Un utilisateur IAM est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme telles que des mots de passe et des clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons d'effectuer une rotation des clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer les ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour plus d'informations, consultez Cas d'utilisation pour les utilisateurs IAM dans le Guide de l'utilisateur IAM.

Rôles IAM

Un rôle IAM est une identité au sein de vous Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Pour assumer temporairement un rôle IAM dans le AWS Management Console, vous pouvez passer d'un rôle d'utilisateur à un rôle IAM (console). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez Méthodes pour endosser un rôle dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré : pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès direct (FAS) : lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains

services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

- Rôle de service : il s'agit d'un rôle IAM attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui envoient des demandes AWS CLI d' AWS API. Cela est préférable au stockage des clés d'accès dans l' EC2 instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l' EC2 instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utiliser un rôle IAM pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le guide de l'utilisateur IAM.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez Vue d'ensemble des politiques JSON dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez Définition d'autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l'utilisateur IAM.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour découvrir comment choisir entre une politique gérée et une politique en ligne, consultez Choix entre les politiques gérées et les politiques en ligne dans le Guide de l'utilisateur IAM.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette

ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et AWS WAF Amazon VPC sont des exemples de services compatibles. ACLs Pour en savoir plus ACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d’autorisations qui vous sont accordées par des types de politiques plus courants.

- Limite d’autorisations : une limite d’autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d’autorisations qu’une politique basée sur l’identité peut accorder à une entité IAM (utilisateur ou rôle IAM). Vous pouvez définir une limite d’autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l’identité d’une entité et de ses limites d’autorisation. Les politiques basées sur les ressources qui spécifient l’utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d’autorisations. Un refus explicite dans l’une de ces politiques annule l’autorisation. Pour plus d’informations sur les limites d’autorisations, consultez Limites d’autorisations pour des entités IAM dans le Guide de l’utilisateur IAM.
- Politiques de contrôle des services (SCPs) : SCPs politiques JSON qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités

figurant dans les comptes des membres, y compris chacune Utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les Organizations SCPs, voir Politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.

- Politiques de contrôle des ressources (RCPs) : RCPs politiques JSON que vous pouvez utiliser pour définir le maximum d'autorisations disponibles pour les ressources de vos comptes sans mettre à jour les politiques IAM associées à chaque ressource que vous possédez. Le RCP limite les autorisations pour les ressources des comptes membres et peut avoir un impact sur les autorisations effectives pour les identités, y compris Utilisateur racine d'un compte AWS, qu'elles appartiennent ou non à votre organisation. Pour plus d'informations sur les Organizations RCPs, y compris une liste de ces Services AWS supports RCPs, consultez la section Resource control policies (RCPs) dans le guide de AWS Organizations l'utilisateur.
- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez Politiques de session dans le Guide de l'utilisateur IAM.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de l'utilisateur IAM.

Comment AWS l'IoT FleetWise fonctionne avec l'IAM

Avant d'utiliser l'IAM pour gérer l'accès à AWS l'IoT FleetWise, découvrez quelles fonctionnalités IAM peuvent être utilisées avec l'IoT AWS . FleetWise

Fonctionnalités IAM que vous pouvez utiliser avec l'IoT AWS FleetWise

Fonctionnalité IAM	AWS FleetWise Support pour l'IoT
Politiques basées sur l'identité	Oui

Fonctionnalité IAM	AWS FleetWise Support pour l'IoT
Politiques basées sur les ressources	Non
Actions de politique	Oui
Ressources de politique	Oui
Clés de condition de politique	Oui
ACLs	Non
ABAC (identifications dans les politiques)	Partielle
Informations d’identification temporaires	Oui
Autorisations de principal	Oui
Fonctions du service	Non
Rôles liés à un service	Non

Pour obtenir une vue d'ensemble de la façon dont AWS l'IoT FleetWise et les autres AWS services fonctionnent avec la plupart des fonctionnalités IAM, consultez la section AWS Services compatibles avec IAM dans le Guide de l'utilisateur IAM.

Politiques basées sur l'identité pour l'IoT AWS FleetWise

Prend en charge les politiques basées sur l’identité : oui

Les politiques basées sur l’identité sont des documents de politique d’autorisations JSON que vous pouvez attacher à une identité telle qu’un utilisateur, un groupe d’utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d’actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l’identité, consultez Définition d’autorisations IAM personnalisées avec des politiques gérées par le client dans le Guide de l’utilisateur IAM.

Avec les politiques IAM basées sur l’identité, vous pouvez spécifier des actions et ressources autorisées ou refusées, ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. Vous ne pouvez pas spécifier le principal dans une politique basée sur une identité,

car celle-ci s’applique à l’utilisateur ou au rôle auquel elle est attachée. Pour découvrir tous les éléments que vous utilisez dans une politique JSON, consultez Références des éléments de politique JSON IAM dans le Guide de l’utilisateur IAM.

Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Pour consulter des exemples de politiques FleetWise basées sur AWS l'identité de l'IoT, voir. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Politiques basées sur les ressources au sein de l'IoT AWS FleetWise

Prend en charge les politiques basées sur les ressources : non

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Par exemple, les politiques de confiance de rôle IAM et les politiques de compartiment Amazon S3 sont des politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l’accès à une ressource spécifique. Pour la ressource dans laquelle se trouve la politique, cette dernière définit quel type d’actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Pour permettre un accès intercompte, vous pouvez spécifier un compte entier ou des entités IAM dans un autre compte en tant que principal dans une politique basée sur les ressources. L’ajout d’un principal intercompte à une politique basée sur les ressources ne représente qu’une partie de l’instauration de la relation d’approbation. Lorsque le principal et la ressource sont différents Comptes AWS, un administrateur IAM du compte sécurisé doit également accorder à l'entité principale (utilisateur ou rôle) l'autorisation d'accéder à la ressource. Pour ce faire, il attache une politique basée sur une identité à l’entité. Toutefois, si une politique basée sur des ressources accorde l’accès à un principal dans le même compte, aucune autre politique basée sur l’identité n’est requise. Pour plus d’informations, consultez Accès intercompte aux ressources dans IAM dans le Guide de l’utilisateur IAM.

Actions politiques pour l' AWS IoT FleetWise

Prend en charge les actions de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une politique afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des FleetWise actions AWS IoT, voir Actions définies par AWS l'IoT FleetWise dans la référence d'autorisation de service.

Les actions politiques dans AWS l'IoT FleetWise utilisent le préfixe suivant avant l'action :

```
iotfleetwise
```

Pour indiquer plusieurs actions dans une seule déclaration, séparez-les par des virgules.

```
"Action": [
      "iotfleetwise:action1",
      "iotfleetwise: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 `List`, incluez l'action suivante :

```
"Action": "iotfleetwise:List*"
```

Pour consulter des exemples de politiques FleetWise basées sur AWS l'identité de l'IoT, voir. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Ressources relatives aux politiques relatives à AWS l'IoT FleetWise

Prend en charge les ressources de politique : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément de politique JSON `Resource` indique le ou les objets auxquels l'action s'applique. 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": "*"
```

Pour consulter la liste des types de FleetWise ressources AWS IoT et leurs caractéristiques ARNs, consultez la section Ressources définies par AWS l'IoT FleetWise dans la référence d'autorisation de service. Pour savoir avec quelles actions vous pouvez spécifier l'ARN de chaque ressource, consultez la section Actions définies par AWS l'IoT FleetWise .

Pour consulter des exemples de politiques FleetWise basées sur AWS l'identité de l'IoT, voir. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Clés de conditions politiques pour AWS l'IoT FleetWise

Prend en charge les clés de condition de politique spécifiques au service : oui

Les administrateurs peuvent utiliser les politiques AWS JSON pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Condition` (ou le bloc `Condition`) vous permet de spécifier des conditions lorsqu'une instruction est appliquée. L'élément `Condition` est facultatif. Vous pouvez créer des expressions conditionnelles qui utilisent des opérateurs de condition, tels que les signes égal ou inférieur à, pour faire correspondre la condition de la politique aux valeurs de la demande.

Si vous spécifiez plusieurs éléments `Condition` dans une instruction, ou plusieurs clés dans un seul élément `Condition`, AWS les évalue à l'aide d'une opération `AND` logique. Si vous spécifiez

plusieurs valeurs pour une seule clé de condition, AWS évalue la condition à l'aide d'une `OR` opération logique. Toutes les conditions doivent être remplies avant que les autorisations associées à l'instruction ne soient accordées.

Vous pouvez aussi utiliser des variables d'espace réservé quand vous spécifiez des conditions. Par exemple, vous pouvez accorder à un utilisateur IAM l'autorisation d'accéder à une ressource uniquement si elle est balisée avec son nom d'utilisateur IAM. Pour plus d'informations, consultez Éléments d'une politique IAM : variables et identifications dans le Guide de l'utilisateur IAM.

AWS prend en charge les clés de condition globales et les clés de condition spécifiques au service. Pour voir toutes les clés de condition AWS globales, voir les clés de contexte de condition AWS globales dans le guide de l'utilisateur IAM.

Pour consulter la liste des clés de FleetWise condition AWS IoT, consultez la section Clés de condition pour AWS l'IoT FleetWise dans la référence d'autorisation de service. Pour savoir avec quelles actions et ressources vous pouvez utiliser une clé de condition, consultez la section Actions définies par AWS l'IoT FleetWise .

Pour consulter des exemples de politiques FleetWise basées sur AWS l'identité de l'IoT, voir. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Listes de contrôle d'accès (ACLs) dans AWS l'IoT FleetWise

Supports ACLs : Non

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLs sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Contrôle d'accès basé sur les attributs (ABAC) avec IoT AWS FleetWise

Prend en charge ABAC (identifications dans les politiques) : partiellement

Le contrôle d'accès par attributs (ABAC) est une stratégie d'autorisation qui définit des autorisations en fonction des attributs. Dans AWS, ces attributs sont appelés balises. Vous pouvez associer des balises aux entités IAM (utilisateurs ou rôles) et à de nombreuses AWS ressources. L'étiquetage des entités et des ressources est la première étape d'ABAC. Vous concevez ensuite des politiques ABAC pour autoriser des opérations quand l'identification du principal correspond à celle de la ressource à laquelle il tente d'accéder.

L'ABAC est utile dans les environnements qui connaissent une croissance rapide et pour les cas où la gestion des politiques devient fastidieuse.

Pour contrôler l'accès basé sur des étiquettes, vous devez fournir les informations d'étiquette dans l'élément de condition d'une politique utilisant les clés de condition `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Si un service prend en charge les trois clés de condition pour tous les types de ressources, alors la valeur pour ce service est Oui. Si un service prend en charge les trois clés de condition pour certains types de ressources uniquement, la valeur est Partielle.

Pour plus d'informations sur ABAC, consultez Définition d'autorisations avec l'autorisation ABAC dans le Guide de l'utilisateur IAM. Pour accéder à un didacticiel décrivant les étapes de configuration de l'ABAC, consultez Utilisation du contrôle d'accès par attributs (ABAC) dans le Guide de l'utilisateur IAM.

> **Note**
>
> AWS L'IoT FleetWise ne prend en charge `iam:PassRole` que ce qui est requis pour le fonctionnement de l'`CreateCampaign`API.

Utilisation d'informations d'identification temporaires avec AWS l'IoT FleetWise

Prend en charge les informations d'identification temporaires : oui

Certains Services AWS ne fonctionnent pas lorsque vous vous connectez à l'aide d'informations d'identification temporaires. Pour plus d'informations, y compris celles qui Services AWS fonctionnent avec des informations d'identification temporaires, consultez Services AWS la section relative à l'utilisation d'IAM dans le guide de l'utilisateur d'IAM.

Vous utilisez des informations d'identification temporaires si vous vous connectez à l' AWS Management Console aide d'une méthode autre qu'un nom d'utilisateur et un mot de passe. Par exemple, lorsque vous accédez à AWS l'aide du lien d'authentification unique (SSO) de votre entreprise, ce processus crée automatiquement des informations d'identification temporaires. Vous créez également automatiquement des informations d'identification temporaires lorsque vous vous connectez à la console en tant qu'utilisateur, puis changez de rôle. Pour plus d'informations sur le changement de rôle, consultez Passage d'un rôle utilisateur à un rôle IAM (console) dans le Guide de l'utilisateur IAM.

Vous pouvez créer manuellement des informations d'identification temporaires à l'aide de l' AWS API AWS CLI or. Vous pouvez ensuite utiliser ces informations d'identification temporaires pour y accéder AWS. AWS recommande de générer dynamiquement des informations d'identification temporaires au lieu d'utiliser des clés d'accès à long terme. Pour plus d'informations, consultez Informations d'identification de sécurité temporaires dans IAM.

Autorisations principales interservices pour l'IoT AWS FleetWise

Prend en charge les sessions d'accès direct (FAS) : oui

Lorsque vous utilisez un utilisateur ou un rôle IAM pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre service. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur une politique lors de la formulation de demandes FAS, consultez Transmission des sessions d'accès.

Rôles de service pour AWS l'IoT FleetWise

Prend en charge les rôles de service : Non

Un rôle de service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer un rôle de service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **Warning**
>
> La modification des autorisations associées à un rôle de service peut perturber les FleetWise fonctionnalités de AWS l'IoT. Modifiez les rôles de service uniquement lorsque AWS l'IoT FleetWise fournit des instructions à cet effet.

Rôles liés aux services pour l'IoT AWS FleetWise

Prend en charge les rôles liés à un service : non

Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.

Pour plus d'informations sur la création ou la gestion des rôles liés à un service, consultez Services AWS qui fonctionnent avec IAM. Recherchez un service dans le tableau qui inclut un `Yes` dans la colonne Rôle lié à un service. Choisissez le lien Oui pour consulter la documentation du rôle lié à ce service.

Utilisation de rôles liés à un service pour AWS IoT FleetWise

AWS IoT FleetWise utilise des rôles AWS Identity and Access Management liés à un service (IAM). Un rôle lié à un service est un type unique de rôle IAM directement lié à AWS IoT. FleetWise Les rôles liés aux services sont prédéfinis par AWS IoT FleetWise et incluent les autorisations dont AWS IoT a FleetWise besoin pour envoyer des métriques à Amazon. CloudWatch Pour de plus amples informations, veuillez consulter Surveillez AWS l'IoT FleetWise avec Amazon CloudWatch.

Un rôle lié à un service FleetWise accélère la configuration d'AWS IoT, car il n'est pas nécessaire d'ajouter manuellement les autorisations nécessaires. AWS IoT FleetWise définit les autorisations associées à ses rôles liés aux services et, sauf indication contraire, seul AWS IoT FleetWise peut assumer ses rôles. Les autorisations définies comprennent la politique d'approbation et la politique d'autorisations. Cette politique d'autorisation ne peut être attachée à aucune autre entité IAM.

Vous pouvez supprimer un rôle lié à un service uniquement après la suppression préalable de ses ressources connexes. Cela protège vos FleetWise ressources AWS IoT, car vous ne pouvez pas supprimer par inadvertance l'autorisation d'accès aux ressources.

Pour plus d'informations sur les autres services qui prennent en charge les rôles liés à un service, consultez la section AWS Services qui fonctionnent avec IAM et recherchez les services dont la valeur est Oui dans la colonne Rôles liés à un service. Pour consulter la documentation du rôle lié à un service, choisissez un Oui ayant un lien.

Autorisations de rôle liées à un service pour AWS IoT FleetWise

AWS IoT FleetWise utilise le rôle lié à un service nommé AWSServiceRoleForIoTFleetWise, une politique gérée par AWS qui est utilisée pour toutes les out-of-the-box autorisations relatives à AWS IoT. FleetWise

Le rôle lié au service AWSService RoleForIo TFleet Wise fait confiance aux services suivants pour assumer le rôle :

- `IoTFleetWise`

La politique d'autorisation de rôle nommée AWSIo TFleetwise ServiceRolePolicy permet FleetWise à AWS IoT d'effectuer les actions suivantes sur les ressources spécifiées :

- Action : `cloudwatch:PutMetricData` sur la ressource : *

Vous devez configurer les autorisations de manière à permettre à une entité IAM (comme un utilisateur, un groupe ou un rôle) de créer, modifier ou supprimer un rôle lié à un service. Pour plus d'informations, consultez Autorisations de rôles liés à un service dans le Guide de l'utilisateur IAM.

Création d'un rôle lié à un service pour AWS IoT FleetWise

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous enregistrez un compte dans la FleetWise console AWS IoT, ou dans l' AWS CLI AWS API, AWS IoT FleetWise crée pour vous le rôle lié au service. Pour de plus amples informations, veuillez consulter Configurez vos FleetWise paramètres AWS IoT.

Création d'un rôle lié à un service dans AWS IoT FleetWise (console)

Vous n'avez pas besoin de créer manuellement un rôle lié à un service. Lorsque vous enregistrez un compte dans la FleetWise console AWS IoT, la AWS CLI ou l' AWS API, AWS IoT FleetWise crée le rôle lié au service pour vous.

Modification d'un rôle lié à un service pour AWS IoT FleetWise

Vous ne pouvez pas modifier le rôle lié au service AWSService RoleForIo TFleet Wise dans AWS IoT. FleetWise Étant donné que différentes entités peuvent faire référence à un rôle lié à un service que vous créez, vous ne pouvez pas modifier le nom du rôle. Néanmoins, vous pouvez modifier la description du rôle à l'aide d'IAM. Pour plus d'informations, consultez Modification d'un rôle lié à un service dans le IAM Guide de l'utilisateur.

Nettoyer un rôle lié à un service

Avant de pouvoir utiliser IAM pour supprimer un rôle lié à un service, vous devez supprimer toutes les ressources utilisées par le rôle.

> **Note**
>
> Si AWS IoT FleetWise utilise le rôle lorsque vous essayez de supprimer les ressources, la suppression risque d'échouer. Si cela se produit, patientez quelques minutes et réessayez. Pour savoir comment supprimer le service-linked-role via la console, la AWS CLI ou l' AWS API, consultez la section Utilisation des rôles liés à un service dans le guide de l'utilisateur IAM.

Si vous supprimez ce rôle lié à un service, puis que vous devez le créer à nouveau, vous pouvez créer un compte auprès d'AWS IoT. FleetWise AWS IoT crée FleetWise ensuite à nouveau le rôle lié au service pour vous.

Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

Par défaut, les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier FleetWise des ressources AWS IoT. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'API AWS Management Console, AWS Command Line Interface (AWS CLI) ou de AWS l'API. Pour 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.

Pour apprendre à créer une politique basée sur l’identité IAM à l’aide de ces exemples de documents de politique JSON, consultez Création de politiques IAM (console) dans le Guide de l’utilisateur IAM.

Pour plus de détails sur les actions et les types de ressources définis par l' AWS IoT FleetWise, y compris le ARNs format de chaque type de ressource, voir Actions, ressources et clés de condition pour l' AWS IoT FleetWise dans la référence d'autorisation de service.

Rubriques

- Bonnes pratiques en matière de politiques
- Utilisation de la FleetWise console AWS IoT
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Accédez aux ressources dans Amazon Timestream

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer FleetWise 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 :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez politiques gérées par AWS ou politiques gérées par AWS pour les activités professionnelles dans le Guide de l'utilisateur IAM.
- Accordez les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation d'IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utilisez des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, consultez Conditions pour éléments de politique JSON IAM dans le Guide de l'utilisateur IAM.
- Utilisez l'Analyseur d'accès IAM pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : l'Analyseur d'accès IAM valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politiques avec IAM Access Analyzer dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue.

Compte AWS Pour exiger la MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Sécurisation de l'accès aux API avec MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

Utilisation de la FleetWise console AWS IoT

Pour accéder à la FleetWise console AWS IoT, vous devez disposer d'un ensemble minimal d'autorisations. Ces autorisations doivent vous permettre de répertorier et d'afficher des informations détaillées sur les FleetWise ressources AWS IoT de votre Compte AWS. Si vous créez une politique basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette politique.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API qu'ils tentent d'effectuer.

Pour garantir que les utilisateurs et les rôles peuvent toujours utiliser la FleetWise console AWS IoT, associez également la politique AWS IoT FleetWise `ConsoleAccess` ou `ReadOnly` AWS gérée aux entités. Pour plus d'informations, consultez Ajout d'autorisations à un utilisateur dans le Guide de l'utilisateur IAM.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
```

```
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Accédez aux ressources dans Amazon Timestream

Avant d'utiliser l' AWS IoT FleetWise, vous devez enregistrer votre AWS compte, les ressources IAM et Amazon Timestream pour AWS autoriser l' FleetWise IoT à envoyer des AWS Cloud données du véhicule en votre nom. Pour vous inscrire, vous devez :

- Une base de données Amazon Timestream.
- Table créée dans la base de données Amazon Timestream spécifiée.
- Rôle IAM qui permet FleetWise à AWS l'IoT d'envoyer des données à Amazon Timestream.

Pour plus d'informations, notamment sur les procédures et les exemples de politiques, consultezConfigurez vos FleetWise paramètres AWS IoT.

Résolution des problèmes FleetWise d'identité et d'accès à l' AWS IoT

Utilisez les informations suivantes pour vous aider à diagnostiquer et à résoudre les problèmes courants que vous pouvez rencontrer lorsque vous travaillez avec AWS l'IoT FleetWise et l'IAM.

Rubriques

- Je ne suis pas autorisé à effectuer une action dans AWS l'IoT FleetWise
- Je ne suis pas autorisé à effectuer iam : PassRole
- Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes FleetWise ressources AWS IoT

Je ne suis pas autorisé à effectuer une action dans AWS l'IoT FleetWise

S'il vous AWS Management Console indique que vous n'êtes pas autorisé à effectuer une action, vous devez contacter votre administrateur pour obtenir de l'aide. Votre administrateur est la personne qui vous a fourni vos informations de connexion.

L'exemple d'erreur suivant se produit lorsque l'utilisateur `mateojackson` IAM essaie d'utiliser la console pour afficher les détails d'une *`myVehicle`* ressource fictive mais ne dispose pas des `iotfleetwise:`*`GetVehicleStatus`* autorisations nécessaires.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotfleetwise:GetVehicleStatus on resource: myVehicle
```

Dans ce cas, Mateo demande à son administrateur de mettre à jour ses politiques pour lui permettre d'accéder à la ressource *`myVehicle`* à l'aide de l'action `iotfleetwise:`*`GetVehicleStatus`*.

Je ne suis pas autorisé à effectuer iam : PassRole

Si vous recevez un message d'erreur indiquant que vous n'êtes pas autorisé à effectuer l'`iam:PassRole`action, vos politiques doivent être mises à jour pour vous permettre de transférer un rôle à AWS l'IoT FleetWise.

Certains vous Services AWS permettent de transmettre un rôle existant à ce service au lieu de créer un nouveau rôle de service ou un rôle lié à un service. Pour ce faire, 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 effectuer une action dans l' AWS IoT FleetWise. 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 politiques de Mary doivent être mises à jour pour lui permettre d'exécuter l'action `iam:PassRole`.

Si vous avez besoin d'aide, contactez votre AWS administrateur. Votre administrateur vous a fourni vos informations d'identification de connexion.

Je souhaite permettre à des personnes extérieures Compte AWS à moi d'accéder à mes FleetWise 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 qui est autorisé à assumer le rôle. Pour les services qui prennent en charge les politiques basées sur les ressources ou les listes de contrôle d'accès (ACLs), vous pouvez utiliser ces politiques pour autoriser les utilisateurs à accéder à vos ressources.

Pour plus d'informations, consultez les éléments suivants :

- Pour savoir si AWS l'IoT FleetWise prend en charge ces fonctionnalités, consultezComment AWS l'IoT FleetWise fonctionne avec l'IAM.
- Pour savoir comment fournir l'accès à vos ressources sur celles Comptes AWS que vous possédez, consultez la section Fournir l'accès à un utilisateur IAM dans un autre utilisateur Compte AWS que vous possédez dans le Guide de l'utilisateur IAM.
- Pour savoir comment fournir l'accès à vos ressources à des tiers Comptes AWS, consultez la section Fournir un accès à des ressources Comptes AWS détenues par des tiers dans le guide de l'utilisateur IAM.
- Pour savoir comment fournir un accès par le biais de la fédération d'identité, consultez Fournir un accès à des utilisateurs authentifiés en externe (fédération d'identité) dans le Guide de l'utilisateur IAM.
- Pour en savoir plus sur la différence entre l'utilisation des rôles et des politiques basées sur les ressources pour l'accès intercompte, consultez Accès intercompte aux ressources dans IAM dans le Guide de l'utilisateur IAM.

AWS Référence des actions et des ressources de l' FleetWise API IoT

Lorsque vous gérez l'accès à l'aide de politiques pour une identité IAM, vous pouvez utiliser le tableau suivant comme référence. Le tableau répertorie chaque FleetWise API AWS IoT, les actions correspondantes pour lesquelles vous pouvez accorder des autorisations pour effectuer l'action et la AWS ressource pour laquelle vous pouvez accorder les autorisations.

Spécifiez les actions dans le champ `Action` de la stratégie, et la valeur des ressources dans le champ `Resource` de la stratégie. Pour spécifier une action, utilisez le `iotfleetwise:` préfixe suivi du nom de l'action. Par exemple, `iotfleetwise:CreateSignalCatalog`.

Actuellement, AWS l'IoT FleetWise prend en charge les actions, ressources et clés de condition suivantes.

Utilisez la barre de défilement pour voir le reste du tableau.

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
AssociateVehicleFleet	IoT au niveau de la flotte : Associate VehicleFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id*
CreateCampaign	IoT au niveau de la flotte : CreateCam paign	arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name* arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*
CreateDecoderManif est	IoT au niveau de la flotte : CreateDec oderManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
		arn ::iotfleetwise : ::model-manifest/ *partition region account_id name*
CreateFleet	IoT au niveau de la flotte : CreateFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
CreateModelManifest	IoT au niveau de la flotte : CreateModelManifest	arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
CreateSignalCatalog	IoT au niveau de la flotte : CreateSignalCatalog	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
CreateStateTemplate	IoT au niveau de la flotte : CreateStateTemplate	arn ::iotfleetwise : ::state-template/ *partition region account_id state-template-id* arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
CreateVehicle	IoT au niveau de la flotte : CreateVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
BatchCreateVehicle	IoT au niveau de la flotte : CreateVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
DeleteCampaign	IoT au niveau de la flotte : DeleteCampaign	arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name*
DeleteDecoderManifest	IoT au niveau de la flotte : DeleteDecoderManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
DeleteFleet	IoT au niveau de la flotte : DeleteFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*
DeleteModelManifest	IoT au niveau de la flotte : DeleteModelManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
DeleteSignalCatalog	IoT au niveau de la flotte : DeleteSignalCatalog	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
DeleteStateTemplate	IoT au niveau de la flotte : DeleteStateTemplate	arn ::iotfleetwise : ::state-template/ *partition region account_id state-template-id*
DeleteVehicle	IoT au niveau de la flotte : DeleteVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
DisassociateVehicleFleet	IoT au niveau de la flotte : DisassociateVehicleFleet	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*
GetCampaign	IoT au niveau de la flotte : GetCampaign	arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name*
GetDecoderManifest	IoT au niveau de la flotte : GetDecoderManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
GetEncryptionConfiguration	IoT au niveau de la flotte : GetEncryptionConfiguration	
GetFleet	IoT au niveau de la flotte : GetFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*
GetLoggingOptions	IoT au niveau de la flotte : GetLoggingOptions	
GetModelManifest	IoT au niveau de la flotte : GetModelManifest	arn ::iotfleetwise : ::model-manifest/ *partition region account_id name*
GetRegisterAccountStatus	IoT au niveau de la flotte : GetRegisterAccountStatus	
GetSignalCatalog	IoT au niveau de la flotte : GetSignalCatalog	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
GetStateTemplate	IoT au niveau de la flotte : GetStateTemplate	arn ::iotfleetwise : ::state-template/ *partition region account_id state-template-id*
GetVehicle	IoT au niveau de la flotte : GetVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id*
GetVehicleStatus	IoT au niveau de la flotte : GetVehicleStatus	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id*
ImportDecoderManifest	IoT au niveau de la flotte : ImportDecoderManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
ImportSignalCatalog	IoT au niveau de la flotte : ImportSignalCatalog	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
ListCampaigns	IoT au niveau de la flotte : ListCampaigns	
ListDecoderManifestNetworkInterfaces	IoT au niveau de la flotte : ListDecoderManifestNetworkInterfaces	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
ListDecoderManifests	IoT au niveau de la flotte : ListDecoderManifests	
ListDecoderManifestSignals	IoT au niveau de la flotte : ListDecoderManifestSignals	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
ListFleets	IoT au niveau de la flotte : ListFleets	
ListFleetsForVehicle	IoT au niveau de la flotte : ListFleetsForVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id*
ListModelManifestNodes	IoT au niveau de la flotte : ListModelManifestNodes	arn ::iotfleetwise : ::model-manifest/ *partition region account_id name*
ListModelManifests	IoT au niveau de la flotte : ListModelManifests	
ListSignalCatalogNodes	IoT au niveau de la flotte : ListSignalCatalogNodes	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
ListSignalCatalogs	IoT au niveau de la flotte : ListSignalCatalogs	
ListStateTemplates	IoT au niveau de la flotte : ListStateTemplates	
ListVehicles	IoT au niveau de la flotte : ListVehicles	
ListVehiclesInFleet	IoT au niveau de la flotte : ListVehiclesInFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
ListTagsForResource	IoT au niveau de la flotte : ListTagsForResource	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name* arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise : ::state-template/ *partition region account_id state-tem plate-id*
PutEncryptionConfiguration	IoT au niveau de la flotte : PutEncryptionConfiguration	
PutLoggingOptions	IoT au niveau de la flotte : PutLoggingOptions	
RegisterAccount	IoT au niveau de la flotte : RegisterAccount	

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
TagResource	IoT au niveau de la flotte : TagResource	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name* arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise : ::state-template/ *partition region account_id state-tem plate-id*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
UntagResource	IoT au niveau de la flotte : UntagResource	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name* arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise : ::state-template/ *partition region account_id state-tem plate-id*
UpdateCampaign	IoT au niveau de la flotte : UpdateCam paign	arn ::iotfleetwise : ::campaign/ *partition region account_id campaign-name*
UpdateDecoderManif est	IoT au niveau de la flotte : UpdateDec oderManifest	arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
UpdateFleet	IoT au niveau de la flotte : UpdateFleet	arn ::iotfleetwise : ::fleet/ *partition region account_id fleet-id*
UpdateModelManifest	IoT au niveau de la flotte : UpdateMod elManifest	arn ::iotfleetwise : ::model-manifest/ *partition region account_id name*

AWS FleetWise Actions liées à l'IoT	Autorisations requises	Ressources
UpdateSignalCatalog	IoT au niveau de la flotte : UpdateSignalCatalog	arn ::iotfleetwise : ::signal-catalog/ *partition region account_id name*
UpdateStateTemplate	IoT au niveau de la flotte : UpdateStateTemplate	arn ::iotfleetwise : ::state-template/ *partition region account_id state-template-id*
UpdateVehicle	IoT au niveau de la flotte : UpdateVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*
BatchUpdateVehicle	IoT au niveau de la flotte : UpdateVehicle	arn ::iotfleetwise : ::vehicle/ *partition region account_id vehicle-id* arn ::iotfleetwise : ::model-manifest/ *partition region account_id name* arn ::iotfleetwise : :decoder-manifest/ *partition region account_id name*

Validation de conformité pour AWS l'IoT FleetWise

> **Note**
>
> AWS L'IoT FleetWise n'entre dans le champ d'aucun programme de AWS conformité.

Pour savoir si un programme Services AWS de conformité Service AWS s'inscrit dans le champ d'application de programmes de conformité spécifiques, consultez Services AWS la section de

conformité et sélectionnez le programme de conformité qui vous intéresse. Pour des informations générales, voir Programmes de AWS conformité Programmes AWS de .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Votre responsabilité en matière de conformité lors de l'utilisation Services AWS est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Conformité et gouvernance de la sécurité : ces guides de mise en œuvre de solutions traitent des considérations architecturales et fournissent les étapes à suivre afin de déployer des fonctionnalités de sécurité et de conformité.
- Référence des services éligibles HIPAA : liste les services éligibles HIPAA. Tous ne Services AWS sont pas éligibles à la loi HIPAA.
- AWS Ressources de https://aws.amazon.com/compliance/resources/ de conformité — Cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- AWS Guides de conformité destinés aux clients — Comprenez le modèle de responsabilité partagée sous l'angle de la conformité. Les guides résument les meilleures pratiques en matière de sécurisation Services AWS et décrivent les directives relatives aux contrôles de sécurité dans de nombreux cadres (notamment le National Institute of Standards and Technology (NIST), le Payment Card Industry Security Standards Council (PCI) et l'Organisation internationale de normalisation (ISO)).
- Évaluation des ressources à l'aide des règles du guide du AWS Config développeur : le AWS Config service évalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub— Cela Service AWS fournit une vue complète de votre état de sécurité interne 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, consultez Référence des contrôles Security Hub.
- Amazon GuardDuty — Cela Service AWS détecte les menaces potentielles qui pèsent sur vos charges de travail Comptes AWS, vos conteneurs et vos données en surveillant votre environnement pour détecter toute activité suspecte et malveillante. GuardDuty peut vous aider à répondre à diverses exigences de conformité, telles que la norme PCI DSS, en répondant aux exigences de détection des intrusions imposées par certains cadres de conformité.

- AWS Audit Manager— Cela vous Service AWS permet d'auditer en permanence votre AWS utilisation afin de simplifier la gestion des risques et la conformité aux réglementations et aux normes du secteur.

Résilience dans AWS l'IoT FleetWise

L'infrastructure AWS mondiale est construite autour des AWS régions et des zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, 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 à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

> **Note**
>
> Les données traitées par AWS l'IoT sont FleetWise stockées dans une base de données Amazon Timestream. Timestream prend en charge les sauvegardes vers d'autres zones de AWS disponibilité ou régions. Cependant, vous pouvez écrire votre propre application à l'aide du SDK Timestream pour interroger les données et les enregistrer dans la destination de votre choix.
>
> Pour plus d'informations sur Amazon Timestream, consultez le manuel du développeur Amazon Timestream.
>
> Amazon Timestream n'est pas disponible dans la région Asie-Pacifique (Mumbai).

La sécurité des infrastructures dans AWS l'IoT FleetWise

En tant que service géré, AWS l'IoT FleetWise est protégé par la sécurité du réseau AWS mondial. Pour plus d'informations sur les services AWS de sécurité et sur la manière dont AWS l'infrastructure est protégée, consultez la section Sécurité du AWS cloud. Pour concevoir votre AWS environnement en utilisant les meilleures pratiques en matière de sécurité de l'infrastructure, consultez la section Protection de l'infrastructure dans le cadre AWS bien architecturé du pilier de sécurité.

Vous utilisez des appels d'API AWS publiés pour accéder à AWS l'IoT FleetWise via le réseau. Les clients doivent prendre en charge les éléments suivants :

- Protocole TLS (Transport Layer Security). Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Ses suites de chiffrement PFS (Perfect Forward Secrecy) comme DHE (Ephemeral Diffie-Hellman) ou ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Vous pouvez appeler ces opérations d'API depuis n'importe quel emplacement du réseau, mais AWS l'IoT FleetWise prend en charge les politiques d'accès basées sur les ressources, qui peuvent inclure des restrictions basées sur l'adresse IP source. Vous pouvez également utiliser FleetWise les politiques AWS IoT pour contrôler l'accès à partir de points de terminaison Amazon Virtual Private Cloud (Amazon VPC) spécifiques ou spécifiques. VPCs En fait, cela isole l'accès réseau à une FleetWise ressource AWS IoT donnée uniquement du VPC spécifique au sein AWS du réseau.

Rubriques

- Connexion à l' AWS IoT FleetWise via un point de terminaison VPC d'interface

Connexion à l' AWS IoT FleetWise via un point de terminaison VPC d'interface

Vous pouvez vous connecter directement à l' AWS IoT en FleetWise utilisant un point de terminaison VPC d'interface (AWS PrivateLink) dans votre Virtual Private Cloud (VPC), au lieu de vous connecter via Internet. Lorsque vous utilisez un point de terminaison VPC d'interface, la communication entre votre VPC et AWS l' FleetWise IoT s'effectue entièrement au sein du réseau. AWS Chaque point de terminaison VPC est représenté par une ou plusieurs interfaces réseau élastiques (ENIs) avec des adresses IP privées dans vos sous-réseaux VPC.

Le point de terminaison VPC de l'interface connecte votre VPC directement à l' AWS IoT FleetWise sans passerelle Internet, appareil NAT, connexion VPN ou connexion. AWS Direct Connect Les instances de votre VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec l'API AWS IoT FleetWise.

Pour utiliser AWS l'IoT FleetWise via votre VPC, vous devez vous connecter depuis une instance située à l'intérieur du VPC ou connecter votre réseau privé à votre VPC à l'aide d'un (VPN) ou. AWS Virtual Private Network AWS Direct Connect Pour obtenir des informations sur Amazon VPN, consultez la rubrique Connexions VPN du Guide de l'utilisateur Amazon Virtual Private Cloud. Pour plus d'informations AWS Direct Connect, voir Création d'une connexion dans le Guide de AWS Direct Connect l'utilisateur.

Vous pouvez créer un point de terminaison VPC d'interface pour vous connecter à l' AWS IoT à l'aide FleetWise de la AWS console ou des commandes AWS Command Line Interface (AWS CLI). Pour plus d'informations, consultez Création d'un point de terminaison d'interface.

Après avoir créé un point de terminaison VPC d'interface, si vous activez des noms d'hôte DNS privés pour le point de terminaison, le point de terminaison AWS IoT par défaut est remplacé par votre point de FleetWise terminaison VPC. Le nom de service par défaut, point de terminaison pour l' AWS IoT, FleetWise est au format suivant.

```
iotfleetwise.Region.amazonaws.com
```

Si vous n'activez pas les noms d'hôte DNS privés, Amazon VPC fournit un nom de point de terminaison DNS que vous pouvez utiliser au format suivant.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

Pour de plus amples informations, consultez Points de terminaison VPC (AWS PrivateLink) dans le Guide de l'utilisateur Amazon VPC.

AWS L' FleetWise IoT permet d'appeler toutes ses actions d'API au sein de votre VPC.

Vous pouvez attacher des politiques de point de terminaison de VPC au point de terminaison d'un VPC pour contrôler l'accès des principaux IAM. Vous pouvez également associer des groupes de sécurité à un point de terminaison VPC pour contrôler l'accès entrant et sortant en fonction de l'origine et de la destination du trafic réseau, comme une plage d'adresses IP. Pour plus d'informations, veuillez consulter Contrôler l'accès aux services avec les points de terminaison d'un VPC.

> **Note**
>
> AWS L'IoT FleetWise prend en charge tous les points de terminaison VPC en mode double pile. Pour plus d'informations sur les points de terminaison de service, consultez la section FleetWise Points de terminaison et quotas AWS IoT.

Création d'une politique de point de terminaison VPC pour l'IoT AWS FleetWise

Vous pouvez créer une politique pour les points de terminaison Amazon VPC pour l' AWS IoT FleetWise afin de spécifier les éléments suivants :

- Principal qui peut ou ne peut pas effectuer des actions
- Les actions qui peuvent ou ne peuvent pas être effectuées

Pour plus d'informations, consultez Contrôle de l'accès aux services avec points de terminaison d'un VPC dans le Guide de l'utilisateur Amazon VPC.

Example — Politique de point de terminaison VPC interdisant tout accès depuis un compte spécifié AWS

La politique de point de terminaison VPC suivante refuse au AWS compte *123456789012* tous les appels d'API utilisant le point de terminaison.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
```

```
        }
    ]
}
```

Example – Politique du point de terminaison d'un VPC pour autoriser l'accès VPC uniquement à un principal (utilisateur) IAM spécifié

La politique de point de terminaison VPC suivante autorise l'accès complet uniquement à un utilisateur inscrit *lijuan* dans AWS le compte. *123456789012* Il refuse à tous les autres principaux IAM l'accès au point de terminaison.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
                ]
            }
        }]
}
```

Example — Politique de point de terminaison VPC pour les actions IoT AWS FleetWise

Voici un exemple de politique relative aux terminaux pour AWS l'IoT FleetWise. Lorsqu'elle est attachée à un point de terminaison, cette politique accorde l'accès aux FleetWise actions AWS IoT répertoriées pour l'utilisateur IAM *fleetWise* dans le Compte AWS *123456789012*.

```
{
    "Statement": [
        {
             "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/fleetWise"
                },
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "iotfleetwise:ListFleets",
```

```
                "iotfleetwise:ListCampaigns",
                "iotfleetwise:CreateVehicle",
            ]
           }
    ]
}
```

Analyse de configuration et de vulnérabilité dans AWS l'IoT FleetWise

Les environnements 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 appareils complexe et source d'erreurs. De plus, étant donné que les appareils sont souvent limités en termes de puissance de calcul, de mémoire et de capacités de stockage, l'utilisation du chiffrement et d'autres formes de sécurité sur les appareils est limitée. Les appareils utilisent souvent des logiciels aux vulnérabilités connues. Ces facteurs font des appareils IoT, notamment les véhicules collectant des données pour l' AWS IoT FleetWise, une cible attrayante pour les pirates informatiques et compliquent leur sécurisation continue.

La configuration et les contrôles informatiques sont une responsabilité partagée entre vous AWS et vous, notre client. Pour plus d'informations, consultez le modèle de responsabilité AWS partagée.

Bonnes pratiques de sécurité pour AWS l'IoT FleetWise

AWS L'IoT FleetWise fournit un certain nombre de fonctionnalités de sécurité à prendre en compte lors de l'élaboration et de la mise en œuvre de vos propres politiques de sécurité. Les bonnes pratiques suivantes doivent être considérées comme des instructions générales et ne représentent pas une solution de sécurité complète. Étant donné que ces bonnes pratiques peuvent ne pas être appropriées ou suffisantes pour votre environnement, considérez-les comme des remarques utiles plutôt que comme des recommandations.

Pour en savoir plus sur la sécurité, AWS IoT consultez la section Bonnes pratiques en matière de sécurité AWS IoT Core dans le guide du AWS IoT développeur

Accorder le moins d'autorisations possibles

Respectez le principe du moindre privilège en utilisant l'ensemble minimal d'autorisations dans les rôles IAM. Limitez l'utilisation du `*` caractère générique pour les `Resource` propriétés `Action` et

dans vos politiques IAM. Au lieu de cela, déclarez un ensemble fini d'actions et de ressources lorsque cela est possible. Pour plus d'informations sur le moindre privilège et les autres bonnes pratiques en matière de stratégie, veuillez consulter the section called “Bonnes pratiques en matière de politiques”.

Ne journalisez pas les informations sensibles

Vous devez empêcher la journalisation des informations d'identification et d'autres informations personnelles identifiables (PII). Nous vous recommandons de mettre en œuvre les mesures de protection suivantes :

- N'utilisez pas d'informations sensibles dans les noms des appareils.
- N'utilisez pas d'informations sensibles dans les noms et IDs les FleetWise ressources AWS IoT, par exemple dans les noms de campagnes, de manifestes de décodeurs, de modèles de véhicules et de catalogues de signaux, ou dans les noms IDs de véhicules et de flottes.

AWS CloudTrail À utiliser pour afficher l'historique des appels d'API

Vous pouvez consulter l'historique des appels d' FleetWise API AWS IoT effectués sur votre compte à des fins d'analyse de sécurité et de résolution des problèmes opérationnels. Pour recevoir l'historique des appels d' FleetWise API AWS IoT effectués sur votre compte, il vous suffit CloudTrail d'activer le AWS Management Console. Pour de plus amples informations, veuillez consulter the section called “CloudTrail journaux”.

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. Les horloges de l'appareil peuvent se décaler au fil du temps ou les batteries peuvent se décharger.

Pour de plus amples informations, veuillez consulter les bonnes pratiques décrites dans la section Veiller à la synchronisation de l'horloge de votre appareil dans le Manuel du développeur AWS IoT Core .

Surveillez AWS l'IoT FleetWise

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de AWS l'IoT FleetWise et de vos autres AWS solutions. AWS fournit les outils de surveillance suivants pour surveiller AWS l'IoT FleetWise, signaler un problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation du processeur ou d'autres indicateurs de vos EC2 instances Amazon et lancer automatiquement de nouvelles instances en cas de besoin. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs peut être utilisé pour surveiller, stocker et accéder à vos fichiers journaux à partir d' EC2 instances Amazon et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- AWS CloudTrail capture les appels d'API et les événements associés effectués par votre Compte AWS ou au nom de ce dernier. Il envoie ensuite les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, l'adresse IP source à partir de laquelle les appels ont été effectués et la date des appels. Pour plus d'informations, consultez le AWS CloudTrail Guide de l'utilisateur .

Surveillez AWS l'IoT FleetWise avec Amazon CloudWatch

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Les CloudWatch métriques Amazon sont un moyen de surveiller vos AWS ressources et leurs performances. AWS L'IoT FleetWise envoie des métriques à CloudWatch. Vous pouvez utiliser l'API AWS Management Console AWS CLI, le ou une API pour répertorier les métriques auxquelles l' AWS IoT FleetWise envoie des données CloudWatch. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.

> **Important**
>
> Vous devez configurer les paramètres pour que AWS l'IoT FleetWise puisse envoyer des métriques à CloudWatch. Pour de plus amples informations, veuillez consulter Configurez vos FleetWise paramètres AWS IoT.

L'espace de noms `AWS/IoTFleetWise` inclut les métriques suivantes.

Métriques du signal

Métrique	Description
IllegalMessageFromEdge	Un message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise ne correspondait pas au format requis. Unités : nombre Dimensions : Aucune Statistiques valides : somme
MessageThrottled	Un message envoyé par le véhicule à l' AWS IoT FleetWise a été limité. Cela est dû au fait que vous avez dépassé les limites de service pour ce compte dans la région actuelle. Unités : nombre Dimensions : Aucune Statistiques valides : somme

Métrique	Description
ModelingError	Un message envoyé par le véhicule et reçu par l' AWS IoT FleetWise contient des signaux qui ne sont pas validés par rapport au modèle du véhicule. Unités : nombre Dimensions : ModelName, StateTemplateName (facultatif), SignalCatalogName (facultatif)
DecodingError	Un message envoyé depuis le véhicule et reçu par l' AWS IoT FleetWise contient des signaux qui ne sont pas décodés par rapport au manifeste du décodeur du véhicule. Unités : nombre Dimensions : DecoderName Statistiques valides : somme
MessageSizeLimitExceeded	Un message envoyé par le véhicule à l' AWS IoT FleetWise a été supprimé. Cela est dû au fait que vous avez dépassé la taille maximale d'un service de messagerie pour ce compte dans la région actuelle. Unités : nombre Dimensions : Aucune Statistiques valides : somme

Métriques relatives aux véhicules

Métrique	Description
VehicleNotFound	Message reçu par AWS l'IoT FleetWise, dont le véhicule est inconnu. Unités : nombre Dimensions : Aucune Statistiques valides : somme

Métriques de campagne

Métrique	Description
CampaignInvalid	Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, lorsque la campagne n'est pas valide. Unités : nombre Dimensions : CampaignName Statistiques valides : somme
CampaignNotFound	Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, dont la campagne est inconnue. Unités : nombre Dimensions : CampaignName Statistiques valides : somme

Métriques du modèle d'état

Métrique	Description
NoStateTemplatesAssociated	Message envoyé depuis le véhicule et reçu par l' AWS IoT FleetWise, dans lequel aucun modèle d'état n'est associé au véhicule. Unités : nombre Statistiques valides : somme

Indicateurs de destination des données de campagne

Métrique	Description
TimestreamWriteError	AWS L'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans le tableau Amazon Timestream. Unités : nombre Dimensions : DatabaseName, TableName Statistiques valides : somme
S3 WriteError	AWS L'IoT n'a pas FleetWise pu écrire de message depuis le véhicule vers le compartim ent Amazon Simple Storage Service (Amazon S3). Unités : nombre Dimensions : BucketName Statistiques valides : somme
S3 ReadError	AWS L'IoT n' FleetWise a pas pu lire une clé d'objet depuis le véhicule dans le compartiment Amazon Simple Storage Service (Amazon S3).

Métrique	Description
	Unités : nombre Dimensions : BucketName Statistiques valides : somme

Indicateurs AWS KMS clés gérés par le client

Métrique	Description
KMSKeyAccessDenied	AWS L'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans la table Timestream ou dans le compartiment Amazon S3 en raison d'une erreur de refus d'accès par AWS KMS clé. Unités : nombre Dimensions : KMSKey Id Statistiques valides : somme

Surveillez AWS l'IoT FleetWise avec Amazon CloudWatch Logs

> **Important**
>
> L'accès à certaines FleetWise fonctionnalités de AWS l'IoT est actuellement restreint. Pour de plus amples informations, veuillez consulter AWS Disponibilité des régions et des fonctionnalités dans AWS l'IoT FleetWise.

Amazon CloudWatch Logs surveille les événements qui se produisent dans vos ressources et vous alerte en cas de problème. Si vous recevez une alerte, vous pouvez accéder aux fichiers journaux pour obtenir des informations sur l'événement en question. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.

Afficher les FleetWise journaux AWS IoT dans la CloudWatch console

> **Important**
>
> Avant de voir le groupe de FleetWise journaux AWS IoT dans la CloudWatch console, assurez-vous que ce qui suit est vrai :
>
> - Vous avez activé la connexion à AWS l'IoT FleetWise. Pour plus d'informations sur la journalisation, consultez Configurer la FleetWise journalisation de AWS l'IoT.
> - Il existe déjà des entrées de journal écrites par AWS IoT les opérations.

Pour consulter vos FleetWise journaux AWS IoT dans la CloudWatch console

1. Ouvrez la CloudWatch console.
2. Dans le volet de navigation, choisissez Logs, Log groups.
3. Choisissez le groupe de journaux .
4. Choisissez Rechercher un groupe de journaux. Vous verrez la liste complète des événements de journal générés pour votre compte.
5. Cliquez sur l'icône d'extension pour consulter un flux individuel et rechercher tous les journaux dont le niveau de journalisation est de`ERROR`.

   Vous pouvez également saisir une requête dans le champ de recherche Filtrer les événements. Par exemple, vous pouvez essayer la requête suivante :

   ```
   { $.logLevel = "ERROR" }
   ```

Pour plus d'informations sur la création d'expressions de filtre, consultez la section Syntaxe des filtres et des modèles dans le guide de l'utilisateur Amazon CloudWatch Logs.

Example entrée dans le journal

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
```

```
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

Types d'événements de signal

Type d’événement	Description
ERREUR_DE MODÉLISATION	Un message envoyé par le véhicule et reçu par l' AWS IoT FleetWise contient des signaux qui ne sont pas validés par rapport au modèle du véhicule. Attributs : VehicleName, CampaignName (facultatif) signalCatalogName, SignalID (facultatif), SignalValue (facultatif), signalVal ueRange Min (facultatif), signalValueRange Max (facultatif), (facultatif), SignalIds, modelManifestName stateTemplateName
MESSAGE_DE_DE_DE_BORD ILLÉGAL	Un message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise ne correspondait pas au format requis. Attributs : nom du véhicule, nom de la campagne, signalCatalogName
ERREUR_DE DÉCODAGE	Un message envoyé depuis le véhicule et reçu par l' AWS IoT FleetWise contient des signaux qui ne sont pas décodés par rapport au manifeste du décodeur du véhicule. Attributs : CampaignName, signalCat alogName, decoderManifestName, (facultatif) SignalName, (facultatif) S3URI
MESSAGE_THROTTLED	Un message envoyé par le véhicule à l' AWS IoT FleetWise a été limité. Cela est dû au fait

Type d’événement	Description
	que vous avez dépassé les limites de service pour ce compte dans la région actuelle. Attributs : AccountID, VehicleName, message, EventType, LogLevel, horodatage
LIMITE DE TAILLE DU MESSAGE DÉPASSÉE	Un message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise dépasse la taille maximale d'une limite de service de messageri e. Attributs : AccountID, VehicleName

Types d'événements liés aux véhicules

Type d’événement	Description
VÉHICULE INTROUVABLE	Un message reçu par AWS l'IoT FleetWise, indiquant que le véhicule était inconnu. Attributs : VehicleName, CampaignName (facultatif), stateTemplateName (facultatif)

Types d'événements liés à la campagne

Type d’événement	Description
CAMPAGNE_INTROUVABLE	Un message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, alors que la campagne était inconnue. Attributs : VehicleName (facultatif), CampaignN ame
CAMPAGNE_INVALID	Message envoyé depuis le véhicule et reçu par l' AWS IoT FleetWise, pour lequel la campagne n'était pas valide.

Type d’événement	Description
	Attributs : VehicleName (facultatif), CampaignName

Types d'événements de destination des données de campagne

Type d’événement	Description
TIMESTREAM_WRITE_ERROR	AWS L'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans le tableau Amazon Timestream. Attributs : nom du véhicule, nom de la campagne,, timestreamDatabaseName timestreamTableName
S3_WRITE_ERROR	AWS L'IoT n'a pas FleetWise pu écrire de message depuis le véhicule vers le compartiment Amazon Simple Storage Service (Amazon S3). Attributs : CampaignName, DestinationName
S3_READ_ERROR	AWS L'IoT n' FleetWise a pas pu lire une clé d'objet depuis le véhicule dans le compartiment Amazon Simple Storage Service (Amazon S3). Attributs : CampaignName, DestinationName

Types d'événements du modèle d'état

Type d’événement	Description
MODÈLE D'ÉTAT NON TROUVÉ	Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, dont le modèle d'état était inconnu.

Type d’événement	Description
	Attributs : VehicleName (facultatif), stateTemplateName

AWS KMS Principaux types d'événements gérés par le client

Type d’événement	Description
KMS_KEY_ACCESS_DENIED	AWS L'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans la table Timestream ou dans le compartiment Amazon S3 en raison d'une erreur de refus d'accès par AWS KMS clé. Attributs : kmsKeyId (facultatif), ResourceArn (facultatif)

Attributs

Toutes les entrées CloudWatch des journaux incluent les attributs suivants :

accountId

Votre Compte AWS identifiant.

eventType

Le type d'événement pour lequel le journal a été créé. La valeur du type d'événement dépend de l'événement qui a généré l'entrée de journal. Chaque description d'entrée de journal inclut la valeur de `eventType` pour cette entrée de journal.

logLevel

Le niveau de journalisation utilisé. Pour plus d'informations, consultez la section Niveaux de journalisation dans le guide du AWS IoT Core développeur.

message

Contient des informations spécifiques sur le journal.

timestamp

Horodatage en millisecondes de l'époque à laquelle AWS l'IoT a traité le journal. FleetWise

Attributs facultatifs

CloudWatch Les entrées des journaux incluent éventuellement ces attributs, en fonction des éléments `eventType` suivants :

decoderManifestName

Nom du manifeste du décodeur qui contient le signal.

Nom de la destination

Nom de la destination pour les données du véhicule. Par exemple, le nom du compartiment Amazon S3.

Nom de la campagne

Nom de la campagne.

signalCatalogName

Nom du catalogue de signaux qui contient le signal.

Identifiant du signal

ID du signal d'erreur.

ID de signal

Liste des signaux d'erreur IDs.

Nom du signal

Le nom du signal.

signalTimestampEpochMme

Horodatage du signal d'erreur.

Valeur du signal

La valeur du signal d'erreur.

signalValueRangeMax

La portée maximale du signal d'erreur.

signalValueRangeMinimum

La plage minimale du signal d'erreur.

S3uri

L'identifiant unique Amazon S3 d'un fichier Amazon Ion issu d'un message relatif au véhicule.

timestreamDatabaseName

Nom de la base de données Timestream.

timestreamTableName

Nom de la table Timestream.

Nom du véhicule

Le nom du véhicule.

Configurer la FleetWise journalisation de AWS l'IoT

Vous pouvez envoyer les données de votre FleetWise journal AWS IoT à un groupe de CloudWatch journaux. CloudWatch Les journaux offrent de la visibilité au cas où AWS l'IoT FleetWise ne parviendrait pas à traiter les messages provenant des véhicules. Cela peut être dû, par exemple, à une configuration défectueuse ou à d'autres erreurs du client. Vous êtes informé de toute erreur afin que vous puissiez identifier et atténuer les problèmes.

Avant de pouvoir envoyer des journaux à CloudWatch, vous devez créer un groupe de CloudWatch journaux. Configurez le groupe de journaux avec le même compte et dans la même région que ceux que vous avez utilisés avec AWS l'IoT FleetWise. Lorsque vous activez la connexion à AWS l'IoT FleetWise, indiquez le nom du groupe de journaux. Une fois la journalisation activée, AWS l'IoT FleetWise envoie les journaux au groupe de CloudWatch journaux sous forme de flux de journaux.

Vous pouvez consulter les données du journal envoyées depuis AWS l'IoT FleetWise dans la CloudWatch console. Pour plus d'informations sur la configuration d'un groupe de CloudWatch journaux et l'affichage des données du journal, consultez la section Utilisation des groupes de journaux.

Autorisations permettant de publier des journaux sur CloudWatch

La configuration de la journalisation pour un groupe de CloudWatch journaux nécessite les paramètres d'autorisation décrits dans cette section. Pour plus d'informations sur la gestion des autorisations, consultez la section Gestion de l'accès aux AWS ressources dans le Guide de l'utilisateur IAM.

Avec ces autorisations, vous pouvez modifier la configuration de journalisation, configurer la livraison des CloudWatch journaux et récupérer des informations sur votre groupe de journaux.

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

            "iotfleetwise:PutLoggingOptions",
            "iotfleetwise:GetLoggingOptions"
         ],
         "Resource":[
            "*"
         ],
         "Effect":"Allow",
         "Sid":"IoTFleetwiseLoggingOptionsAPI"
      }
      {
         "Sid":"IoTFleetwiseLoggingCWL",
         "Action":[
            "logs:CreateLogDelivery",
            "logs:GetLogDelivery",
            "logs:UpdateLogDelivery",
            "logs:DeleteLogDelivery",
            "logs:ListLogDeliveries",
            "logs:PutResourcePolicy",
            "logs:DescribeResourcePolicies",
            "logs:DescribeLogGroups"
         ],
         "Resource":[
            "*"
         ],
         "Effect":"Allow"
      }
   ]
```

```
}
```

Lorsque des actions sont autorisées sur toutes les AWS ressources, cela est indiqué dans la politique avec un `"Resource"` paramètre de`"*"`. Cela signifie que les actions sont autorisées sur toutes les AWS ressources prises en charge par chaque action.

Configurer la journalisation dans AWS l'IoT FleetWise (console)

Cette section décrit comment utiliser la FleetWise console AWS IoT pour configurer la journalisation.

Pour utiliser la FleetWise console AWS IoT pour configurer la journalisation

1. Ouvrez la FleetWiseconsole AWS IoT.
2. Dans le volet de gauche, choisissez Settings (Paramètres).
3. Dans la section Journalisation de la page Paramètres, choisissez Modifier.
4. Dans la section de CloudWatch journalisation, entrez le groupe de journaux.
5. Pour enregistrer vos modifications, choisissez Soumettre.

Après avoir activé la journalisation, vous pouvez consulter les données de votre journal dans la CloudWatch console.

Configurer la journalisation par défaut dans AWS l'IoT FleetWise (CLI)

Cette section décrit comment configurer la journalisation pour AWS l'IoT à FleetWise l'aide de la CLI.

Vous pouvez également exécuter cette procédure avec l'API en utilisant les méthodes de l' AWS API qui correspondent aux commandes CLI présentées ici. Vous pouvez utiliser l'opération GetLoggingOptionsAPI pour récupérer la configuration actuelle et l'opération PutLoggingOptionsAPI pour modifier la configuration.

Pour utiliser la CLI afin de configurer la journalisation pour AWS l'IoT FleetWise

1. Pour obtenir les options de journalisation de votre compte, utilisez la get-logging-options commande.

   ```
   aws iotfleetwise get-logging-options
   ```

2. Pour activer la journalisation, utilisez la put-logging-options commande.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
 logType=ERROR,logGroupName=MyLogGroup
```

où :

logType

Type de journal pour envoyer des données à CloudWatch Logs. Pour désactiver la journalisation, remplacez la valeur par`OFF`.

logGroupName

Le groupe CloudWatch Logs auquel l'opération envoie des données. Assurez-vous de créer le nom du groupe de journaux avant d'activer la journalisation pour AWS l'IoT FleetWise.

Après avoir activé la journalisation, voir Rechercher des entrées de journal à l'aide de la AWS CLI.

Enregistrez les appels d' FleetWise API AWS IoT à l'aide de AWS CloudTrail

AWS L'IoT FleetWise est intégré à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans AWS l'IoT FleetWise. CloudTrail capture tous les appels d'API pour AWS l'IoT FleetWise sous forme d'événements. Les appels capturés incluent des appels provenant de la FleetWise console AWS IoT et des appels de code vers les opérations de FleetWise l'API AWS IoT. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrailévénements vers un compartiment Amazon S3, y compris des événements pour AWS l'IoT FleetWise. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les événements les plus récents dans la CloudTrail console dans Historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à l' AWS IoT FleetWise, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

AWS FleetWise Informations sur l'IoT dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans AWS l'IoT FleetWise, cette activité est enregistrée dans un CloudTrail événement avec d'autres

événements de AWS service dans l'historique des événements. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS . Pour plus d'informations, consultez la section Affichage des événements avec l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris les événements liés à AWS l'IoT FleetWise, créez un parcours. Un suivi permet CloudTrail de fournir des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions AWS. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception de fichiers CloudTrail journaux provenant de plusieurs régions
- Réception de fichiers CloudTrail journaux provenant de plusieurs comptes

Toutes les FleetWise actions relatives à l' AWS IoT sont enregistrées CloudTrail et documentées dans la référence des FleetWise API AWS IoT. Par exemple, les appels aux `CreateCampaignAssociateVehicleFleet`, et `GetModelManifest` les actions génèrent des entrées dans les fichiers CloudTrail journaux.

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 des informations d'identification d'utilisateur root ou IAM.
- Si la demande a été effectuée avec des informations d'identification de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail .

Comprendre les entrées des fichiers FleetWise journaux de l' AWS IoT

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des appels d'API publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'*AssociateVehicleFleet*opération.


```
{
      "eventVersion": "1.05",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:assumed-role/NikkiWolf",
        "accountId": "111122223333",
        "accessKeyId": "access-key-id",
        "userName": "NikkiWolf"
      },
      "eventTime": "2021-11-30T09:56:35Z",
      "eventSource": "iotfleetwise.amazonaws.com",
      "eventName": "AssociateVehicleFleet",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.21",
      "userAgent": "aws-cli/2.3.2 Python/3.8.8 Darwin/18.7.0 botocore/2.0.0",
      "requestParameters": {
          "fleetId": "f1234567890",
          "vehicleId": "v0213456789"
       },
      "responseElements": {
      },
      "requestID": "9f861429-11e3-11e8-9eea-0781b5c0ac21",
      "eventID": "17385819-4927-41ee-a6a5-29ml0br812v4",
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
```

Historique du document relatif au guide du FleetWise développeur de AWS l'IoT

Le tableau suivant décrit les versions de documentation relatives à AWS l'IoT FleetWise.

Modification	Description	Date
Expansion de région	AWS L'IoT FleetWise est désormais disponible dans la région Asie-Pacifique (Mumbai) (accès sécurisé uniquement).	21 novembre 2024
Disponibilité générale restreinte des nouvelles fonctionnalités	AWS L'IoT prend FleetWise désormais en charge l'accès restreint pour les campagnes visant à stocker et à transférer des données, à configurer un sujet MQTT comme destination de données et à collecter des données de code de diagnostic. Il prend également désormais en charge l'accès sécurisé pour la collecte de données indépendante du réseau à l'aide d'interfaces de décodage personnalisées, de commandes de configuration et de surveillance du dernier état connu des véhicules.	21 novembre 2024
Envoyer des données de campagne vers un sujet MQTT	AWS L'IoT permet FleetWise désormais d'envoyer les données collectées pendant les campagnes vers un sujet MQTT que vous spécifiez,	1 mai 2024

	en plus de la possibilité de stocker les données dans Amazon S3 ou Amazon Timestream.	
Aperçu des données du système Vision	Vous pouvez utiliser l'aperçu des données des systèmes de vision issues de AWS l'IoT FleetWise pour collecter et organiser les données provenant des systèmes de vision des véhicules, notamment des caméras, des radars et des lidars. Il permet de synchroniser automatiquement dans le cloud les données du système de vision structurées et non structurées, les métadonné es (identifiant d'événement, campagne, véhicule) et le capteur standard (données de télémétrie).	26 novembre 2023

AWS KMS clés gérées par le client	AWS L'IoT prend FleetWise désormais en charge les clés gérées par les AWS KMS clients. Vous pouvez utiliser la clé KMS pour chiffrer les données côté serveur relatives aux FleetWise ressource s AWS IoT (catalogue de signaux, modèle de véhicule, manifeste du décodeur, véhicules et configurations de campagnes de collecte de données) stockées dans. AWS Cloud	16 octobre 2023
Stockage d'objets dans Amazon S3	AWS L'IoT prend FleetWise désormais en charge le stockage de données à l'aide d'Amazon Simple Storage Service (Amazon S3). Vous pouvez stocker les données collectées pendant les campagnes dans Amazon S3, en plus d'Amazon Timestream.	1er juin 2023
Disponibilité générale	Il s'agit de la version publique de l' AWS IoT FleetWise.	27 septembre 2022
Première version	Il s'agit de la version prélimina ire du guide du FleetWise développeur de AWS l'IoT.	30 novembre 2021

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.