aws

Guide du développeur

# AWS IoT FleetWise

# AWS IoT FleetWise: Guide du développeur

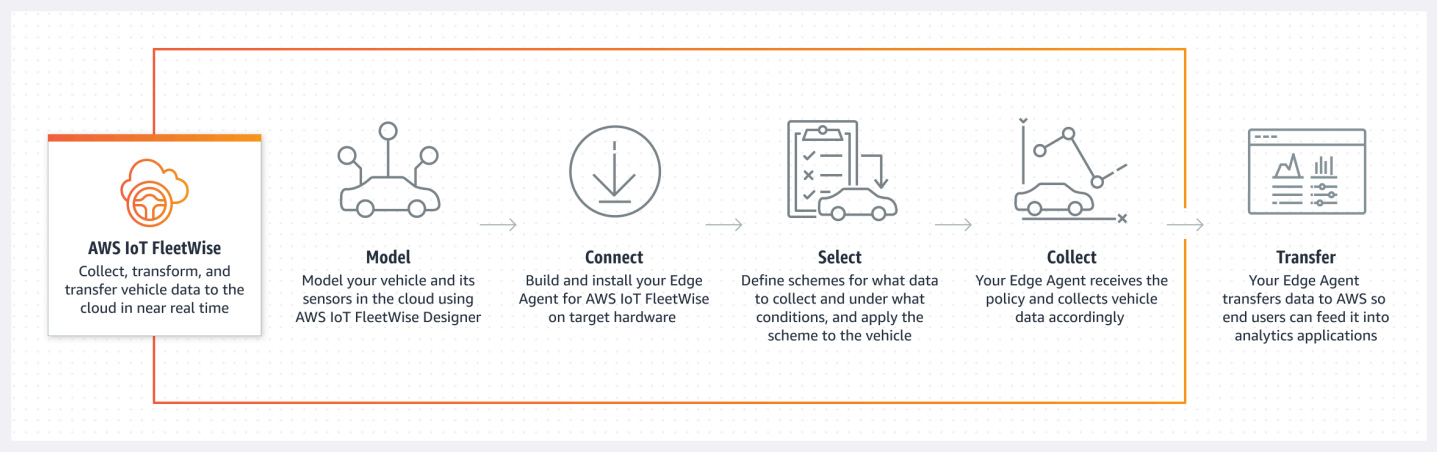
# Table of Contents

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

AWSL'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 afin de contrôler les données du véhicule à collecter et le moment où ces données doivent être transférées 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'AWSIoT FleetWise.




Rubriques

- Comment FleetWise fonctionne AWS l'IoT
- Services connexes

# Avantages

Les principaux avantages de l'AWSIoT 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.

> **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'AWSIoT 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.

## 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 :

- Comment FleetWise fonctionne AWS l'IoT
- Configuration de AWS l'IoT FleetWise
- Démonstration du logiciel Edge Agent
- Ingestion de données 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'AWSIoT. Pour en savoir plus sur le protocole de messagerie MQTT, consultez MQTT dans le guide du AWS IoT Coredéveloppeur.

# Comment FleetWise fonctionne AWS l'IoT

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 Configuration de AWS l'IoT FleetWise section pour savoir comment configurer AWS l'IoT FleetWise.

Rubriques



## Concepts clés

AWSL'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'AWSIoT 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'AWSIoT 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 en savoir plus, consultez Création et gestion de catalogues de signaux.

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'AWSIoT 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 en savoir plus, consultez Création et gestion de catalogues de signaux.

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 en savoir plus, consultez Création et gestion de modèles de véhicules.

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 en savoir plus, consultez Création et gestion des manifestes du décodeur.

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 (ECU). Les ECU peuvent 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 les calculateurs. Il fournit un certain nombre de codes de diagnostic standard (DTC) 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**
>
> AWSL'IoT FleetWise prend en charge le middleware ROS 2 pour les données du système de vision.

Signal du décodeur

Fournit des informations de décodage détaillées pour un signal spécifique. Chaque signal spécifié dans le modèle du véhicule doit être associé à un signal de décodeur. 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 les signaux du décodeur OBD.

Le manifeste du décodeur doit contenir les signaux du décodeur de messages s'il contient également des interfaces intergicielles pour véhicules.

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.

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.

## Caractéristiques de l'AWSIoT FleetWise

Les principales caractéristiques de l'AWSIoT 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.

# Services connexes

AWSL'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. 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.

# Configuration de AWS l'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

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

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

   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 en matière de sécurité consiste à attribuer un accès administratif à un utilisateur et à n'utiliser que l'utilisateur root pour effectuer les tâches nécessitant un accès utilisateur root.

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

# Création d'un utilisateur doté d'un accès administratif

Une fois que vous vous êtes inscrit à un utilisateur administratif Compte AWS, que vous Utilisateur racine d'un compte AWS l'avez sécurisé AWS IAM Identity Center, que vous l'avez activé et que vous en avez créé un, afin de ne pas utiliser l'utilisateur root pour les tâches quotidiennes.

Sécurisez votre Utilisateur racine d'un compte AWS

1. Connectez-vous en AWS Management 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, accordez 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.

Connectez-vous en tant qu'utilisateur disposant 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.

Attribuer l'accès à des utilisateurs supplémentaires

1. Dans IAM Identity Center, créez un ensemble d'autorisations conforme aux meilleures pratiques en matière d'application des autorisations du moindre privilège.

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

2. Affectez des utilisateurs à un groupe, puis attribuez un accès d'authentification unique au groupe.

   Pour obtenir des instructions, voir Ajouter des groupes dans le guide de AWS IAM Identity Center l'utilisateur.

> **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 plus d'informations, consultez 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. Accédez à la FleetWiseconsole 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 véhicule (console).

# Configuration des paramètres

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, consultezSurveillance de 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.

## Configurer les paramètres (console)

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

1. Accédez à la FleetWiseconsole 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 plus d’informations, consultez Gestion des clés.

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 plus d'informations, consultez 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 est correctement enregistrée.
- `REGISTRATION_PENDING`— AWS FleetWise L'IoT traite 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.

# Débuter avec AWS l'IoT FleetWise

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

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

- Ingestion de données dans le cloud
- Modélisation de véhicules
- Créez, approvisionnez et gérez des véhicules
- Créez et gérez des flottes
- Collectez et transférez des données grâce aux campagnes

## Prérequis

Vous devez avoir un Compte AWS pour commencer à utiliser l' AWS IoT FleetWise. Si vous n'en avez pas, veuillez consulter Configuration de AWS l'IoT FleetWise.

Utilisez une région où AWS l'IoT FleetWise est disponible. Pour plus d'informations, consultez la section FleetWise Points de terminaison et quotas AWS IoT. Vous pouvez utiliser le sélecteur de région dans le AWS Management Console pour passer à l'une de ces régions.

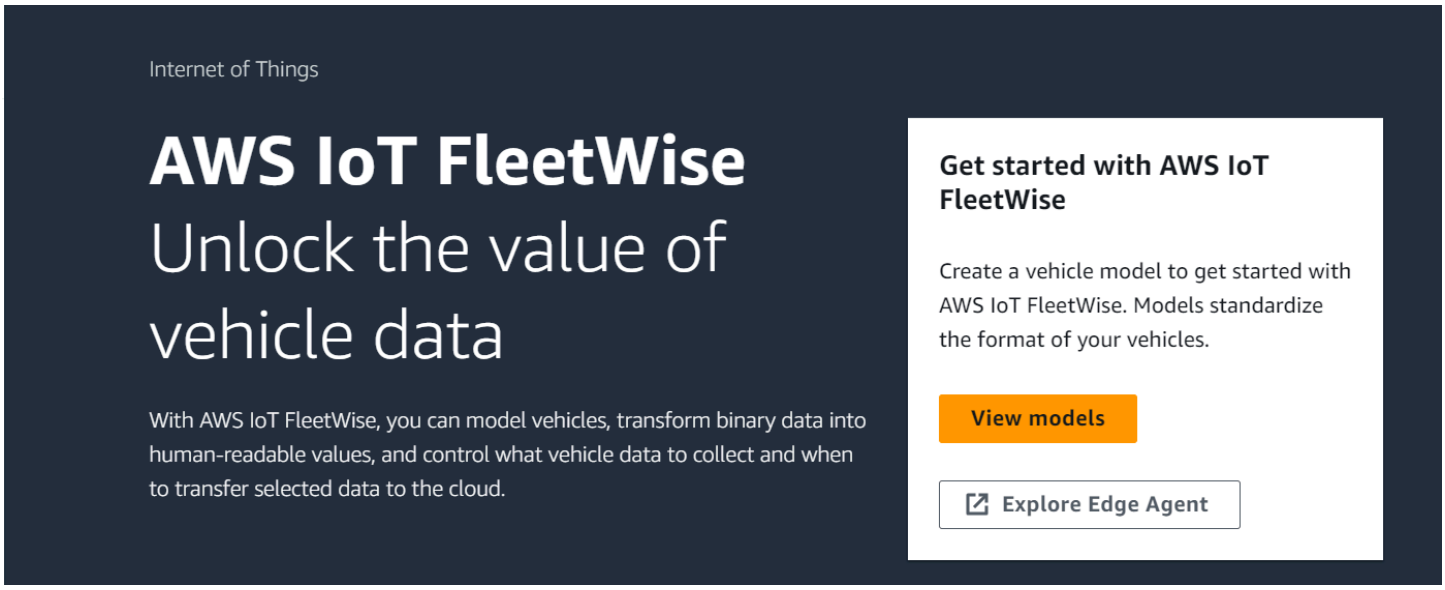
## Démonstration du logiciel Edge Agent

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 graviton Amazon EC2 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 plus d'informations sur la démo de démarrage rapide, procédez comme suit pour télécharger le guide du développeur du logiciel Edge Agent.

Pour télécharger la démo de démarrage rapide

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.




# Tutoriel : Débuter avec AWS l'IoT FleetWise (console)

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, consultezComment FleetWise fonctionne AWS l'IoT.

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.

Rubriques



## 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 AWS Account Management 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).
- 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.

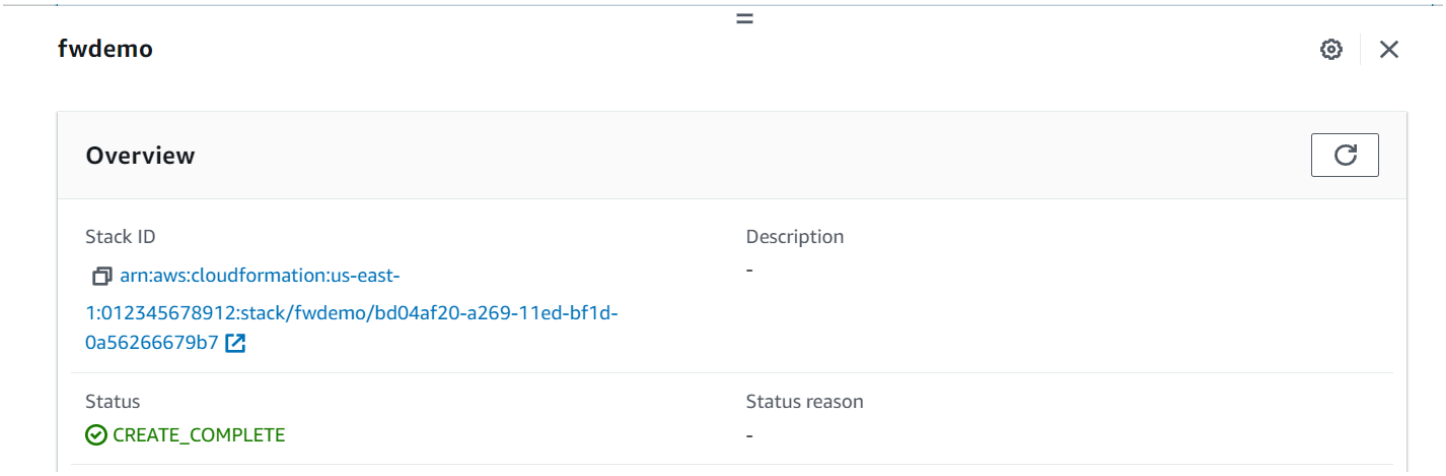
# Étape 1 : configurer le logiciel Edge Agent pour AWS l'IoT FleetWise

> **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. IoT CoreRegion - Vous pouvez spécifier la région dans laquelle l' AWS IoT objet est créé en mettant à jour le CoreRegion paramètre IoT. 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 Régions et 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.

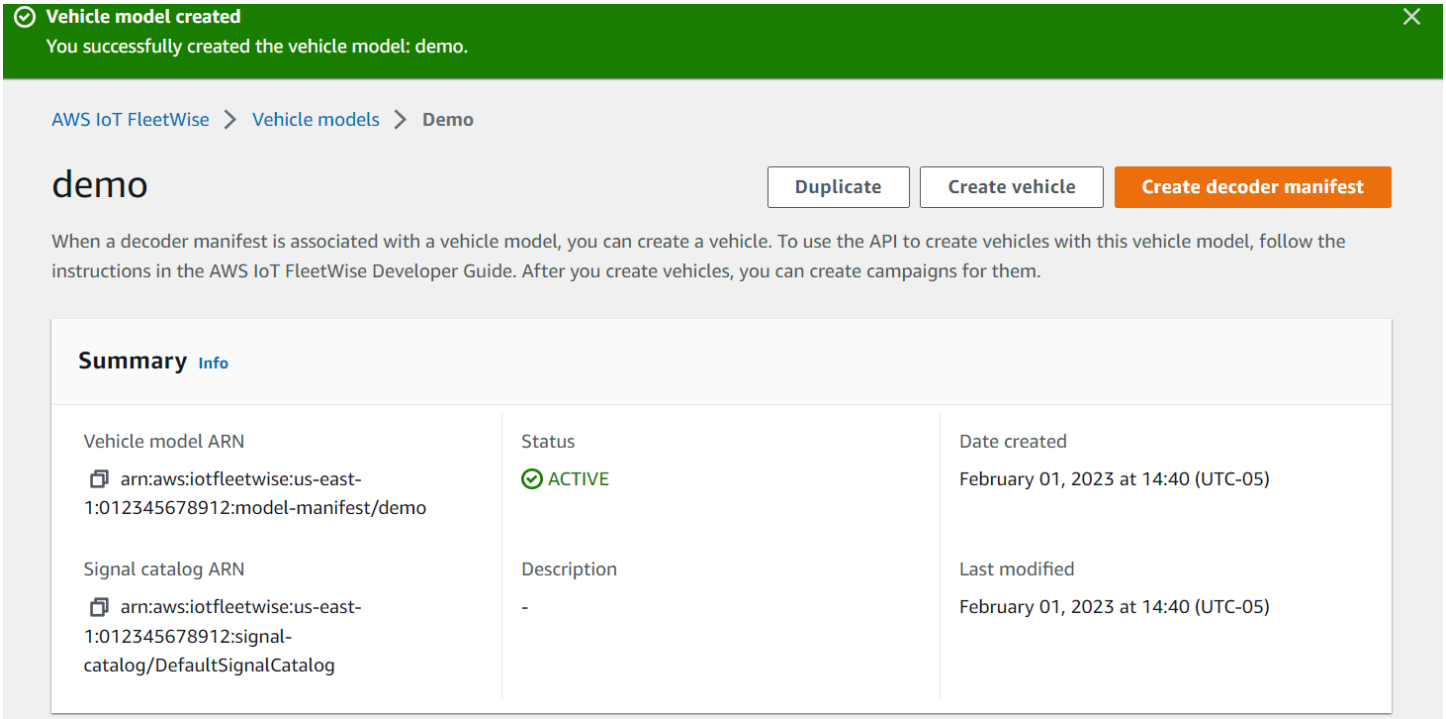
   c. Unité - (Facultatif) Entrez une valeur unitaire, telle que le kilomètre ou le Celsius.

   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 : Création d'un manifeste de 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 plus d'informations, consultez Création d'un manifeste de décodeur ()AWS CLI.

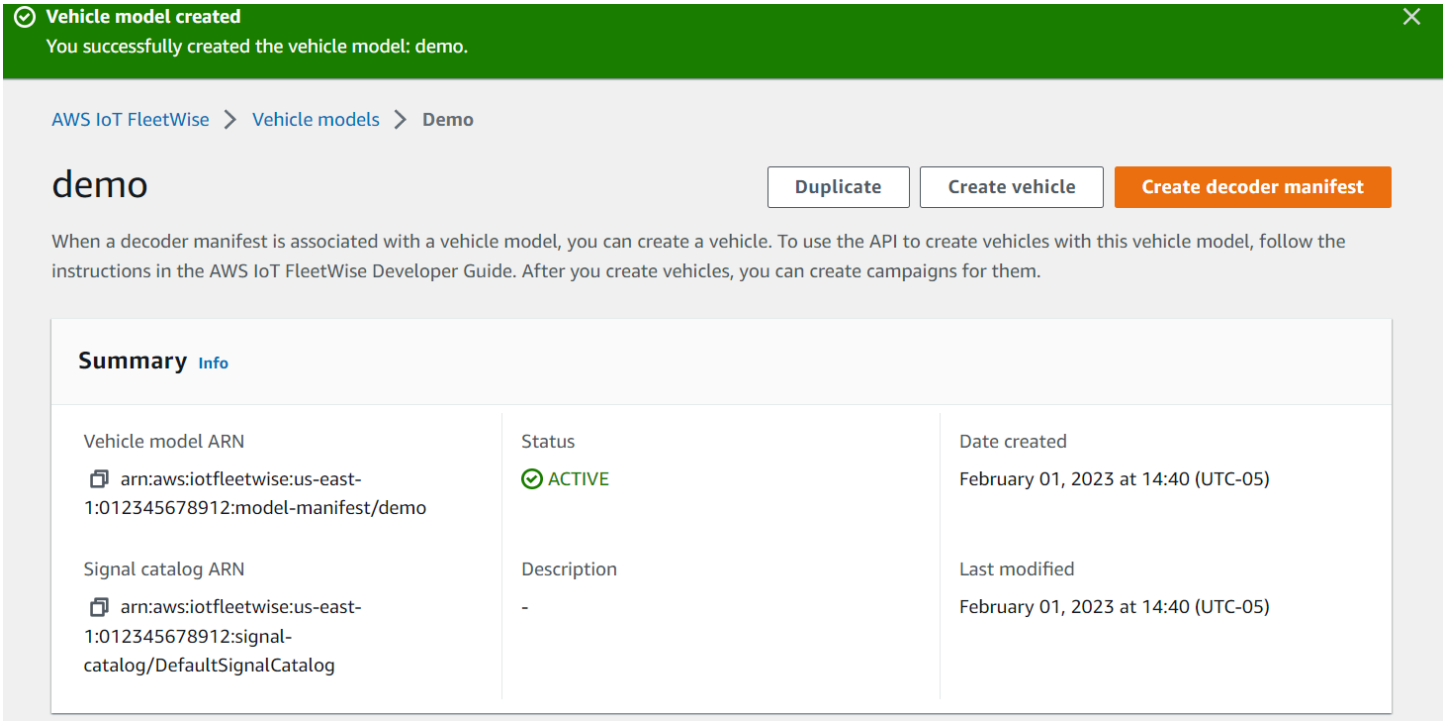
1. Pour vous aider à identifier le manifeste de votre décodeur, entrez un nom et une description facultative pour celui-ci. Ensuite, choisissez Suivant.
2. Pour ajouter une ou plusieurs interfaces réseau, choisissez le type CAN_INTERFACE ou OBD_INTERFACE.
   - Interface de diagnostic embarquée (OBD) - 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 (ECU). Ce protocole fournit un certain nombre de codes de diagnostic (DTC) standard 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 qui définit la manière dont les données sont communiquées entre les calculateurs. Les calculateurs peuvent être des unités de commande moteur, des 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 Create. 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, choisissez Campaigns.
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 plus d’informations, consultez Traitement et visualisation des données du véhicule.
2. Vous pouvez résoudre les problèmes liés à l' AWS IoT FleetWise. Pour plus d'informations, voir Résolution des problèmes liés à AWS l'IoT FleetWise.

# Ingestion de données dans le cloud

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**
>
> - AWSL'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'AWSIoT 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 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.
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 de signaux (AWS CLI) 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 en savoir plus, consultez Création d'un modèle de véhicule (console).
> - AWSL'IoT prend FleetWise actuellement en charge un catalogue de signaux pour chaque AWS compte et parRégion AWS.

3. Utilisez les signaux du catalogue de signaux pour créer un modèle de véhicule. Pour en savoir plus, consultez Création d'un modèle de véhicule.

> **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 en savoir plus, consultez Création d'un modèle de véhicule (console).
> - 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 en savoir plus, consultez Mettre à jour un modèle de véhicule (AWS CLI).
> - 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 en savoir plus, consultez Création et gestion des manifestes du décodeur.

> **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 en savoir plus, consultez Mettre à jour le manifeste d'un décodeur ()AWS CLI.
> - 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.

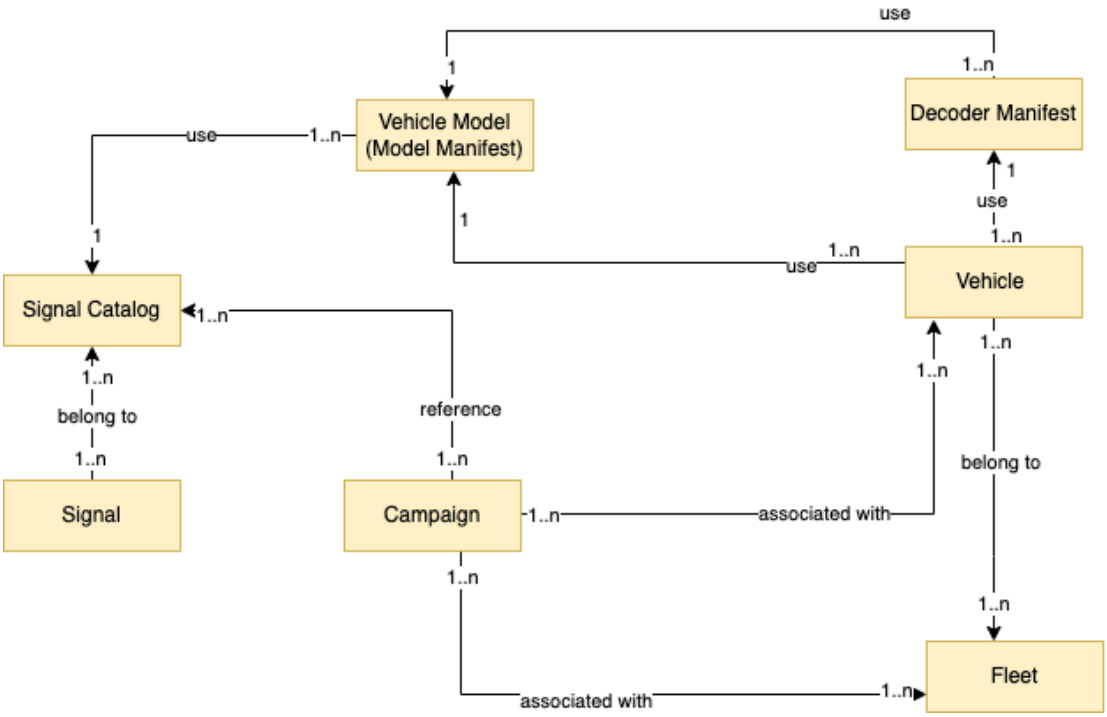
5. Créez des véhicules à partir du modèle du véhicule. Les véhicules créés à partir du même modèle de véhicule héritent du même groupe de signaux. Vous devez l'utiliser AWS IoT Core pour approvisionner votre véhicule avant de pouvoir ingérer des données dans le cloud. Pour en savoir plus, consultez Créez, approvisionnez et gérez des véhicules.
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 en savoir plus, consultez Créez et gérez des flottes.
7. 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 en savoir plus, consultez Collectez et transférez des données grâce aux campagnes.

> **Note**
>
> Vous devez utiliser l'opération de l'`UpdateCampaign`API pour approuver la campagne avant que l'AWSIoT ne FleetWise puisse la déployer sur le véhicule ou le parc de véhicules. Pour en savoir plus, consultez Mettre à jour une campagne (AWS CLI).

Le logiciel Edge Agent transfère les données du véhicule vers le sujet réservé`$aws/iotfleetwise/vehicles/`*`vehicleName`*`/signals`, qui envoie les données vers AWS l'IoT FleetWise. AWS IoT Core AWS L'IoT transmet FleetWise ensuite les données à une table Timestream ou à un compartiment Amazon S3. Vous pouvez utiliser Timestream pour interroger vos données, et utiliser Amazon ou QuickSight Grafana pour visualiser vos données. Pour plus d'informations, consultez Traitement et visualisation des données du véhicule.

# Modélisation de véhicules

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 plus d'informations, consultez Création et gestion de catalogues de signaux.

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 plus d'informations, consultez Création et gestion de catalogues de signaux.

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 plus d'informations, consultez Création et gestion de modèles de véhicules.

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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

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 de signaux (AWS CLI) 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 plus d'informations, consultez Création d'un modèle de véhicule (console).
   > - AWS L'IoT prend FleetWise actuellement en charge un catalogue de signaux pour chaque AWS compte et par Région AWS.

2. Utilisez les signaux du catalogue de signaux pour créer un modèle de véhicule. Pour plus d'informations, consultez Création d'un modèle de véhicule.

> **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 plus d'informations, consultez Création d'un modèle de véhicule (console).
> - 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 plus d'informations, consultez Mettre à jour un modèle de véhicule (AWS CLI).
> - 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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

> **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 plus d'informations, consultez Mettre à jour le manifeste d'un décodeur ()AWS CLI.
> - 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 signaux de décodeur. 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



# Création et gestion de catalogues de signaux

> **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.

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 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 plus d'informations, consultez Création d'un modèle de véhicule (console).
> - 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 plus d'informations, consultez Création d'un modèle de véhicule (console).
> - 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 selon 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 des signaux
- Création d'un catalogue de signaux (AWS CLI)
- Importer un catalogue de signaux
- Mettre à jour un catalogue de signaux (AWS CLI)
- Supprimer un catalogue de signaux (AWS CLI)
- Obtenir les informations du catalogue de signaux (AWS CLI)

## Configuration des signaux

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, UINT8, INT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, 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, UINT8, INT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, 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 timestamp). 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 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.

### 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, UINT8, INT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, UINT32_ARRAY, INT64_ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY, STRUCT, STRUCT_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 de signaux (AWS CLI)

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) Remplacez la *`description`* 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 des signaux.

```
{
    "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 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.

# Importer un catalogue de signaux

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 FleetWise console 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 :
     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 :

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 une *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 des signaux.

```
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.

## Mettre à jour un catalogue de signaux (AWS CLI)

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 des signaux.

> **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
     }
    }]
   }
```

## Supprimer un catalogue de signaux (AWS CLI)

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 véhicule
> - Supprimer un manifeste de décodeur

- Supprimer un véhicule
- Supprimer une flotte (AWS CLI)
- Supprimer une campagne

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
```

> **Note**
>
> Cette commande ne produit pas de sortie.

## Obtenir les informations du catalogue de signaux (AWS CLI)

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
```

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
```

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
```

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

## Création et gestion de modèles de véhicules

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 plus d'informations, consultez Créez, approvisionnez et gérez des véhicules.

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**
>
> - Si vous souhaitez utiliser l'opération `CreateModelManifest` API pour créer le premier modèle de véhicule, vous devez d'abord créer un catalogue de signaux. Pour plus d'informations, consultez Création d'un catalogue de signaux (AWS CLI).
> - 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 véhicule

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

> **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.

Rubriques



### 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

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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

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

1. Accédez à la FleetWiseconsole 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. Accédez à la FleetWiseconsole 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

Dans Informations générales, procédez comme suit.

1. Entrez un nom pour le modèle du véhicule.
2. (Facultatif) Entrez une description.
3. Choisissez 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 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 : 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, UINT8, INT16, UINT16, INT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY,

INT64_ARRAY _ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY ou 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. Comme dans 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 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, UINT8, INT16, UINT16, INT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX_TIMESTAMP, INT8_ARRAY, INT16_ARRAY, UINT16_ARRAY, INT32_ARRAY, INT64_ARRAY _ARRAY, UINT64_ARRAY, BOOLEAN_ARRAY, FLOAT_ARRAY, DOUBLE_ARRAY, STRING_ARRAY, UNIX_TIMESTAMP_ARRAY ou UNKNOWN
   - (Facultatif) Unité : unité scientifique de l'attribut, telle que le km ou le Celsius.

- (Facultatif) Chemin : chemin d'accès au signal. Comme dans 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 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. Accédez à la FleetWiseconsole 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**
>
> Si vous souhaitez utiliser l' FleetWise API AWS IoT pour créer le premier modèle de véhicule, vous devez d'abord créer un catalogue de signaux. Pour plus d'informations sur la création d'un catalogue de signaux, consultezCréation d'un catalogue de signaux (AWS CLI).

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.
- Remplacez *Signal-Catalog-ARN par* le Amazon Resource Name (ARN) du catalogue de signaux.
- (Facultatif) Remplacez la *description* 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 des signaux.

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

## Mettre à jour un modèle de véhicule (AWS CLI)

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) Remplacez la *description* 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"],
}
```

## Supprimer un modèle de véhicule

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 véhicule et Supprimer un manifeste de décodeur.

## 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. Accédez à la FleetWiseconsole 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. Sélectionnez Delete (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
```

> **Note**
>
> Cette commande ne produit pas de sortie.

## Obtenir des informations sur le modèle du véhicule (AWS CLI)

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
```

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
```

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

Remplacez *vehicle-model* 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.

# Création et gestion des manifestes du décodeur

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. Les signaux d'interface réseau et de

décodeur sont les principaux composants avec lesquels vous travaillez pour configurer les manifestes du décodeur.

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 (ECU). Les ECU peuvent 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 les calculateurs. Il fournit un certain nombre de codes de diagnostic standard (DTC) 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.

Signal du décodeur

Fournit des informations de décodage détaillées pour un signal spécifique. Chaque signal spécifié dans le modèle du véhicule doit être associé à un signal de décodeur. 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 les signaux du décodeur OBD.

Le manifeste du décodeur doit contenir les signaux du décodeur de messages s'il contient également des interfaces intergicielles pour véhicules.

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 les GetDecoderManifest API ListDecoderManifests et 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

- Obtenir les informations du manifeste du décodeur ()AWS CLI

## Configuration des interfaces réseau et des signaux du décodeur

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

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 les signaux du décodeur OBD.

Rubriques

- Configuration des interfaces réseau
- Configuration des signaux du décodeur

### 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 (DTC) au véhicule en quelques secondes. Par exemple, si la valeur spécifiée est 120, le logiciel Edge Agent collecte les DTC enregistrés 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 FleetWise est actuellement compatible avec la norme ISO 15765-4 du World Wide Harmonization On-Board Diagnostics (WWH-OBD).
- (Facultatif) `pidRequestIntervalSeconds` — À quelle fréquence demander des PID OBD II au véhicule. Par exemple, si la valeur spécifiée est 120, le logiciel Edge Agent collecte les PID OBD II toutes les 2 minutes.
- (Facultatif) `useExtendedIds` — S'il faut utiliser des identifiants étendus 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`

## Configuration des signaux du décodeur

Pour configurer un signal de décodeur 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.

Pour configurer un signal de décodeur 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.

Pour configurer un signal de décodeur 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.

## Création d'un manifeste de décodeur

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

> **Important**
>
> Vous devez disposer d'un modèle de véhicule pour pouvoir créer un manifeste de décodeur. Chaque manifeste du décodeur doit être associé à un modèle de véhicule. Pour plus d'informations, consultez Création et gestion de modèles de véhicules.

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. Accédez à la FleetWiseconsole 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 : ajouter des interfaces réseau

- 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 Suivant.

Étape 2 : 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

- Dans Interface réseau, procédez comme suit.
    a. Pour le type d'interface réseau, choisissez CAN_INTERFACE ou OBD_INTERFACE.
    b. Entrez un nom unique pour votre interface réseau.
    c. Entrez un ID d'interface réseau unique. Vous pouvez utiliser l'identifiant généré par AWS l'IoT FleetWise.
    d. Sélectionnez un ou plusieurs signaux spécifiés dans le modèle de votre véhicule pour les associer aux signaux du décodeur.
    e. Pour fournir des informations de décodage, téléchargez un fichier .dbc. AWS L'IoT FleetWise analyse le fichier .dbc pour récupérer les signaux du décodeur.
    f. Dans la section Signaux couplés, assurez-vous que chaque signal est associé à un signal de décodeur.
    g. Choisissez Suivant.

> **Note**
>
> - Vous ne pouvez télécharger qu'un seul fichier .dbc pour chaque interface réseau.
> - Assurez-vous que chaque signal spécifié dans le modèle de votre véhicule est associé à un signal de décodeur.

- Une fois que vous avez choisi d'ajouter une autre interface réseau, vous ne pouvez pas modifier celle que vous êtes en train de modifier. Vous pouvez supprimer toutes les interfaces réseau existantes.

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**
>
> Avant de créer un manifeste de décodeur, créez d'abord un modèle de véhicule. Pour plus d'informations, consultez Création d'un modèle de véhicule.

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.
- Remplacez le *Vehicle-Model-ARN par* le Amazon Resource Name (ARN) du modèle de véhicule.
- (Facultatif) Remplacez la *description* 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 des interfaces réseau 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.
- Remplacez le *Vehicle-Model-ARN par* le Amazon Resource Name (ARN) du modèle de véhicule.
- (Facultatif) Remplacez la *description* 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 des interfaces réseau 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"
         }
        }
       }
      ]
     }
    }
   }
  ]
}
```

> **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.

## Mettre à jour le manifeste d'un décodeur ()AWS CLI

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
```

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

## Supprimer un manifeste de décodeur

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 plus d'informations, consultez Supprimer un véhicule.

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 le manifeste d'un décodeur

1. Accédez à la FleetWiseconsole 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 plus d'informations, consultez Supprimer un véhicule.

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
```

## Obtenir les informations du manifeste du décodeur ()AWS CLI

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 des résumés de tous les manifestes du décodeur, exécutez la commande suivante.

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

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
```

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
```

Vous pouvez utiliser l'opération GetDecoderManifestAPI pour vérifier si les interfaces réseau et les signaux du décodeur dans le 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.

Remplacez *decoder-manifest 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.

# Créez, approvisionnez et gérez des véhicules

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, consultezModélisation de véhicules.

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

- Véhicules de ravitaillement
- Rubriques réservées
- Création d'un véhicule
- Mettre à jour un véhicule (AWS CLI)
- Mettre à jour plusieurs véhicules (AWS CLI)
- Supprimer un véhicule
- Obtenir des informations sur le véhicule (AWS CLI)

# Véhicules de ravitaillement

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 plus d'informations, consultez Création d'un véhicule (AWS CLI).

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.
  - Remplacez *le nom* de la politique par le nom de la stratégie que vous souhaitez créer.
  - Remplacez *file-name* 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 *la région* par la AWS région dans laquelle vous avez créé FleetWise les ressources AWS IoT.
- Remplacez *AWSAccount* 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 plus d'informations, consultez Rubriques réservées.

```
{
    "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.

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

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

## Rubriques réservées

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` | Publish | Le logiciel Edge Agent publie des informations sur l'état du véhicule dans cette rubrique.<br><br>Les informations sur l'état du véhicule sont échangées au format protobuf (protobuf). Pour plus d'informations, consultez le | |

| Rubrique | Opération du client autorisée | Description | |
|---|---|---|---|
| | | guide du développeur du FleetWise logiciel Edge Agent for AWS IoT. | |
| `$aws/iotfleetwise/vehicles/vehicleName/signals` | Publish | Le logiciel Edge Agent publie des signaux relatifs à cette rubrique.<br><br>Les informations de signal sont échangées au format protobuf (protobuf). Pour plus d'informations, consultez le guide du développeur du FleetWise logiciel Edge Agent for AWS IoT. | |
| `$aws/iotfleetwise/vehicles/vehicleName/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. | |

| Rubrique | Opération du client autorisée | Description | |
|---|---|---|---|
| `$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. | |

# Création d'un véhicule

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 êtreACTIVE. Pour plus d'informations, consultez Création et gestion de modèles de véhicules.
> - 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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

Rubriques

- Création d'un véhicule (console)
- Création d'un véhicule (AWS CLI)
- Créez plusieurs véhicules (AWS CLI)

## Création d'un véhicule (console)

Vous pouvez utiliser la FleetWise console 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 êtreACTIVE. Pour plus d'informations, consultez Création et gestion de modèles de véhicules.
> - 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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

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

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.

> **Important**
>
> Avant de commencer, vérifiez les points suivants :
>
> - Vous devez avoir un modèle de véhicule et le statut du modèle de véhicule doit êtreACTIVE. Pour plus d'informations, consultez Création et gestion de modèles de véhicules.
> - 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 plus d'informations, consultez Création et gestion des manifestes du décodeur.

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. Remplacez *thing-name* 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 plus d'informations, consultez Véhicules de ravitaillement.

- Remplacez le *nom du véhicule* par l'un des 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 (:).
- Remplacez *Model-manifest-ARN 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"
    }
}
```

## Créez plusieurs véhicules (AWS CLI)

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.

Remplacez *file-name* par le nom du fichier JSON contenant les configurations de plusieurs véhicules.

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

Example configurations de 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, voirCréation d'un véhicule (AWS CLI).

## Mettre à jour un véhicule (AWS CLI)

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.

Remplacez *file-name* 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

- Remplacez le *nom du véhicule* par l'identifiant du véhicule que vous souhaitez mettre à jour.
- (Facultatif) Remplacez *Model-manifest-ARN par* l'ARN du modèle de véhicule (manifeste du modèle) que vous utilisez pour remplacer le modèle de véhicule utilisé.
- (Facultatif) Remplacez *Decoder-Manifest-ARN* par l'ARN de votre manifeste de 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"
         }
}
```

## Mettre à jour plusieurs véhicules (AWS CLI)

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.

Remplacez *file-name* par le nom du fichier JSON contenant les configurations de plusieurs véhicules.

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

Example configurations de 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 véhicule (AWS CLI).

# Supprimer un véhicule

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 Créez et gérez des flottes et Collectez et transférez des données grâce aux 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. Accédez à la FleetWiseconsole AWS IoT.
2. Dans le volet de navigation, sélectionnez Véhicules.
3. Sur la page Véhicules, sélectionnez le bouton 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.

Remplacez le *`nom du véhicule`* par l'identifiant du véhicule que vous souhaitez supprimer.

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

# Obtenir des informations sur le véhicule (AWS CLI)

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
```

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.

Remplacez le *nom du véhicule* 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.

# Créez et gérez des flottes

Une flotte représente un groupe de véhicules. Une flotte sans véhicules associés est une entité vide. Avant de pouvoir utiliser la flotte pour gérer plusieurs véhicules en même temps, vous devez associer des véhicules à la flotte. Un véhicule peut appartenir à plusieurs flottes. Vous pouvez contrôler les données à collecter à partir d'une flotte de véhicules et à quel moment les collecter en déployant une campagne. Pour plus d'informations, veuillez consulter Collectez et transférez des données grâce aux campagnes.

Une flotte contient les informations suivantes.

`fleetId`

L'ID de la flotte.

(Facultatif) `description`

Une description qui vous aide à trouver la flotte.

`signalCatalogArn`

Amazon Resource Name (ARN) du catalogue de signaux.

AWSL'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 des identifiants de toutes les flottes auxquelles appartient le véhicule.
- ListVehiclesInFleet— Récupère une liste paginée de résumés de tous les véhicules d'une flotte.
- GetFleet— Récupère des informations sur une flotte.

Rubriques



# Création d'une flotte (AWS CLI)

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

> **Important**
>
> Vous devez avoir un catalogue de signaux avant de créer une flotte. Pour plus d'informations, veuillez consulter Création d'un catalogue de signaux (AWS CLI).

Pour créer une flotte, exécutez la commande suivante.

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

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

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

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

# Associer un véhicule à une flotte (AWS CLI)

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 plus d'informations, veuillez consulter Créez, approvisionnez et gérez des véhicules.
> - 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 l'ID de la flotte.
- Remplacez le *nom du véhicule* par l'identifiant du véhicule.

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

# Dissocier un véhicule d'une flotte (AWS CLI)

Vous pouvez utiliser l'opération d'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 l'ID de la flotte.
- Remplacez le *nom du véhicule* par l'identifiant du véhicule.

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

# Mettre à jour une flotte (AWS CLI)

Vous pouvez utiliser l'opération UpdateFleetd'API 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.

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

  La description peut comporter 1 à 2048 caractères.

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

# Supprimer une flotte (AWS CLI)

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, consultezDissocier un véhicule d'une flotte (AWS CLI).

Pour supprimer une flotte, exécutez la commande suivante.

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

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

# Obtenir des informations sur la flotte (AWS CLI)

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 de résumés de toutes les flottes, exécutez la commande suivante.

```
aws iotfleetwise list-fleets
```

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

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

Remplacez le *nom du véhicule* par l'identifiant du véhicule.

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

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 de résumés de tous les véhicules d'une flotte, exécutez la commande suivante.

Remplacez *fleet-id* par l'ID de la flotte.

```
aws iotfleetwise list-vehicles-in-fleet \
            --fleet-id fleet-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 l'ID 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 répercutées immédiatement.

# Collectez et transférez des données grâce aux campagnes

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, AWS l'IoT la déploie FleetWise automatiquement sur les véhicules. Vous pouvez choisir de déployer une campagne sur un véhicule ou 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.

> **Note**
>
> 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. 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.
>
> - Vous avez pris les dispositions nécessaires AWS IoT Core pour approvisionner votre véhicule. Pour en savoir plus, consultez Véhicules de ravitaillement.

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`

L'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 campagnes.

(Facultatif) `triggerMode` peut être 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.

(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) `conditionLanguageVersion`

Version du langage d'expression conditionnelle.

`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.

(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 où la campagne transférera les données du véhicule. Vous pouvez choisir de stocker les données dans Amazon S3 ou Amazon Timestream.

S3 est un mécanisme de stockage de données rentable qui offre des fonctionnalités 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.

(Facultatif) `dataExtraDimensions`

Vous pouvez ajouter un ou plusieurs attributs afin de fournir des informations supplémentaires pour un signal.

(Facultatif) `description`

Vous pouvez ajouter 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 (DTC) enregistrés 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 de l'air d'admission 2 (IAT2) est inférieure à la plage normale du capteur.

Par défaut (`OFF`), le logiciel Edge Agent n'envoie pas de codes de diagnostic.

(Facultatif) `expiryTime`

Vous pouvez définir 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`

Vous pouvez spécifier 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.

> **Important**
>
> Les signaux utilisés dans l'expression du schéma de collecte basé sur les conditions doivent être spécifiés dans ce champ.

`name`

Nom du signal à partir duquel les données sont collectées lorsque le schéma de collecte de données est invoqué.

(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 dès le 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 en savoir plus, consultez Mettre à jour une campagne (AWS CLI).
- `RUNNING` — La campagne est active.

- `SUSPENDED`— La campagne est suspendue. Pour reprendre la campagne, utilisez l'opération `UpdateCampaign` API.

AWSL'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

- Créer une campagne
- Mettre à jour une campagne (AWS CLI)
- Supprimer une campagne
- Obtenir des informations sur la campagne (AWS CLI)

## Créer une campagne

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. 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.

- Vous avez pris les dispositions nécessaires AWS IoT Core pour approvisionner votre véhicule. Pour en savoir plus, consultez Véhicules de ravitaillement.

Rubriques

- Création d'une campagne (console)
- Création d'une campagne (AWS CLI)
- Expressions logiques pour les campagnes

## Création d'une campagne (console)

Vous pouvez utiliser 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. Accédez à la FleetWiseconsole AWS IoT.
2. Dans le volet de navigation, choisissez Campaigns.
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 : définir la destination de stockage
- Étape 3 : Ajouter des véhicules
- Étape 4 : vérifier et créer
- Étape 5 : 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 Création et gestion de catalogues de signaux et Créez, approvisionnez et gérez des véhicules.

- Après avoir créé une campagne, vous devez l'approuver. Pour en savoir plus, consultez Étape 5 : Déployer une campagne.

## 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 sous forme de variable, d'opérateur de comparaison et de valeur de comparaison.

   Par exemple, si vous spécifiez l'`$variable.`myVehicle.InVehicleTemperature` > 50.0`expression, l'AWSIoT 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 campagnes.

   Entrez l'expression logique utilisée pour identifier les données à collecter.

4. (Facultatif) Vous pouvez spécifier la version linguistique de l'expression conditionnelle. La valeur par défaut est 1.

5. (Facultatif) Vous pouvez spécifier l'intervalle de déclenchement minimal, qui correspond à la durée minimale entre deux événements de collecte de données. Par exemple, si un signal change souvent, vous souhaiterez peut-être collecter des données plus lentement.

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) Vous pouvez modifier 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, vous pouvez spécifier 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**
     >
     > AWSL'IoT prend FleetWise actuellement en charge l'extension de format de fichier .json.

  2. AWSL'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.

### Spécifier les signaux

Vous pouvez spécifier les signaux à partir desquels collecter les données lorsque le schéma de collecte de données est invoqué.

> **Important**
>
> Les signaux utilisés dans l'expression du schéma de collecte basé sur les conditions doivent être spécifiés dans ce champ.

Pour spécifier les signaux à partir desquels collecter des données

1. Recherchez le nom complet du signal.

   > **Note**
   >
   > 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.

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 lorsque le schéma de collecte de données est invoqué.
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 2 : définir la destination de stockage

> **Note**
>
> Vous ne pouvez transférer les données du véhicule vers Amazon S3 que si la campagne contient des signaux de données du système de vision.

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

Choisissez la destination où vous souhaitez stocker les données collectées par la campagne. Vous pouvez transférer les données du véhicule vers Amazon S3 ou Amazon Timestream.

Dans les paramètres de destination, procédez comme suit :

- Choisissez S3 ou Timestream dans la liste déroulante.

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.

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.

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 de la forme sous laquelle les fichiers sont enregistrés dans S3.

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.

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.

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.

- Choisissez Suivant.

## Étape 3 : 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 sélectionnez-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 4 : vérifier et créer

Vérifiez les configurations de la campagne, puis choisissez Créer une campagne.

> **Note**
>
> Une fois la campagne créée, vous ou votre équipe devez la déployer sur les véhicules.

### Étape 5 : 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 Deploy (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 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 un stockage d'objets offrant une évolutivité, une disponibilité des données, une sécurité et des performances de pointe.

> **Important**
>
> Vous ne pouvez transférer les données du véhicule que si AWS l'IoT FleetWise est autorisé 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.

## 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 Création et gestion de catalogues de signaux, Créez, approvisionnez et gérez des véhicules et Créez et gérez des flottes.
> - Après la création d'une campagne, vous devez utiliser l'opération `UpdateCampaign` API pour l'approuver. Pour plus d'informations, consultez Mettre à jour une campagne (AWS CLI).

Pour créer une campagne, exécutez la commande suivante.

Remplacez *`file-name`* par le nom du fichier JSON contenant la configuration de la campagne.

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

- Remplacez *`campaign-name`* par le nom de la campagne que vous créez.
- *`signal-catalog-arn`*Remplacez-le par le Amazon Resource Name (ARN) du catalogue de signaux.
- Remplacez *`target-arn`* par l'ARN d'une flotte ou d'un véhicule que vous avez créé.
- Remplacez *`bucket-arn`* 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"
      }
    }
  ]
}
```

- Remplacez *campaign-name* par le nom de la campagne que vous créez.
- *signal-catalog-arn*Remplacez-le par le Amazon Resource Name (ARN) du catalogue de signaux.
- Remplacez *target-arn* par l'ARN d'une flotte ou d'un véhicule que vous avez créé.
- Remplacez *role-arn* par l'ARN du rôle d'exécution des tâches qui FleetWise autorise l'AWSIoT 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"
      }
    }
  ]
}
```

## Expressions logiques pour les campagnes

AWSL'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 campagne (AWS CLI)

Vous pouvez utiliser l'opération UpdateCampaignAPI pour mettre à jour une campagne existante. La commande suivante utiliseAWS CLI.

- Remplacez *campaign-name* par le nom de la campagne que vous mettez à jour.
- Remplacez *l'action* par l'une des actions 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 est redéployée sur tous les véhicules et les véhicules recommenceront à envoyer des données.
  - UPDATE— Met à jour la campagne en définissant des attributs et en les associant à un signal.

```
aws iotfleetwise update-campaign \
            --name campaign-name \
            --action action
```

## Supprimer une campagne

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. Accédez à la FleetWiseconsole AWS IoT.
2. Dans le volet de navigation, choisissez Campaigns.
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 le *`nom de la campagne`* par le nom du véhicule que vous supprimez.

```
aws iotfleetwise delete-campaign --name campaign-name
```

# Obtenir des informations sur la campagne (AWS CLI)

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
```

Vous pouvez utiliser l'opération GetCampaignAPI pour récupérer les informations du véhicule. L'exemple suivant repose sur AWS CLI.

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

Remplacez *`campaign-name`* 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.

# Traitement et visualisation des données du véhicule

Le FleetWise logiciel Edge Agent for AWS IoT transfère certaines données du véhicule 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 visualiser et les partager.

## Traitement des 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. |
| `time` | timestamp | Horodatage du point de données. |
| `measure_name` | varchar | Le nom du signal. |

| Nom de champ | Type de données | Description | |
|---|---|---|---|
| `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. | |

## Visualisation des 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 à l'aide d'Amazon QuickSight.

## Traitement des données du véhicule dans 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 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. | |

| Nom de champ | Type de données | Description |
| --- | --- | --- |
| `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. |

# Format d'objet S3

AWSL'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
```

# Analyse des données du véhicule stockées dans 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 pour les flux de travail d'étiquetage et d'apprentissage automatique (ML) en aval.

Pour plus d'informations, consultez les rubriques suivantes du manuel Amazon SageMaker 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 des 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 à l'aide d'Amazon 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 Amazon QuickSight ne prend pas en charge 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

# AWS CLIetAWSKits SDK

Cette section fournit des informations sur la fabrication deAWSIoT FleetWise Demandes d'API. Pour en savoir plus surAWSIoT FleetWise opérations et types de données, consultez le.AWSIoT FleetWise Référence API.

Pour utiliserAWSIoT FleetWise avec une variété de langages de programmation, utilisezAWSKits SDK, qui contiennent 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, utilisezAWSIoT FleetWise avec leAWS CLI. Vous pouvez contrôlerAWSIoT FleetWise, et vos autres services, à partir de la ligne de commande, et automatisez-les à l'aide 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



## Problèmes liés au manifeste du décodeur

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 surDRAFT, puis réessayez. |
| `UpdateOperationFailure.ConflictingModelUpdate` | AWSL'IoT FleetWise essaie de valider par rapport à un modèle de véhicule (manifest e du modèle) modifié par quelqu'un d'autre. Attendez et réessayez. |

| Erreur | Consignes pour la résolution des problèmes |
| --- | --- |
| `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` | AWSL'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. |
| `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. |

| Erreur | Consignes pour la résolution des problèmes |
| --- | --- |
| `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)` | AWSL'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 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 |

| Erreur | Consignes pour la résolution des problèmes |
| --- | --- |
| | 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)` | AWSL'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] [IoT FleetWiseEngine : :connect] : [Impossible d'initialiser la bibliothèque de persistance]
- Problème : Le logiciel Edge Agent ne collecte pas les PID de diagnostic intégrés (OBD) II ni les codes de diagnostic (DTC).
- 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. 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.

- ```
  [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 toutes les interfaces réseau définies dans la configuration sont disponibles, 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 en savoir plus, consultez Véhicules de ravitaillement.

Pour plus d'informations sur le dépannage, consultez Kit SDK des appareils AWS IoT pour C++les questions fréquemment posées.

## Problème : [ERREUR] [IoT FleetWiseEngine : :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 PID de diagnostic intégrés (OBD) II ni les codes de diagnostic (DTC).

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 les identifiants d'arbitrage étendus du Controller Area Network (bus CAN), 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.

# 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 qui s'appliquent 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

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.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- 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 140-2 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 (Federal Information Processing Standard) disponibles, consultez Federal Information Processing Standard (FIPS) 140-2 (Normes de traitement de l'information fédérale).

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS l'IoT FleetWise ou un autre outil Services AWS à l'aide de la console, de l'API ou AWS des SDK. 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 pris en charge 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.

## Chiffrement au repos

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.

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 plus d'informations, consultez the section called “Véhicules de ravitaillement”.

# Chiffrement des données

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 les API hébergées sur des FleetWise passerelles AWS IoT et accessibles via le réseau local. Ces API sont exposées via une connexion TLS soutenue par un certificat de serveur appartenant au connecteur AWS IoT FleetWise Edge. Pour l'authentification du client, ces API 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



## Chiffrement au repos

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 des FleetWise autorisations AWS IoT pour créer une clé gérée par le client en votre nom.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:RevokeGrant"
      ],
      "Effect": "Allow",
      "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 sur`KMS_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.

## Gestion des clés

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 avec des clés gérées par le client |
|---|---|---|
| Catalogue de signaux | | description |
| | 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 |

| AWS FleetWise Ressource IoT | Type de données | Champs chiffrés au repos avec des clés gérées par le client |
|---|---|---|
| | 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, scalabilité, pidResponseLength ServiceMode, StartByte, bitMaskLength bitRightS hift |
| Véhicule | | attributs |
| Campagne | | description |
| | conditionBasedCollectionSchéma | expression, minimumTriggerInte rval Ms conditionLanguageVersion, TriggerMode |
| | TimeBasedCollectionScheme | Période DMS |

> **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 Créer une AWS KMS clé.

     > **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 l'*identifiant de clé KMS* par l'ID de la clé KMS.

```
aws iotfleetwise put-encryption-configuration –kms-key-id KMS key id –encryption-type
 KMS_BASED_ENCRYPTION
```

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.

```
{
  "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:Decrypt",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*"
}
```

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.

# Contrôler l'accès avec AWS IoT FleetWise

Les sections suivantes expliquent comment contrôler l'accès à et depuis vos AWS IoT FleetWise ressources. 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.

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



## 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 des ACL pour tous 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.

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) et en Europe (Francfort).
> - 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.
- Remplacez *trust-policy* 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.
- Remplacez *permissions-policy* 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.

# 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

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 vous-même les demandes, consultez la section Signature des demandes AWS 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 en savoir plus, consultez Authentification multifactorielle dans le Guide de l'utilisateur AWS IAM Identity Center et Utilisation de l'authentification multifactorielle (MFA) dans l'interface AWS 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 root, consultez Tâches nécessitant des informations d'identification d'utilisateur root 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 tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification dans le Guide de l'utilisateur IAM.

Un groupe IAM est une identité qui concerne un ensemble d'utilisateurs IAM. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations

pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, consultez Quand créer un utilisateur IAM (au lieu d'un rôle) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un rôle IAM est une 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. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console changeant de rôle. 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 Utilisation de rôles IAM dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour obtenir des informations sur les rôles pour la fédération, consultez Création d'un rôle pour un fournisseur d'identité tiers (fédération) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique 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 Différence entre les rôles IAM et les politiques basées sur les ressources 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 la politique relative à la transmission de demandes FAS, consultez Sessions de transmission 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 une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.
  - Rôle lié 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 instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de 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 qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2 dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez Quand créer un rôle IAM (au lieu d'un utilisateur) dans le Guide de l'utilisateur IAM.

# Gestion des accès à l’aide de politiques

Vous contrôlez 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 Création de politiques IAM dans le Guide de l’utilisateur IAM.

Les politiques basées sur l’identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez 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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. 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 (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principals (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez Vue d'ensemble des listes de contrôle d'accès (ACL) dans le Guide du développeur Amazon Simple Storage Service.

### Autres types de politique

AWS prend en charge d'autres types de politiques moins 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 remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, consultez Limites d'autorisations pour des entités IAM dans le Guide de l'utilisateur IAM.

- Politiques de contrôle des services (SCP) — Les SCP sont des politiques JSON qui spécifient 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 les politiques de contrôle des services (SCP) 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 organisations et les SCP, consultez Fonctionnement des SCP dans le Guide de l'utilisateur AWS Organizations .
- Politiques de séance : les politiques de séance sont des politiques avancées que vous 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 séance 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

| Fonction IAM | AWS FleetWise Support pour l'IoT |
|---|---|
| Politiques basées sur l’identité | Oui |
| Politiques basées sur les ressources | Non |
| Actions de politique | Oui |
| Ressources de politique | Oui |
| Clés de condition d’une politique | Oui |
| ACL | 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 Création de politiques IAM 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 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. Des politiques basées sur les ressources sont, par exemple, les politiques de confiance de rôle IAM et des politiques de compartiment. 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 entre comptes 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 Différence entre les rôles IAM et les politiques basées sur une ressource 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 stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Pour consulter la liste des FleetWise actions AWS IoT, consultez la section 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 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 de leurs ARN, consultez la section Ressources définies par l' AWS 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 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 IoT, voir. Exemples de politiques basées sur l'identité pour l'IoT AWS FleetWise

## Listes de contrôle d'accès (ACL) dans l'IoT AWS FleetWise

| | |
|---|---|
| Prend en charge les listes ACL | Non |

Les listes de contrôle d’accès (ACL) vérifient quels principaux (membres de compte, utilisateurs ou rôles) ont l’autorisation d’accéder à une ressource. Les listes de contrôle d’accès sont similaires aux politiques basées sur les ressources, bien qu’elles n’utilisent pas le format de document de politique JSON.

## Contrôle d'accès basé sur les attributs (ABAC) avec IoT AWS FleetWise

| | |
|---|---|
| Prise en charge d’ABAC (identifications dans les politiques) | Partielle |

Le contrôle d’accès basé sur les attributs (ABAC) est une politique 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 l’ABAC, consultez Qu’est-ce que le contrôle d’accès basé sur les attributs (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 Changement de rôle (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 fonctions de service | Non |

Une fonction du service est un rôle IAM qu'un service endosse pour accomplir des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez Création d'un rôle pour la délégation d'autorisations à un Service AWS dans le Guide de l'utilisateur IAM.

> **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 plus d'informations, consultez Surveillance de 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 AWSServiceRoleForIoTFleetWise lié à un service fait confiance aux services suivants pour assumer le rôle :

- `IoTFleetWise`

La politique d'autorisation de rôle nommée AWSIoTFleetwiseServiceRolePolicy 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 plus d'informations, consultez Configuration des paramètres.

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 AWSServiceRoleForIoTFleetWise lié à un service 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 guide de l'utilisateur IAM.

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 dans l’onglet JSON 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 format des ARN pour chacun des types de ressources, 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.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation de IAM pour appliquer des autorisations, consultez politiques et autorisations dans IAM dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées 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 IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez Validation de politique 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 le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez Configuration de l'accès aux API protégé par MFA dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez Bonnes pratiques de sécurité dans IAM dans le Guide de l'utilisateur IAM.

## Utilisation de la 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 stratégie basée sur l'identité qui est plus restrictive que l'ensemble minimum d'autorisations requis, la console ne fonctionnera pas comme prévu pour les entités (utilisateurs ou rôles) tributaires de cette stratégie.

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 l' AWS IoT FleetWise `ConsoleAccess` ou la politique `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, voir Configurer les paramètres.

## 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 (ACL), vous pouvez utiliser ces politiques pour donner l'accès à vos ressources.

Pour en savoir plus, consultez les éléments suivants :

- Pour savoir si AWS 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 découvrir quelle est la différence entre l'utilisation des rôles et l'utilisation des politiques basées sur les ressources pour l'accès entre comptes, consultez Différence entre les rôles IAM et les politiques basées sur les ressources dans le Guide de l'utilisateur IAM.

## 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 :

- Guides de démarrage rapide sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur AWS la sécurité et la conformité.
- Architecture axée sur la sécurité et la conformité HIPAA sur Amazon Web Services : ce livre blanc décrit comment les entreprises peuvent créer des applications AWS conformes à la loi HIPAA.

  > **Note**
  >
  > Tous ne Services AWS sont pas éligibles à la loi HIPAA. Pour plus d'informations, consultez le HIPAA Eligible Services Reference.

- 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.

# 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 depuis des points de terminaison Amazon Virtual Private Cloud (Amazon VPC) spécifiques ou des VPC spécifiques. 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 d'un VPC est représenté par une ou plusieurs interfaces réseau Elastic (ENI) 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 à partir d'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 pour AWS l'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 plus d'informations, consultez la section Interface VPC endpoints (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.

## 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 un accès complet uniquement à un utilisateur 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 de point de terminaison pour AWS l'IoT FleetWise. *Lorsqu'elle est attachée à un point de terminaison, cette politique autorise l'utilisateur IAM FleetWise à accéder aux FleetWise actions AWS IoT répertoriées dans le 123456789012. Compte AWS*

```
{
    "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 les identifiants des FleetWise ressources AWS IoT, par exemple dans les noms de campagnes, les manifestes de décodeurs, les modèles de véhicules et les catalogues de signaux, ou les identifiants 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 plus d’informations, consultez the section called “CloudTrailJournaux ”.

## 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 .

# Surveillance de 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. AWSfournit 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 instances Amazon EC2 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'instances Amazon EC2 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 qui ont appelé AWS, l’adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le Guide de l’utilisateur AWS CloudTrail.

## Surveillance de AWS l'IoT FleetWise avec Amazon CloudWatch

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 ConsoleAWS CLI, le ou une API pour répertorier les métriques auxquelles l'AWSIoT 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 en savoir plus, consultez Configuration des paramètres.

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.<br><br>Unités : nombre<br><br>Dimensions : VehicleName<br><br>Statistiques valides : somme |
| MessageThrottled | Un message envoyé par le véhicule à l'AWSIoT FleetWise a été limité. Cela est dû au fait que vous avez dépassé les limites de service pour ce compte dans la région actuelle.<br><br>Unités : nombre<br><br>Dimensions : VehicleName<br><br>Statistiques valides : somme |
| ModelingError | Un message envoyé par le véhicule et reçu par l'AWSIoT FleetWise contient des signaux qui ne sont pas validés par rapport au modèle du véhicule.<br><br>Unités : nombre<br><br>Dimensions : ModelManifestName |

| Métrique | Description |
| --- | --- |
| | Statistiques valides : somme |
| DecodingError | Un message envoyé depuis le véhicule et reçu par l'AWSIoT FleetWise contient des signaux qui ne sont pas décodés par rapport au manifeste du décodeur du véhicule.<br><br>Unités : nombre<br><br>Dimensions : DecoderName<br><br>Statistiques valides : somme |

Métriques de campagne

| Métrique | Description |
| --- | --- |
| VehicleNotFound | Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, lorsque le véhicule est inconnu.<br><br>Unités : nombre<br><br>Dimensions : VehicleName<br><br>Statistiques valides : somme |
| CampaignInvalid | Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, lorsque la campagne n'est pas valide.<br><br>Unités : nombre<br><br>Dimensions : CampaignName<br><br>Statistiques valides : somme |

| Métrique | Description |
| --- | --- |
| CampaignNotFound | Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, dont la campagne est inconnue.<br><br>Unités : nombre<br><br>Dimensions : CampaignName<br><br>Statistiques valides : somme |

Indicateurs de destination des données de campagne

| Métrique | Description |
| --- | --- |
| TimestreamWriteError | AWSL'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans le tableau Amazon Timestream.<br><br>Unités : nombre<br><br>Dimensions : DatabaseName, TableName<br><br>Statistiques valides : somme |
| S3 WriteError | AWSL'IoT n'a pas FleetWise pu écrire de message depuis le véhicule vers le compartiment Amazon Simple Storage Service (Amazon S3).<br><br>Unités : nombre<br><br>Dimensions : BucketName<br><br>Statistiques valides : somme |
| S3 ReadError | AWSL'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<br><br>Dimensions : BucketName<br><br>Statistiques valides : somme |

Indicateurs AWS KMS clés gérés par le client

| Métrique | Description |
| --- | --- |
| KMS KeyAccessDenied | AWSL'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.<br><br>Unités : nombre<br><br>Dimension : KMS KeyId<br><br>Statistiques valides : somme |

# Surveillance de AWS l'IoT FleetWise avec Amazon CloudWatch Logs

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 de l'AWSIoT 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 console CloudWatch .
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 deERROR.

   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'AWSIoT FleetWise contient des signaux qui ne sont pas validés par rapport au modèle du véhicule.<br><br>Attributs : nom du véhicule, nom de la campagne, identifiant du signal signalCat alogName, valeur du signal, min, maximum, signalValueRange signalValueRange modelManifestName |
| 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.<br><br>Attributs : nom du véhicule, nom de la campagne, signalCatalogName |
| ERREUR_DE DÉCODAGE | Un message envoyé depuis le véhicule et reçu par l'AWSIoT FleetWise contient des signaux qui ne sont pas décodés par rapport au manifeste du décodeur du véhicule.<br><br>Attributs : CampaignName, signalCat alogName, decoderManifestName, (facultatif) SignalName, (facultatif) S3URI |

Types d'événements liés à la campagne

| Type d’événement | Description |
| --- | --- |
| VÉHICULE INTROUVABLE | Message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, alors que le véhicule était inconnu. |

| Type d’événement | Description |
| --- | --- |
| | Attributs : nom du véhicule, nom de la campagne |
| CAMPAGNE_INTROUVABLE | Un message envoyé depuis le véhicule et reçu par AWS l'IoT FleetWise, alors que la campagne était inconnue.<br><br>Attributs : VehicleName (facultatif), CampaignName |
| CAMPAGNE_INVALID | Message envoyé depuis le véhicule et reçu par l'AWSIoT FleetWise, pour lequel la campagne n'était pas valide.<br><br>Attributs : VehicleName (facultatif), CampaignName |

Types d'événements de destination des données de campagne

| Type d’événement | Description |
| --- | --- |
| TIMESTREAM_WRITE_ERROR | AWSL'IoT FleetWise n'a pas pu écrire de message depuis le véhicule dans le tableau Amazon Timestream.<br><br>Attributs : nom du véhicule, nom de la campagne, timestreamDatabaseName timestreamTableName |
| S3_WRITE_ERROR | AWSL'IoT n'a pas FleetWise pu écrire de message depuis le véhicule vers le compartiment Amazon Simple Storage Service (Amazon S3).<br><br>Attributs : CampaignName, DestinationName |

| Type d’événement | Description |
| --- | --- |
| S3_READ_ERROR | AWSL'IoT n' FleetWise a pas pu lire une clé d'objet depuis le véhicule dans le compartiment Amazon Simple Storage Service (Amazon S3).<br><br>Attributs : CampaignName, DestinationName |

AWS KMSPrincipaux types d'événements gérés par le client

| Type d’événement | Description |
| --- | --- |
| KMS_KEY_ACCESS_DENIED | AWSL'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

Toutes les entrées CloudWatch des journaux incluent les attributs suivants :

accountId

Votre Compte AWS carte d'identité.

eventType

Le type d'événement pour lequel le journal a été créé. La valeur du type d'événement dépend de l'événement qui a généré l'entrée de journal. Chaque description d'entrée de journal inclut la valeur de `eventType` pour cette entrée de journal.

logLevel

Le niveau de journalisation utilisé. Pour plus d'informations, 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.

Identifiants de signal

Liste des identifiants de signaux d'erreur.

Nom du signal

Le nom du signal.

signalTimestampEpochMme

Horodatage du signal d'erreur.

Valeur du signal

La valeur du signal d'erreur.

signalValueRangeMaximum

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.

## Configuration de 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 effectuer cette procédure avec l'API en utilisant les méthodes de l'API AWS qui correspondent aux commandes d'interface de ligne de commande indiquées ici. 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 à utiliser 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.

# JournalisationAWSIoTFleetWiseAppels d'API utilisantAWS CloudTrail

AWSIoTFleetWiseest intégré àAWS CloudTrail, un service qui fournit un enregistrement des actions effectuées par un utilisateur, un rôle ou unAWSservice dansAWSIoTFleetWise. CloudTrailcapture tous les appels d'API pourAWSIoTFleetWiseen tant qu'événements. Les appels capturés incluent les appels provenant duAWSIoTFleetWiseappels de console et de code versAWSIoTFleetWiseOpérations d'API. Si vous créez un journal, vous pouvez activer la diffusion continue deCloudTrailévénements vers un compartiment Amazon S3, y compris des événements pourAWSIoTFleetWise. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Historique des événements. Utilisation des informations collectées parCloudTrail, vous pouvez déterminer la demande qui a été faite àAWSIoTFleetWise, l'adresse IP à partir de laquelle la demande a été faite, qui l'a faite, quand elle a été faite et des détails supplémentaires.

Pour en savoir plus sur CloudTrail, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.

## AWSIoTFleetWiseinformations dansCloudTrail

CloudTrail est activé sur votre compte AWS lorsque vous créez le compte. Lorsque l'activité se produit dansAWSIoTFleetWise, cette activité est enregistrée dans unCloudTrailévénement avec

d'autresAWSévénements de service àHistorique de l'événement. Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS. Pour plus d'informations, consultez Affichage des événements avec l'historique des événements CloudTrail.

Pour un enregistrement continu des événements survenus dans votreAWScompte, y compris les événements pourAWSIoTFleetWise, créez un parcours. UNsentierpermetCloudTrailpour envoyer des fichiers journaux vers un compartiment Amazon S3. Par défaut, lorsque vous créez un journal d'activité dans la console, il s'applique à toutes les régions AWS. Le journal d'activité consigne les événements de toutes les régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour en savoir plus, consultez les ressources suivantes :

- Présentation de la création d'un journal d'activité
- Intégrations et services pris en charge par CloudTrail
- Configuration des notifications Amazon SNS pour CloudTrail
- Réception des fichiers journaux CloudTrail de plusieurs régions
- Réception des fichiers journaux CloudTrail de plusieurs comptes

TousAWSIoTFleetWiseles actions sont enregistrées parCloudTrailet sont documentés dans leAWSIoTFleetWiseRéférence d'API. Par exemple, les appels adressés aux actions `CreateCampaign` `AssociateVehicleFleet`, `GetModelManifest` génèrent des entrées dans les fichiers journaux CloudTrail.

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été effectuée avec les autorisations utilisateur root ou IAM .
- Si la demande a été effectuée avec des autorisations de sécurité temporaires pour un rôle ou un utilisateur fédéré.
- Si la requête a été effectuée par un autre service AWS.

Pour de plus amples informations, veuillez consulter l'élément userIdentity CloudTrail.

# CompréhensionAWSIoTFleetWiseentrées du fichier journal

Un journal d'activité est une configuration qui permet d'envoyer des événements sous forme de fichiers journaux à un compartiment Simple Storage Service (Amazon S3) que vous spécifiez. Les fichiers journaux CloudTrail contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la requête, etc. Les fichiers journaux CloudTrail ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple suivant montre une entrée de journal CloudTrail qui illustre l'opération *`AssociateVehicleFleet`*.

```
{
      "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 |
| --- | --- | --- |
| 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 KMSclés gérées par le client | AWSL'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, | 16 octobre 2023 |

| | | |
|---|---|---|
| | véhicules et configurations de campagnes de collecte de données) stockées dans. AWS Cloud | |
| Stockage d'objets dans Amazon S3 | AWSL'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'AWSIoT FleetWise. | 27 septembre 2022 |
| Première version | Il s'agit de la version préliminaire 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.