



Guia do Desenvolvedor

# AWS IoT FleetWise



# AWS IoT FleetWise: Guia do Desenvolvedor

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

As marcas comerciais e imagens comerciais da Amazon não podem ser usadas no contexto de nenhum produto ou serviço que não seja da Amazon, nem de qualquer maneira que possa gerar confusão entre os clientes ou que deprecie ou desprestige a Amazon. Todas as outras marcas comerciais que não são propriedade da Amazon pertencem aos respectivos proprietários, os quais podem ou não ser afiliados, estar conectados ou ser patrocinados pela Amazon.

---

# Table of Contents

O que é o AWS IoT FleetWise? .....	1
Benefícios .....	2
Casos de uso .....	2
O AWS IoT FleetWise é novidade para você? .....	3
Acessar o AWS IoT FleetWise .....	3
Preços do AWS IoT FleetWise .....	3
Como funciona o AWS IoT FleetWise .....	3
Principais conceitos .....	4
Recursos do AWS IoT FleetWise .....	8
Serviços relacionados .....	9
Configurando a AWS IoT FleetWise .....	10
Configure sua Conta da AWS .....	10
Inscreva-se para uma Conta da AWS .....	10
Criar um usuário com acesso administrativo .....	11
Comece pelo console .....	12
Configuração de definições .....	12
Definir configurações (console) .....	13
Definir configurações (AWS CLI) .....	14
Conceitos básicos .....	16
Requisitos .....	16
Usando a demonstração do software Edge Agent .....	16
Conceitos básicos (console) .....	17
Pré-requisitos .....	18
Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise .....	19
Etapa 2: criar um modelo de veículo .....	20
Etapa 3: criar um manifesto do decodificador .....	22
Etapa 4: configurar um manifesto do decodificador .....	23
Etapa 5: criar um veículo .....	24
Etapa 6: criar uma campanha .....	25
Etapa 7: limpar .....	27
Próximas etapas .....	27
Ingestão de dados na nuvem .....	28
Modelagem de veículos .....	31
Catálogos de sinais .....	34

Configurar sinais .....	37
Criar um catálogo de sinais (AWS CLI) .....	43
Importar um catálogo de sinais .....	48
Atualizar um catálogo de sinais (AWS CLI) .....	57
Excluir um catálogo de sinais (AWS CLI) .....	59
Obter informações do catálogo de sinal (AWS CLI) .....	60
Modelos de veículos .....	61
Criar um modelo de veículo .....	62
Atualizar o modelo de um veículo (AWS CLI) .....	68
Excluir um modelo de veículo .....	69
Obter informações sobre o modelo do veículo (AWS CLI) .....	70
Manifestos de decodificadores .....	71
Configurar interfaces de rede e sinais de decodificador .....	73
Criar um manifesto do decodificador .....	76
Atualizar um manifesto do decodificador (AWS CLI) .....	84
Excluir um manifesto do decodificador .....	85
Obter informações de manifesto do decodificador (AWS CLI) .....	86
Veículos .....	88
Provisionar veículos .....	89
Autenticar veículos .....	90
Autorizar veículos .....	92
Tópicos reservados .....	93
Criar um veículo .....	95
Criar um veículo (console) .....	95
Criar um veículo (AWS CLI) .....	97
Criar vários veículos (AWS CLI) .....	99
Atualizar um veículo (AWS CLI) .....	100
Atualizar vários veículos (AWS CLI) .....	102
Excluir um veículo .....	103
Excluir um veículo (console) .....	103
Excluir um veículo (AWS CLI) .....	103
Obter informações do veículo (AWS CLI) .....	104
Frotas .....	105
Criar uma frota (AWS CLI) .....	106
Associar um veículo a uma frota (AWS CLI) .....	107
Desassociar um veículo de uma frota (AWS CLI) .....	107

Atualizar uma frota (AWS CLI) .....	108
Excluir uma frota (AWS CLI) .....	108
Obter informações da frota (AWS CLI) .....	108
Campanhas .....	110
Criar uma campanha .....	115
Criar uma campanha (console) .....	116
Criar uma campanha (AWS CLI) .....	123
Expressões lógicas para campanhas .....	127
Atualizar uma campanha (AWS CLI) .....	128
Excluir uma campanha .....	129
Excluir uma campanha (console) .....	129
Excluir uma campanha (AWS CLI) .....	129
Obter informações da campanha (AWS CLI) .....	129
Processamento e visualização de dados do veículo .....	131
Processamento dos dados do veículo no Timestream .....	131
Visualização de dados do veículo armazenados no Timestream .....	132
Processamento de dados do veículo no S3 .....	132
Formato de objeto S3 .....	133
Analisar dados do veículo armazenados no S3 .....	133
AWS CLI e AWS SDKs .....	137
Solução de problemas .....	138
Problemas com o manifesto do decodificador .....	138
Problemas do software Edge Agent do AWS IoT FleetWise .....	142
Problema: o software Edge Agent não inicia. ....	142
Problema: [ERROR] [IoTFleetWiseEngine::connect]: [Failed to init persistency library] .....	144
Problema: o software Edge Agent não coleta PIDs de diagnóstico integrado (OBD) II e códigos de problemas de diagnóstico (DTCs). ....	144
Problema: o software Edge Agent do AWS IoT FleetWise não coleta dados da rede ou não consegue aplicar regras de inspeção de dados. ....	144
Problema: [ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error] ou [WARN] [AwsIotChannel::send]: [No alive MQTT Connection.] .....	145
Segurança .....	146
Proteção de dados .....	147
Criptografia inativa .....	148
Criptografia em trânsito .....	148
Criptografia de dados .....	149

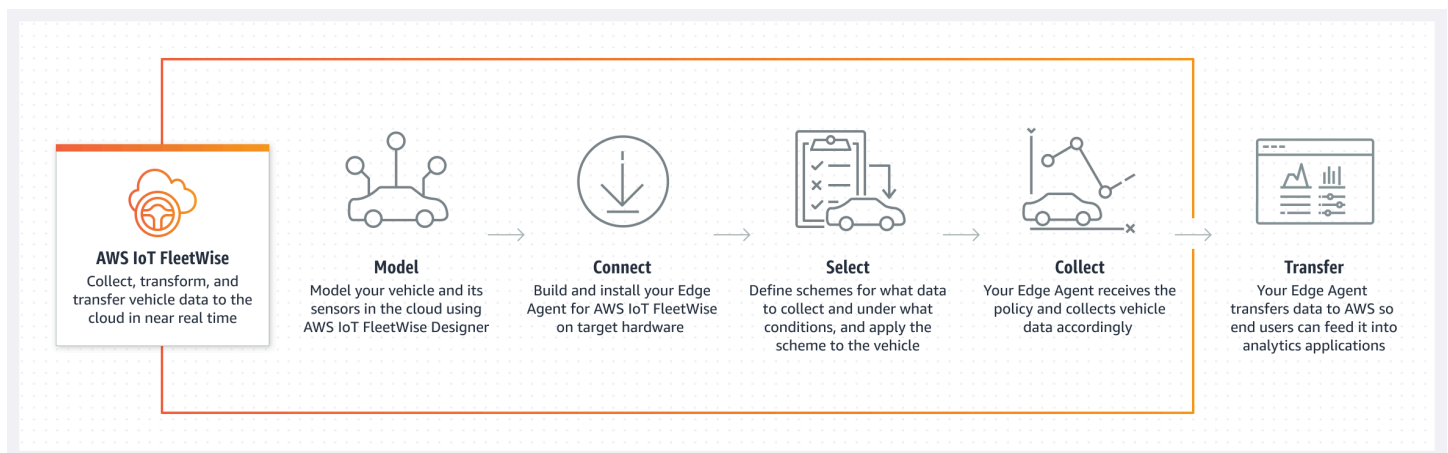
Controlar o acesso .....	157
Conceder AWS IoT FleetWise acesso a um destino do Amazon S3 .....	157
Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream .....	160
Identity and Access Management .....	163
Público .....	164
Autenticando com identidades .....	164
Gerenciamento do acesso usando políticas .....	168
Como a AWS IoT FleetWise funciona com o IAM .....	171
Exemplos de políticas baseadas em identidade .....	181
Solução de problemas .....	184
Compliance Validation .....	186
Resiliência .....	188
Segurança da infraestrutura .....	188
Conectando-se à AWS IoT FleetWise por meio de uma interface VPC endpoint .....	189
Análise de configuração e vulnerabilidade .....	192
Melhores práticas de segurança .....	193
Conceder o mínimo possível de permissões .....	193
Não registrar em log informações confidenciais .....	193
Use AWS CloudTrail para visualizar o histórico de chamadas da API .....	194
Manter o relógio do dispositivo sincronizado .....	194
Monitoramento .....	195
Monitoramento com CloudWatch .....	195
Monitorar com o CloudWatch Logs .....	199
Visualizar logs do AWS IoT FleetWise no console do CloudWatch .....	199
Configurar o registro em log do .....	205
Logs do CloudTrail .....	208
Informações sobre o AWS IoT FleetWise no CloudTrail .....	208
Noções básicas sobre entradas do arquivo de log do AWS IoT FleetWise .....	209
Histórico do documento .....	211
.....	ccxiii

# O que é o AWS IoT FleetWise?

O AWS IoT FleetWise é um serviço gerenciado que você pode usar para coletar dados do veículo e organizá-los na nuvem. É possível usar os dados coletados para melhorar a qualidade, o desempenho e a autonomia do veículo. Com o AWS IoT FleetWise, você pode coletar e organizar dados de veículos que utilizam diferentes protocolos e formatos de dados. O AWS IoT FleetWise ajuda a transformar mensagens de baixo nível em valores legíveis por humanos e a padronizar o formato de dados na nuvem para análise de dados. Você também pode definir campanhas de coleta de dados para controlar quais dados do veículo serão coletados e quando transferir esses dados para a nuvem.

Quando os dados do veículo estão na nuvem, você pode usá-los para aplicativos que analisam a saúde da frota de veículos. Esses dados ajudam a identificar possíveis problemas de manutenção, tornar mais inteligentes os sistemas de infotainment em veículos e melhorar tecnologias avançadas, como direção autônoma e sistemas de assistência ao motorista com análises e machine learning (ML).

O diagrama a seguir mostra a arquitetura básica do AWS IoT FleetWise.



## Tópicos

- [Benefícios](#)
- [Casos de uso](#)
- [O AWS IoT FleetWise é novidade para você?](#)
- [Acessar o AWS IoT FleetWise](#)
- [Preços do AWS IoT FleetWise](#)

- [Como funciona o AWS IoT FleetWise](#)
- [Serviços relacionados](#)

## Benefícios

Veja os principais benefícios do AWS IoT FleetWise:

Coletar dados do veículo de forma mais inteligente

Melhore a relevância dos dados com a coleta inteligente de dados que envia à nuvem para análise somente os dados de que você precisa.

Analisar facilmente os dados padronizados de toda a frota

Analise dados padronizados de uma frota de veículos sem precisar desenvolver um sistema personalizado de coleta ou registro em log de dados.

Sincronização automática de dados na nuvem

Tenha uma visão unificada dos dados coletados dos sensores-padrão (dados de telemetria) e dos sistemas de visão (dados de câmeras, radares e lidars) e mantenha-os sincronizados automaticamente na Nuvem AWS. O IoT FleetWise mantém dados de sistemas de visão estruturados e não estruturados, metadados e dados de sensores-padrão sincronizados automaticamente na nuvem. Isso simplifica o processo de montar uma visão geral dos eventos e receber informações.

### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Casos de uso

Os cenários nos quais o AWS IoT FleetWise pode ser usado incluem:

Treinar modelos de IA/ML

Melhore continuamente os modelos de machine learning usados em sistemas autônomos e avançados de assistência ao motorista coletando dados de veículos de produção.



## Melhorar a experiência digital do cliente

Use dados de sistemas de infotenimento para tornar mais relevantes as informações no aplicativo e os conteúdos audiovisuais no veículo.

## Manter a integridade da frota de veículos

Use informações dos dados da frota para monitorar a integridade da bateria de EV e os níveis de carga, gerenciar cronogramas de manutenção, analisar o consumo de combustível e muito mais.

# O AWS IoT FleetWise é novidade para você?

Se você for iniciante no AWS IoT FleetWise, recomendamos ler as seguintes seções para começar:

- [Como funciona o AWS IoT FleetWise](#)
- [Configurando a AWS IoT FleetWise](#)
- [Demo do software Edge Agent](#)
- [Ingestão de dados na nuvem](#)

## Acessar o AWS IoT FleetWise

Use a API ou o console do AWS IoT FleetWise para acessar o AWS IoT FleetWise.

## Preços do AWS IoT FleetWise

Os veículos enviam dados para a nuvem por meio de mensagens MQTT. Você paga no final de cada mês pelos veículos que criou no AWS IoT FleetWise. Você também paga pelas mensagens coletadas dos veículos. Para obter informações atuais sobre preços, consulte a página [Preços do AWS IoT FleetWise](#). Para saber mais sobre o protocolo de mensagens MQTT, consulte [MQTT](#) no Guia do desenvolvedor do AWS IoT Core.

## Como funciona o AWS IoT FleetWise

As seções a seguir fornecem uma visão geral dos componentes do serviço AWS IoT FleetWise e como eles interagem.

Depois de ler esta introdução, consulte a seção [Configurando a AWS IoT FleetWise](#) para saber como configurar o AWS IoT FleetWise.

## Tópicos

- [Principais conceitos](#)
- [Recursos do AWS IoT FleetWise](#)

## Principais conceitos

O AWS IoT FleetWise fornece uma estrutura de modelagem de veículos para você modelar seu veículo e os sensores e atuadores na nuvem. Para permitir a comunicação segura entre seu veículo e a nuvem, o AWS IoT FleetWise também fornece uma implementação de referência para ajudar a desenvolver o software Edge Agent que você pode instalar em seu veículo. Você pode definir esquemas de coleta de dados na nuvem e implantá-los no veículo. O software Edge Agent executado no veículo usa esquemas de coleta de dados para controlar quais dados coletar e quando transferi-los para a nuvem.

Veja a seguir os conceitos básicos do AWS IoT FleetWise.

### Signal

Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. Um sinal pode ser um atributo, uma ramificação, um sensor ou um atuador. Por exemplo, você pode criar um sensor para receber valores de temperatura no veículo e armazenar os metadados, incluindo um nome de sensor, um tipo de dados e uma unidade. Para obter mais informações, consulte [Criar e gerenciar catálogos de sinais](#).

### Atributo

Os atributos representam informações estáticas que geralmente não mudam, como fabricante e data de fabricação.

### Ramificação

As ramificações representam sinais em uma estrutura aninhada. As ramificações demonstram hierarquias de sinais. Por exemplo, a ramificação `Vehicle` tem uma ramificação secundária, `Powertrain`. A ramificação `Powertrain` tem uma ramificação secundária, `combustionEngine`. Para localizar a ramificação `combustionEngine`, use a expressão `Vehicle.Powertrain.combustionEngine`.

### Sensor

Os dados do sensor informam o estado atual do veículo e mudam ao longo do tempo, à medida que o estado do veículo muda, como níveis de fluidos, temperaturas, vibrações ou tensão.

## Actuator

Os dados do atuador relatam o estado de um dispositivo do veículo, como motores, ar quente e fechaduras de portas. A alteração do estado de um dispositivo do veículo pode atualizar os dados do atuador. Por exemplo, você pode definir um atuador para representar o ar quente. O atuador recebe novos dados quando você liga ou desliga o ar quente.

## Estrutura personalizada

Uma estrutura personalizada (também conhecida como struct) representa uma estrutura de dados complexa ou de ordem superior. Ela facilita o agrupamento ou a vinculação lógica de dados originários da mesma fonte. Uma struct é usada quando os dados são lidos ou gravados em uma operação atômica, como para representar um tipo de dados complexo ou uma forma de ordem superior.

Um sinal do tipo struct é definido no catálogo de sinais usando uma referência a um tipo de dados de struct em vez de um tipo de dados primitivo. As structs podem ser usadas para todos os tipos de sinais, incluindo sensores, atributos, atuadores e tipos de dados de sistemas de visão. Caso um sinal do tipo struct seja enviado ou recebido, o AWS IoT FleetWise espera que todos os itens incluídos tenham valores válidos, portanto, todos os itens são obrigatórios. Por exemplo, caso uma struct contenha os itens `Vehicle.Camera.Image.height`, `Vehicle.Camera.Image.width` e `Vehicle.Camera.Image.data`, espera-se que o sinal enviado contenha valores para todos esses itens.

### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Propriedade personalizada

Uma propriedade personalizada representa um membro da estrutura de dados complexa. O tipo de dados da propriedade pode ser primitivo ou outra struct.

Ao representar uma forma de ordem superior usando uma struct e uma propriedade personalizada, a forma de ordem superior pretendida é sempre definida e visualizada como uma estrutura em árvore. A propriedade personalizada é usada para definir todos os nós da folha, enquanto a struct é usada para definir todos os nós que não são da folha.

## Catálogo de sinais

Um catálogo de sinais contém uma coleção de sinais. Os sinais em um catálogo de sinais podem ser usados para modelar veículos que usam protocolos e formatos de dados diferentes. Por exemplo, existem dois carros fabricados por montadoras diferentes: um usa o protocolo Control Area Network (barramento CAN); o outro usa o protocolo On-board Diagnostics (OBD). Você pode definir um sensor no catálogo de sinais para receber valores de temperatura no veículo. Esse sensor pode ser usado para representar os termopares nos dois carros. Para obter mais informações, consulte [Criar e gerenciar catálogos de sinais](#).

## Modelo do veículo (manifesto do modelo)

Modelos de veículos são estruturas declarativas que você pode usar para padronizar o formato dos veículos e definir relações entre os sinais nos veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo. Você adiciona sinais para criar modelos de veículos. Para obter mais informações, consulte [Criar e gerenciar modelos de veículos](#).

## Manifesto do decodificador

Os manifestos do decodificador contêm informações de decodificação para cada sinal nos modelos de veículos. Sensores e atuadores em veículos transmitem mensagens de baixo nível (dados binários). Com manifestos de decodificadores, o AWS IoT FleetWise consegue transformar dados binários em valores legíveis por humanos. Cada manifesto do decodificador está associado a um modelo de veículo. Para obter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

## Interface de rede

Contém informações sobre o protocolo usado pela rede do veículo. O AWS IoT FleetWise é compatível com os seguintes protocolos.

### Controller Area Network (barramento CAN)

Um protocolo que define como os dados são comunicados entre unidades de controle eletrônico (ECUs). As ECUs podem ser a unidade de controle do motor, airbags ou sistema de áudio.

### On-board diagnostic (OBD) II

Um protocolo mais desenvolvido que define como os dados de autodiagnóstico são comunicados entre as ECUs. Ele fornece uma série de códigos de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado no veículo.

## Middleware de veículo

O middleware de veículo definido como um tipo de interface de rede. Exemplos de middleware de veículo incluem Robot Operating System (ROS 2) e Scalable Service-Oriented Middleware over IP (SOME/IP).

### Note

O AWS IoT FleetWise é compatível com o middleware ROS 2 para dados de sistemas de visão.

## Sinal do decodificador

Fornecer informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um sinal de decodificador. Se o manifesto do decodificador contiver interfaces de rede CAN, ele deverá conter sinais do decodificador CAN. Se o manifesto do decodificador contiver interfaces de rede OBD, ele deverá conter sinais do decodificador OBD.

O manifesto do decodificador deverá conter sinais do decodificador de mensagens se também contiver interfaces de middleware do veículo.

## Veículo

Uma representação virtual do veículo físico, como um carro ou caminhão. Veículos são instâncias de modelos de veículos. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Cada veículo corresponde a uma coisa da AWS IoT.

## Frota

Uma frota representa um grupo de veículos. Antes de poder gerenciar facilmente uma frota de veículos, você deve associar veículos individuais a uma frota.

## Campanha

Contém esquemas de coleta de dados. Você define uma campanha na nuvem e a implanta em um veículo ou frota. As campanhas fornecem instruções ao software Edge Agent sobre como selecionar, coletar e transferir dados para a nuvem.

## Esquema de coleta de dados

Os esquemas de coleta de dados fornecem instruções ao software Edge Agent sobre como coletar dados. Atualmente, o AWS IoT FleetWise é compatível com o esquema de coleta baseado em condições e o esquema de coleta baseado em tempo.

### Esquema de coleta baseado em condições

Use uma expressão lógica para reconhecer quais dados coletar. O software Edge Agent coleta dados quando a condição é atendida. Por exemplo, se a expressão for `$variable.myVehicle.InVehicleTemperature >35.0`, o software Edge Agent coletará valores de temperatura maiores que 35,0.

### Esquema de coleta baseado no tempo

Especifique um período em milissegundos para definir a frequência de coleta de dados. Por exemplo, se o período for de 10.000 milissegundos, o software Edge Agent coletará dados uma vez a cada 10 segundos.

## Recursos do AWS IoT FleetWise

Veja a seguir os principais recursos do AWS IoT FleetWise.

### Modelagem de veículos

Crie representações virtuais dos veículos e aplique um formato comum para organizar os sinais dos veículos. O AWS IoT FleetWise é compatível com a [Vehicle Signal Specification \(VSS\)](#), que você pode usar para padronizar os sinais de veículos.

### Coleta de dados baseada em esquema

Defina esquemas para transferir apenas dados de veículos de alto valor para a nuvem. Você pode definir esquemas baseados em condições para controlar quais dados coletar, como dados de valores de temperatura no veículo superiores a 40 graus. Você também pode definir esquemas baseados em tempo para controlar a frequência de coleta de dados.

### Software Edge Agent para AWS IoT FleetWise

O software Edge Agent executado em veículos facilita a comunicação entre veículos e a nuvem. Enquanto os veículos estão conectados à nuvem, o software Edge Agent recebe continuamente esquemas de coleta de dados e os coleta adequadamente.

## Serviços relacionados

O AWS IoT FleetWise integra-se aos seguintes serviços da AWS para melhorar a disponibilidade e a escalabilidade das soluções em nuvem.

- **AWS IoT Core** — Registre e controle dispositivos AWS IoT que carregam dados do veículo no AWS IoT FleetWise. Para obter mais informações, consulte [O que é o AWS IoT?](#) no Guia do desenvolvedor do AWS IoT.
- **Amazon Timestream** — Use um banco de dados de séries temporais para armazenar e analisar os dados do veículo. Para obter mais informações, consulte [O que é o Amazon Timestream?](#) no Guia do desenvolvedor do Amazon Timestream.
- **Amazon S3** — Use um serviço de armazenamento de objetos para armazenar e gerenciar os dados do veículo. Para obter mais informações, consulte [O que é o Amazon S3?](#) no Guia do usuário do Amazon Simple Storage Service.

# Configurando a AWS IoT FleetWise

Antes de usar a AWS IoT FleetWise pela primeira vez, conclua as etapas nas seções a seguir.

Tópicos

- [Configure seu Conta da AWS](#)
- [Comece pelo console](#)
- [Configuração de definições](#)

## Configure seu Conta da AWS

Conclua as tarefas a seguir para se inscrever AWS e criar um usuário administrativo.

## Inscreva-se para um Conta da AWS

Se você não tiver um Conta da AWS, conclua as etapas a seguir para criar um.

Para se inscrever em um Conta da AWS

1. Abra <https://portal.aws.amazon.com/billing/signup>.
2. Siga as instruções on-line.

Parte do procedimento de inscrição envolve receber uma chamada telefônica e digitar um código de verificação no teclado do telefone.

Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWS é criado. O usuário-raiz tem acesso a todos os Serviços da AWS e recursos na conta. Como prática recomendada de segurança, atribua o acesso administrativo a um usuário e use somente o usuário-raiz para executar [tarefas que exigem acesso de usuário-raiz](#).

AWS envia um e-mail de confirmação após a conclusão do processo de inscrição. A qualquer momento, é possível visualizar as atividades da conta atual e gerenciar sua conta acessando <https://aws.amazon.com/> e selecionando Minha conta.



## Criar um usuário com acesso administrativo

Depois de se inscrever em um Conta da AWS, proteja seu Usuário raiz da conta da AWS AWS IAM Identity Center, habilite e crie um usuário administrativo para que você não use o usuário root nas tarefas diárias.

### Proteja seu Usuário raiz da conta da AWS

1. Faça login [AWS Management Console](#) como proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, digite sua senha.

Para obter ajuda ao fazer login usando o usuário-raiz, consulte [Signing in as the root user](#) (Fazer login como usuário-raiz) no Guia do usuário do Início de Sessão da AWS .

2. Habilite a autenticação multifator (MFA) para o usuário-raiz.

Para obter instruções, consulte [Habilitar um dispositivo de MFA virtual para seu usuário Conta da AWS raiz \(console\) no Guia](#) do usuário do IAM.

### Criar um usuário com acesso administrativo

1. Habilitar o IAM Identity Center.

Para obter instruções, consulte [Habilitar AWS IAM Identity Center](#) no Guia do usuário do AWS IAM Identity Center .

2. No Centro de Identidade do IAM, conceda o acesso administrativo para um usuário.

Para ver um tutorial sobre como usar o Diretório do Centro de Identidade do IAM como fonte de identidade, consulte [Configurar o acesso do usuário com o padrão Diretório do Centro de Identidade do IAM](#) no Guia AWS IAM Identity Center do usuário.

### Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com seu usuário do Centro de Identidade do IAM, use a URL de login que foi enviada ao seu endereço de e-mail quando você criou o usuário do Centro do Usuário do IAM.

Para obter ajuda para fazer login usando um usuário do IAM Identity Center, consulte [Como fazer login no portal de AWS acesso](#) no Guia Início de Sessão da AWS do usuário.

## Atribuir acesso a usuários adicionais

1. No Centro de Identidade do IAM, crie um conjunto de permissões que siga as práticas recomendadas de aplicação de permissões com privilégio mínimo.

Para obter instruções, consulte [Create a permission set](#) no Guia do usuário do AWS IAM Identity Center .

2. Atribua usuários a um grupo e, em seguida, atribua o acesso de autenticação única ao grupo.

Para obter instruções, consulte [Add groups](#) no Guia do usuário do AWS IAM Identity Center .

### Note

Você pode usar uma função vinculada a serviços com a IoT AWS . FleetWise As funções vinculadas ao serviço são predefinidas pela AWS IoT FleetWise e incluem as permissões que a IoT AWS precisa para enviar métricas para a Amazon FleetWise . CloudWatch Para ter mais informações, consulte [Usando funções vinculadas a serviços para o AWS IoT FleetWise](#).

## Comece pelo console

Se você ainda não estiver conectado ao seu Conta da AWS, entre e abra o console de [AWS IoT FleetWise](#) . Para começar a usar a AWS IoT FleetWise, crie um modelo de veículo. Um modelo de veículo padroniza o formato de seus veículos.

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. Em Começar com AWS IoT FleetWise, escolha Começar.

Para obter mais informações sobre a criação de modelo de veículo, consulte [Criar um modelo de veículo \(console\)](#).

## Configuração de definições

Você pode usar o FleetWise console de AWS IoT ou a API para definir configurações para métricas do Amazon CloudWatch Logs, Amazon CloudWatch Logs e criptografar dados com um. Chave gerenciada pela AWS

Com CloudWatch métricas, você pode monitorar a AWS IoT FleetWise e outros AWS recursos. Você pode usar CloudWatch métricas para coletar e monitorar métricas, como para determinar se há um limite de serviço excedido. Para obter mais informações sobre CloudWatch métricas, consulte [Monitoramento do AWS IoT FleetWise com Amazon CloudWatch](#).

Com o CloudWatch Logs, a AWS IoT FleetWise envia dados de registro para um grupo de CloudWatch registros, onde você pode usá-los para identificar e mitigar quaisquer problemas. Para obter mais informações sobre CloudWatch registros, consulte [Configurar o registro em log do AWS IoT FleetWise](#).

Com a criptografia de dados, a AWS IoT FleetWise usa para Chaves gerenciadas pela AWS criptografar dados. Você também pode optar por criar e gerenciar chaves com AWS KMS. Para obter mais informações sobre criptografia, consulte [Criptografia de dados](#).

## Definir configurações (console)

Se você ainda não estiver conectado ao seu Conta da AWS, entre e abra o console de [AWS IoT FleetWise](#).

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel esquerdo, selecione Configurações.
3. Em Métricas, escolha Ativar. A AWS IoT associa FleetWise automaticamente uma política CloudWatch gerenciada à função vinculada ao serviço e ativa métricas. CloudWatch
4. Em Registro, selecione Editar.
  - a. Na seção de CloudWatch registro, insira o grupo Registro.
  - b. Selecione Enviar para salvar as alterações.
5. Na seção Criptografia, selecione Editar.
  - a. Selecione o tipo de chave a ser usada. Para ter mais informações, consulte [Gerenciamento de chaves](#).
    - i. Use a AWS chave — a AWS IoT FleetWise possui e gerencia a chave.
    - ii. Escolha uma AWS Key Management Service chave diferente — Você gerencia as AWS KMS keys que estão na sua conta.
  - b. Selecione Enviar para salvar as alterações.

## Definir configurações (AWS CLI)

No AWS CLI, registre a conta para definir as configurações.

1. Para definir as configurações, execute o seguinte comando.

```
aws iotfleetwise register-account
```

2. Para verificar suas configurações, execute o comando a seguir para recuperar o status do registro.

### Note

A função vinculada ao serviço só é usada para publicar métricas de AWS FleetWise IoT em CloudWatch. Para ter mais informações, consulte [Usando funções vinculadas a serviços para o AWS IoT FleetWise](#).

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

### Example Retorno

```
{
  "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/AWSIoT FleetwiseServiceRole"
  },
  "lastModificationTime": "2022-07-28T11:31:22.854000-07:00",
}
```

O status de registro pode ser um dos seguintes:

- **REGISTRATION\_SUCCESS**— O AWS recurso foi registrado com sucesso.

- **REGISTRATION\_PENDING**— AWS A IoT FleetWise está processando a solicitação de registro. Esse processo leva cerca de 5 minutos para ser concluído.
- **REGISTRATION\_FAILURE**— AWS A IoT não FleetWise consegue registrar o AWS recurso. Tente novamente mais tarde.

# Começando com a AWS IoT FleetWise

Com a AWS IoT FleetWise, você pode coletar, transformar e transferir os dados do seu veículo. Use os tutoriais desta seção para começar a usar a IoT AWS . FleetWise

Consulte os tópicos a seguir para saber mais sobre a AWS IoT: FleetWise

- [Ingestão de dados na nuvem](#)
- [Modelagem de veículos](#)
- [Criar, provisionar e gerenciar veículos](#)
- [Criar e gerenciar frotas](#)
- [Coletar e transferir dados com campanhas](#)

## Requisitos

Você deve ter um Conta da AWS para começar a usar a AWS IoT FleetWise. Se você não tiver uma, consulte [Configurando a AWS IoT FleetWise](#).

Use uma região em que a AWS IoT FleetWise esteja disponível. Para obter mais informações, consulte [FleetWise endpoints e cotas de AWS IoT](#). Você pode usar o seletor de região no AWS Management Console para alternar para uma dessas regiões.

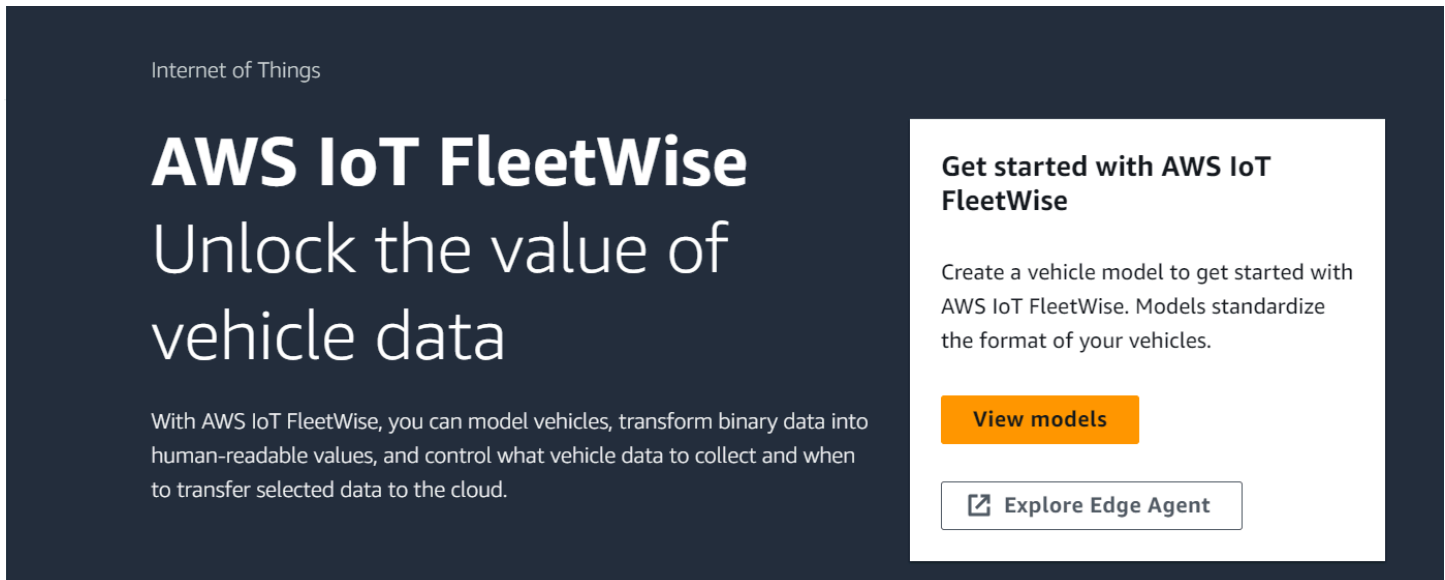
## Demo do software Edge Agent

Você pode usar a demonstração de início rápido do Explore Edge Agent para explorar a AWS IoT FleetWise e aprender a desenvolver o software Edge Agent para AWS IoT. FleetWise Esta demonstração usa um AWS CloudFormation modelo. Ele orienta você na análise da implementação de referência do Edge Agent, no desenvolvimento do Edge Agent e, em seguida, na implantação do software Edge Agent em um Amazon EC2 Graviton e na geração de dados de amostra do veículo. A demonstração também fornece um script que você pode usar para criar um catálogo de sinais, um modelo de veículo, um manifesto de decodificador, um veículo, uma frota e uma campanha, tudo na nuvem. Para ter mais informações sobre a demonstração de início rápido, faça o seguinte para baixar o Guia do desenvolvedor do software Edge Agent.

Para baixar a demonstração de início rápido

1. Navegue até o console [AWS de IoT. FleetWise](#)

2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.



Internet of Things

# AWS IoT FleetWise

## Unlock the value of vehicle data

With AWS IoT FleetWise, you can model vehicles, transform binary data into human-readable values, and control what vehicle data to collect and when to transfer selected data to the cloud.

**Get started with AWS IoT FleetWise**

Create a vehicle model to get started with AWS IoT FleetWise. Models standardize the format of your vehicles.

[View models](#)

[Explore Edge Agent](#)

## Tutorial: Introdução à AWS IoT FleetWise (console)

Use FleetWise a AWS IoT para coletar, transformar e transferir o formato de dados exclusivo de veículos automatizados para a nuvem quase em tempo real. Você tem acesso a informações de toda a frota. Isso pode ajudar a detectar e amenizar com eficiência os problemas de integridade do veículo, transferir sinais de dados de alto valor e diagnosticar problemas remotamente, tudo isso com redução de custos.

Este tutorial mostra como começar a usar a AWS IoT FleetWise. Você aprenderá como criar um modelo de veículo (manifesto de modelo), um manifesto de decodificador, um veículo e uma campanha.

Para obter mais informações sobre os principais componentes e conceitos da AWS IoT FleetWise, consulte. [Como funciona o AWS IoT FleetWise](#)

Tempo estimado: 45 minutos.

### Important

Você será cobrado pelos FleetWise recursos de AWS IoT que essa demonstração cria e consome. Para obter mais informações, consulte [AWS IoT FleetWise](#) na página de preços de AWS FleetWise IoT.

## Tópicos

- [Pré-requisitos](#)
- [Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise](#)
- [Etapa 2: criar um modelo de veículo](#)
- [Etapa 3: criar um manifesto do decodificador](#)
- [Etapa 4: configurar um manifesto do decodificador](#)
- [Etapa 5: criar um veículo](#)
- [Etapa 6: criar uma campanha](#)
- [Etapa 7: limpar](#)
- [Próximas etapas](#)

## Pré-requisitos

Para concluir este tutorial de conceitos básicos, você precisa:

- Um Conta da AWS. Se você não tiver um Conta da AWS, consulte [Criação de um Conta da AWS](#) no Guia de AWS Account Management referência.
- Acesso a um Região da AWS que ofereça suporte à AWS IoT FleetWise. Atualmente, a AWS IoT FleetWise é suportada no Leste dos EUA (Norte da Virgínia) e na Europa (Frankfurt).
- Recursos do Amazon Timestream:
  - Um banco de dados do Amazon Timestream. Para obter mais informações, consulte [Criar um banco de dados](#) no Guia do desenvolvedor do Amazon Timestream.
  - Uma tabela do Amazon Timestream criada no Amazon Timestream que armazenará seus dados. Para obter mais informações, consulte [Criar uma tabela](#) no Guia do desenvolvedor do Amazon Timestream.



## Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise

### Note

A CloudFormation pilha nesta etapa usa dados de telemetria. Você também pode criar uma CloudFormation pilha usando dados do sistema de visão. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Seu software Edge Agent para AWS IoT FleetWise facilita a comunicação entre veículos e a nuvem. Ele recebe instruções de esquemas de coleta de dados sobre como coletar dados de veículos conectados à nuvem.

Para configurar o software Edge Agent, em Informações gerais, faça o seguinte:

1. Abra o [CloudFormation modelo de lançamento](#).
2. Na página de criação rápida da pilha, em Nome da pilha, insira o nome da sua pilha de recursos de IoT AWS . FleetWise Uma pilha é um nome amigável que aparece como prefixo nos nomes dos recursos que esse AWS CloudFormation modelo cria.
3. Em Parâmetros, insira os valores personalizados para os parâmetros relacionados à pilha.
  - a. Fleetsize: você pode aumentar o número de veículos na frota atualizando o parâmetro Fleetsize.
  - b. IoT CoreRegion - Você pode especificar a região em AWS IoT que a coisa é criada atualizando o parâmetro IoT. CoreRegion Você deve usar a mesma região que usou para criar seus veículos de AWS IoT FleetWise . Para obter mais informações sobre Regiões da AWS, consulte [Regiões e zonas - Amazon Elastic Compute Cloud](#).
4. Na seção Capacidades, selecione a caixa para confirmar que AWS CloudFormation cria recursos do IAM.
5. Selecione Criar pilha e aguarde aproximadamente 15 minutos para que o status da pilha mostre CREATE\_COMPLETE.
6. Para confirmar que a pilha foi criada, selecione a guia Informações da pilha, atualize a visualização e procure CREATE\_COMPLETE.

fwdemo

**Overview**

Stack ID	Description
----------	-------------

<a href="#">arn:aws:cloudformation:us-east-</a>	-
-------------------------------------------------	---

<a href="#">1:012345678912:stack/fwdemo/bd04af20-a269-11ed-bf1d-0a56266679b7</a>	
----------------------------------------------------------------------------------	--

Status	Status reason
--------	---------------

CREATE_COMPLETE	-
-----------------	---

**⚠ Important**

Você será cobrado pelos FleetWise recursos de AWS IoT que essa demonstração cria e consome. Para obter mais informações, consulte [AWS IoT FleetWise](#) na página de preços de AWS FleetWise IoT.

## Etapa 2: criar um modelo de veículo

**⚠ Important**


Você não pode criar um modelo de veículo com sinais de dados do sistema de visão no console de AWS IoT FleetWise . Use a AWS CLI em vez disso.

Você usa modelos de veículos para padronizar o formato dos veículos e para ajudar a definir a relação entre os sinais nos veículos que você cria. Um catálogo de sinais também é criado quando você cria um modelo de veículo. Um catálogo de sinais é uma coleção de sinais padronizados que podem ser reutilizados para criar modelos de veículos. Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. No momento, o FleetWise serviço de AWS IoT oferece suporte a apenas um catálogo de sinais Região da AWS por conta. Isso ajuda a verificar se os dados processados de uma frota de veículos são consistentes.

Para criar um modelo de veículo

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Modelos de veículos.

3. Na página Modelos de veículos, selecione Criar modelo de veículo.
4. Na seção Informações gerais, insira o nome do modelo do seu veículo, como Veículo1, e uma descrição opcional. Em seguida, escolha Próximo.
5. Escolha um ou mais sinais do catálogo de sinais. Você pode filtrar sinais por nome no catálogo de pesquisa ou escolhê-los na lista. Por exemplo, você pode escolher sinais de pressão dos pneus e pressão dos freios para poder coletar dados relacionados a esses sinais. Escolha Próximo.
6. Escolha seus arquivos .dbc e carregue-os do seu dispositivo local. Escolha Próximo.

 Note

Para este tutorial, você pode baixar um [exemplo de arquivo.dbc](#) para fazer o upload nesta etapa.

7. Adicione atributos ao modelo do seu veículo e selecione Avançar.
  - a. Nome — Insira o nome do atributo do veículo, como o nome do fabricante ou a data de fabricação.
  - b. Tipo de dados — No menu Tipo de dados, selecione um tipo de dados.
  - c. Unidade — (Opcional) Insira um valor unitário, como quilômetro ou Celsius.
  - d. Caminho — (Opcional) Insira um nome para o caminho para um sinal, como Vehicle.Engine.Light.. O ponto (.) indica que é um sinal secundário.
  - e. Valor padrão — (Opcional) Insira um valor padrão.
  - f. Descrição — (Opcional) Insira uma descrição do atributo.
8. Analise as configurações. Quando estiver pronto, escolha Create (Criar). Uma notificação aparece informando que o modelo do seu veículo foi criado com sucesso.

✔ **Vehicle model created**  
You successfully created the vehicle model: demo. ✕

AWS IoT FleetWise > Vehicle models > Demo

## demo

[Duplicate](#) [Create vehicle](#) [Create decoder manifest](#)

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

### Summary [Info](#)

Vehicle model ARN arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo	Status ✔ ACTIVE	Date created February 01, 2023 at 14:40 (UTC-05)
Signal catalog ARN arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog	Description -	Last modified February 01, 2023 at 14:40 (UTC-05)

## Etapa 3: criar um manifesto do decodificador

Os manifestos do decodificador estão associados aos modelos de veículos que você cria. Eles contêm informações que ajudam a AWS IoT a FleetWise decodificar e transformar dados de veículos de um formato binário em valores legíveis por humanos que podem ser analisados. As interfaces de rede e sinais de decodificador são componentes que ajudam a configurar manifestos de decodificador. Uma interface de rede contém informações sobre o protocolo CAN ou OBD usado pela rede do seu veículo. O sinal do decodificador fornece informações de decodificação para um sinal específico.

Para criar um manifesto do decodificador

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Modelos de veículos.
3. Na seção Modelos de veículos, selecione o modelo de veículo que você quer usar para criar um manifesto do decodificador.
4. Selecione Criar manifesto do decodificador.

## Etapa 4: configurar um manifesto do decodificador

Para configurar um manifesto do decodificador

### Important

Você não pode configurar sinais de dados do sistema de visão em manifestos de decodificadores usando o console de IoT AWS . FleetWise Use a AWS CLI em vez disso. Para ter mais informações, consulte [Criar um manifesto do decodificador \(AWS CLI\)](#).

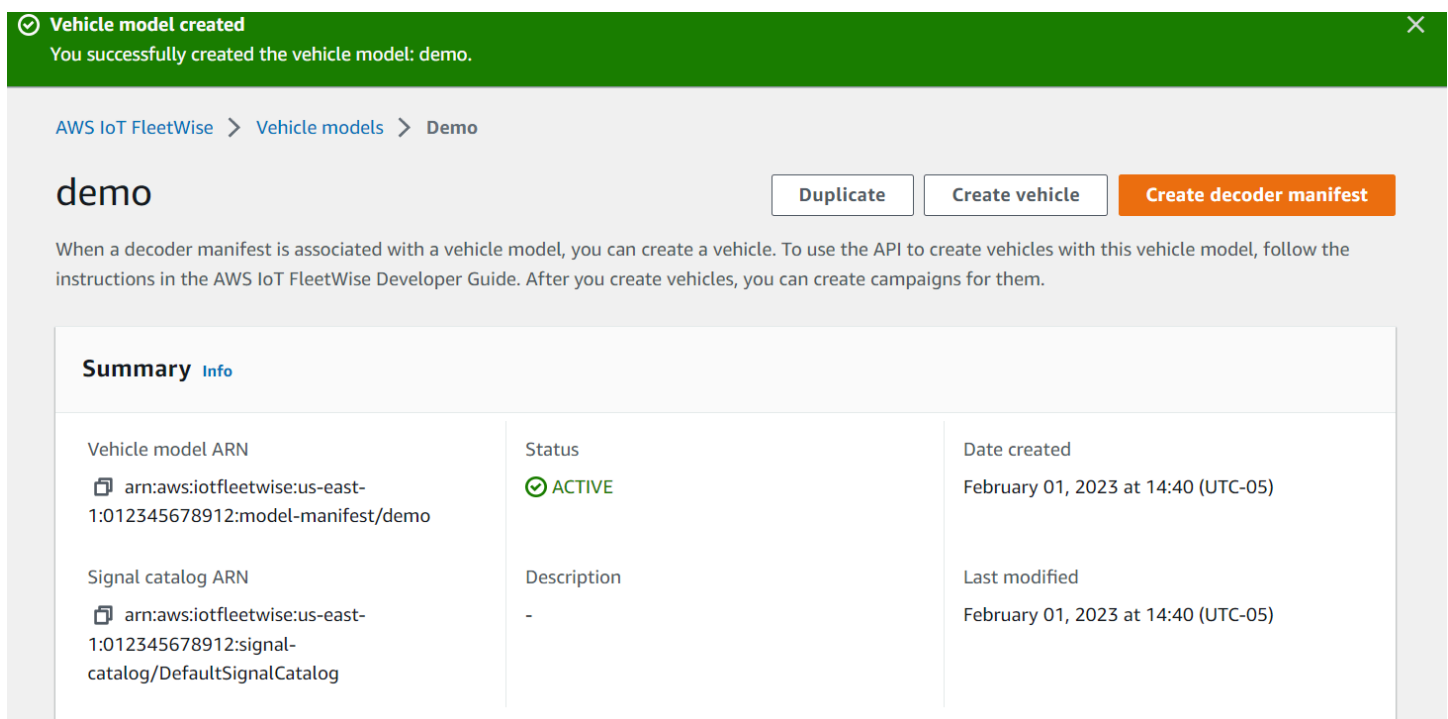
1. Para ajudar a identificar o manifesto do decodificador, insira um nome e uma descrição opcional para ele. Em seguida, clique em Próximo.
2. Para adicionar uma ou mais interfaces de rede, escolha o tipo CAN\_INTERFACE ou OBD\_INTERFACE.
  - Interface de diagnóstico integrado (OBD) — Selecione este tipo de interface para ter um protocolo que defina como os dados de autodiagnóstico são comunicados entre unidades de controle eletrônico (ECUs). Este protocolo fornece vários códigos de problemas de diagnóstico (DTCs) padrão que podem ajudar a solucionar problemas no veículo.
  - Interface de rede de área de controlador (barramento CAN) — Selecione este tipo de interface para ter um protocolo que defina como os dados são comunicados entre as ECUs. As ECUs podem ser unidades de controle do motor, airbags ou sistema de áudio.
3. Insira um nome de interface de rede.
4. Para adicionar sinais à interface de rede, escolha um ou mais sinais da lista.
5. Selecione um sinal de decodificador para o sinal adicionado na etapa anterior. Para fornecer informações de decodificação, faça upload de um arquivo .dbc. Cada sinal no modelo do veículo deve ser emparelhado com um sinal de decodificador, que você pode escolher na lista.
6. Para adicionar uma interface de rede secundária, selecione Adicionar interface de rede. Quando terminar de adicionar interfaces de rede, selecione Avançar.
7. Analise as configurações e selecione Criar. É exibida uma notificação informando que o manifesto do decodificador foi criado com sucesso.

## Etapa 5: criar um veículo

Na AWS IoT FleetWise, os veículos são representações virtuais do seu veículo físico real. Todos os veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais, e cada veículo criado corresponde a um item de IoT recém-criado. Você deve associar todos os veículos a um manifesto do decodificador.

### Pré-requisitos

1. Verifique se você já criou o modelo do veículo e o manifesto do decodificador. Além disso, verifique se o status do modelo do veículo está **ATIVO**.
  - a. Para verificar se o status do modelo do veículo está **ATIVO**, abra o console de AWS IoT FleetWise .
  - b. No painel de navegação, selecione Modelos de veículos.
  - c. Na seção Resumo, em Status, verifique o status do seu veículo.






**Vehicle model created**  
You successfully created the vehicle model: demo.

AWS IoT FleetWise > Vehicle models > Demo

### demo

[Duplicate](#) [Create vehicle](#) [Create decoder manifest](#)

When a decoder manifest is associated with a vehicle model, you can create a vehicle. To use the API to create vehicles with this vehicle model, follow the instructions in the AWS IoT FleetWise Developer Guide. After you create vehicles, you can create campaigns for them.

Summary <a href="#">Info</a>		
Vehicle model ARN  arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo	Status  <b>ACTIVE</b>	Date created February 01, 2023 at 14:40 (UTC-05)
Signal catalog ARN  arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog	Description -	Last modified February 01, 2023 at 14:40 (UTC-05)

### Para criar um veículo

1. Abra o FleetWise console da AWS.
2. No painel de navegação, selecione Veículos.

3. Selecione Criar veículo.
4. Para definir as propriedades do veículo, insira o nome do veículo e selecione um manifesto do modelo (modelo do veículo) e um manifesto do decodificador.
5. (Opcional) Para definir os atributos do veículo, insira um par de valores-chave e selecione Adicionar atributos.
6. (Opcional) Para rotular o recurso da AWS, adicione tags e selecione Adicionar nova tag.
7. Escolha Próximo.
8. Para configurar o certificado do veículo, você pode fazer upload do próprio certificado ou selecionar Gerar automaticamente um novo certificado. Recomendamos gerar automaticamente o certificado para uma configuração mais rápida. Se já tiver um certificado, será possível escolher usá-lo.
9. Faça download dos arquivos de chave pública e privada e selecione Avançar.
10. Para anexar uma política ao certificado do veículo, você pode inserir um nome de política existente ou criar uma nova política. Para criar uma nova política, selecione Criar política > Avançar.
11. Analise as configurações. Após concluir, selecione Criar veículo.

## Etapa 6: criar uma campanha

Na AWS IoT FleetWise, as campanhas são usadas para facilitar a seleção, coleta e transferência de dados dos veículos para a nuvem. As campanhas contêm esquemas de coleta de dados que fornecem instruções ao software Edge Agent sobre como coletar dados com um esquema de coleta baseado em condições ou um esquema de coleta baseado em tempo.

### Como criar uma campanha

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Campanhas.
3. Escolha Create campaign (Criar campanha).
4. Insira o nome da sua campanha e uma descrição opcional.
5. Para configurar o esquema de coleta de dados da campanha, você pode definir manualmente o esquema de coleta de dados ou fazer upload de um arquivo .json por seu dispositivo local. O upload de um arquivo .json define automaticamente o esquema de coleta de dados.

- a. Para definir manualmente o esquema de coleta de dados, selecione Definir esquema de coleta de dados e selecione o tipo de esquema de coleta de dados que você quer usar para sua campanha. Você pode escolher um esquema de coleta baseado em condições ou um esquema de coleta baseado em tempo.
  - b. Se você escolher um esquema de coleta baseado em tempo, deverá especificar por quanto tempo sua campanha coletará os dados do veículo.
  - c. Se escolher um esquema de coleta baseado em condições, será necessário especificar uma expressão para reconhecer quais dados devem ser coletados. Verifique se o nome do sinal está especificado como uma variável, um operador de comparação e um valor de comparação.
  - d. (Opcional) Escolha a versão do idioma da sua expressão ou mantenha-a como o valor padrão de 1.
  - e. (Opcional) Especifique o intervalo de acionamento entre dois eventos de coleta de dados.
  - f. Para coletar dados, selecione a condição do modo Acionador para o software Edge Agent. Por padrão, o FleetWise software Edge Agent para AWS IoT sempre coleta dados sempre que a condição é atendida. Ou poderá coletar dados somente quando a condição for atendida pela primeira vez, no primeiro acionamento.
  - g. (Opcional) Você pode escolher opções de esquema mais avançadas.
6. Para especificar os sinais dos quais o esquema de coleta de dados coletará dados, procure o nome do sinal no menu.
  7. (Opcional) Você pode escolher uma contagem máxima de amostras ou um intervalo mínimo de amostragem. Você também pode adicionar mais sinais.
  8. Escolha Próximo.
  9. Defina o destino de armazenamento para o qual a campanha deve transferir os dados. Você pode armazenar dados no Amazon S3 ou no Amazon Timestream.
    - a. Amazon S3 — Escolha o bucket S3 que AWS IoT FleetWise tem permissões para.
    - b. Amazon Timestream — escolha o banco de dados Timestream e o nome da tabela. Insira uma função do IAM que permita enviar dados AWS IoT FleetWise para o Timestream.
  10. Escolha Próximo.
  11. Selecione atributos ou nomes de veículos na caixa de pesquisa.
  12. Insira o valor relacionado ao atributo ou nome que você escolheu para seu veículo.
  13. Selecione os veículos dos quais sua campanha coletará dados. Em seguida, clique em Próximo.



14. Analise as configurações da sua campanha e selecione Criar campanha. Você ou sua equipe devem implantar a campanha em veículos.

## Etapa 7: limpar

Para evitar cobranças adicionais pelos recursos usados durante este tutorial, exclua a AWS CloudFormation pilha e todos os recursos da pilha.

Para excluir a AWS CloudFormation pilha

1. Abra o [console de AWS CloudFormation](#).
2. Na lista de pilhas, escolha a pilha que você criou na etapa 1.
3. Escolha Delete.
4. Para confirmar a exclusão, selecione Delete (Excluir). A pilha leva cerca de 15 minutos para ser excluída.

## Próximas etapas

1. Você pode processar e visualizar os dados do veículo que sua campanha coleta. Para ter mais informações, consulte [Processamento e visualização de dados do veículo](#).
2. Você pode solucionar e resolver problemas com a AWS IoT. FleetWise Para ter mais informações, consulte [Solução de problemas do AWS IoT FleetWise](#).

# Ingestão de dados na nuvem

O software Edge Agent do AWS IoT FleetWise, quando instalado e executado em veículos, foi projetado para facilitar a comunicação segura entre os veículos e a nuvem.

## Note

- O AWS IoT FleetWise não se destina ao uso ou em associação com a operação de quaisquer ambientes perigosos ou sistemas críticos que possam causar lesões corporais graves ou morte ou causar danos ambientais ou materiais. Os dados do veículo coletados por meio do uso do AWS IoT FleetWise são apenas para fins informativos, e você não pode usar o AWS IoT FleetWise para controlar ou operar funções do veículo.
- Os dados do veículo coletados por meio do uso do AWS IoT FleetWise devem ser avaliados quanto à precisão, conforme apropriado para o caso de uso, inclusive para fins de cumprimento de quaisquer obrigações de conformidade que você possa ter de acordo com os regulamentos de segurança do veículo aplicáveis (como monitoramento de segurança e obrigações de relatórios). Essa avaliação deve incluir a recolha e análise de informações por meio de outros meios e fontes padrão do setor (tais como relatórios de condutores de veículos).

Para ingerir dados na nuvem, faça o seguinte:

1. Desenvolva e instale o software Edge Agent do AWS IoT FleetWise no veículo. Para obter mais informações sobre como trabalhar com o software Edge Agent, faça o seguinte para baixar o [Guia do desenvolvedor do software Edge Agent do AWS IoT FleetWise](#).
  1. Acesse o [console do AWS IoT FleetWise](#).
  2. Na página inicial do serviço, na seção Comece a usar o AWS IoT FleetWise, selecione Explorar o Edge Agent.
2. Crie ou importe um catálogo de sinais contendo sinais que você usará para criar um modelo de veículo. Para obter mais informações, consulte [Criar um catálogo de sinais \(AWS CLI\)](#) e [Importar um catálogo de sinais \(AWS CLI\)](#).

**Note**

- Se você usar o console do AWS IoT FleetWise para criar o primeiro modelo de veículo, não precisará criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, o AWS IoT FleetWise cria automaticamente um catálogo de sinais para você. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Atualmente, o AWS IoT FleetWise é compatível com um catálogo de sinais para cada conta da AWS por Região da AWS.

3. Use sinais no catálogo de sinais para criar um modelo de veículo. Para obter mais informações, consulte [Criar um modelo de veículo](#).

**Note**

- Se você usar o console do AWS IoT FleetWise para criar um modelo de veículo, poderá fazer upload de arquivos .dbc para importar sinais. .dbc é um formato de arquivo compatível com os bancos de dados Controller Area Network (barramento CAN). Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para obter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar a operação da API `CreateModelManifest` para criar um modelo de veículo, deverá usar a operação da API `UpdateModelManifest` para ativar o modelo de veículo. Para obter mais informações, consulte [Atualizar o modelo de um veículo \(AWS CLI\)](#).
- Se você usar o console do AWS IoT FleetWise para criar um modelo de veículo, o AWS IoT FleetWise ativará automaticamente o modelo do veículo para você.

4. Crie um manifesto do decodificador. O manifesto do decodificador contém informações de decodificação para cada sinal especificado no modelo do veículo criado na etapa anterior. O manifesto do decodificador está associado ao modelo de veículo que você criou. Para obter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

**Note**

- Se você usar a operação da API `CreateDecoderManifest` para criar um manifesto do decodificador, deverá usar a operação da API `UpdateDecoderManifest` para

ativar o manifesto do decodificador. Para obter mais informações, consulte [Atualizar um manifesto do decodificador \(AWS CLI\)](#).

- Se você usar o console do AWS IoT FleetWise para criar um manifesto do decodificador, o AWS IoT FleetWise ativará automaticamente o manifesto do decodificador.

5. Crie veículos com base no modelo do veículo. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Você deve usar AWS IoT Core para provisionar seu veículo antes de poder ingerir dados na nuvem. Para obter mais informações, consulte [Criar, provisionar e gerenciar veículos](#).
6. (Opcional) Crie uma frota para representar um grupo de veículos e depois associe veículos individuais à frota. Isso ajuda você a gerenciar vários veículos ao mesmo tempo. Para obter mais informações, consulte [Criar e gerenciar frotas](#).
7. Crie campanhas. As campanhas são implantadas em um veículo ou frota de veículos. As campanhas fornecem instruções ao software Edge Agent sobre como selecionar, coletar e transferir dados para a nuvem. Para obter mais informações, consulte [Coletar e transferir dados com campanhas](#).

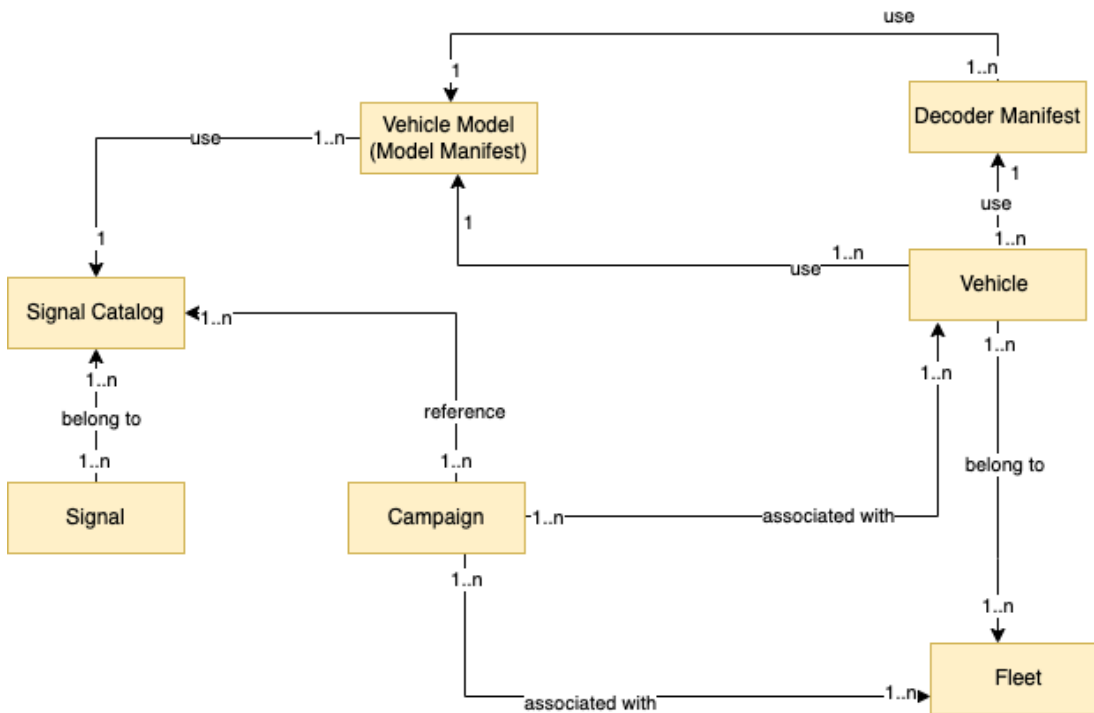
#### Note

Você deve usar a operação de API UpdateCampaign para aprovar a campanha, antes que o AWS IoT FleetWise possa implantá-la no veículo ou na frota. Para obter mais informações, consulte [Atualizar uma campanha \(AWS CLI\)](#).

O software Edge Agent transfere dados do veículo para AWS IoT Core usando o tópico reservado `$aws/iotfleetwise/vehicles/vehicleName/signals`, que envia dados para o AWS IoT FleetWise. O AWS IoT FleetWise então envia os dados para uma tabela Timestream ou bucket do Amazon S3. Você pode usar o Timestream para consultar os dados e usar o Amazon QuickSight ou o Grafana para visualizá-los. Para obter mais informações, consulte [Processamento e visualização de dados do veículo](#).

# Modelagem de veículos

AWS A IoT FleetWise fornece uma estrutura de modelagem de veículos que você pode usar para criar representações virtuais de seus veículos na nuvem. Sinais, catálogos de sinais, modelos de veículos e manifestos de decodificadores são os principais componentes com os quais você trabalha para modelar seus veículos.



## Signal

Sinais são estruturas fundamentais que você define para conter dados do veículo e seus metadados. Um sinal pode ser um atributo, uma ramificação, um sensor ou um atuador. Por exemplo, você pode criar um sensor para receber valores de temperatura no veículo e armazenar os metadados, incluindo um nome de sensor, um tipo de dados e uma unidade. Para ter mais informações, consulte [Criar e gerenciar catálogos de sinais](#).

## Catálogo de sinais

Um catálogo de sinais contém uma coleção de sinais. Os sinais em um catálogo de sinais podem ser usados para modelar veículos que usam protocolos e formatos de dados diferentes. Por exemplo, existem dois carros fabricados por montadoras diferentes: um usa o protocolo Control Area Network (barramento CAN); o outro usa o protocolo On-board Diagnostics (OBD). Você pode definir um sensor no catálogo de sinais para receber valores de temperatura no veículo.

Esse sensor pode ser usado para representar os termopares nos dois carros. Para ter mais informações, consulte [Criar e gerenciar catálogos de sinais](#).

## Modelo do veículo (manifesto do modelo)

Modelos de veículos são estruturas declarativas que você pode usar para padronizar o formato dos veículos e definir relações entre os sinais nos veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo. Você adiciona sinais para criar modelos de veículos. Para ter mais informações, consulte [Criar e gerenciar modelos de veículos](#).

## Manifesto do decodificador

Os manifestos do decodificador contêm informações de decodificação para cada sinal nos modelos de veículos. Sensores e atuadores em veículos transmitem mensagens de baixo nível (dados binários). Com os manifestos do decodificador, a AWS IoT FleetWise é capaz de transformar dados binários em valores legíveis por humanos. Cada manifesto do decodificador está associado a um modelo de veículo. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

Você pode usar o FleetWise console de AWS IoT ou a API para modelar veículos da seguinte maneira.

1. Crie ou importe um catálogo de sinais contendo sinais que você usará para criar um modelo de veículo. Para ter mais informações, consulte [Criar um catálogo de sinais \(AWS CLI\)](#) e [Importar um catálogo de sinais \(AWS CLI\)](#).

### Note

- Se você usa o FleetWise console de AWS IoT para criar o primeiro modelo de veículo, não precisa criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, a AWS IoT cria FleetWise automaticamente um catálogo de sinais para você. Para ter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- AWS FleetWise Atualmente, a IoT suporta um catálogo de sinais para cada AWS conta por. Região da AWS

2. Use sinais no catálogo de sinais para criar um modelo de veículo. Para ter mais informações, consulte [Criar um modelo de veículo](#).

**Note**

- Se você usar o FleetWise console de AWS IoT para criar um modelo de veículo, poderá carregar arquivos.dbc para importar sinais. .dbc é um formato de arquivo suportado pelos bancos de dados da Controller Area Network (barramento CAN). Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para ter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar a operação da API CreateModelManifest para criar um modelo de veículo, deverá usar a operação da API UpdateModelManifest para ativar o modelo de veículo. Para ter mais informações, consulte [Atualizar o modelo de um veículo \(AWS CLI\)](#).
- Se você usa o FleetWise console de AWS IoT para criar um modelo de veículo, a AWS IoT ativa FleetWise automaticamente o modelo de veículo para você.

3. Crie um manifesto do decodificador. O manifesto do decodificador contém informações de decodificação para cada sinal especificado no modelo do veículo criado na etapa anterior. O manifesto do decodificador está associado ao modelo de veículo que você criou. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

**Note**

- Se você usar a operação da API CreateDecoderManifest para criar um manifesto do decodificador, deverá usar a operação da API UpdateDecoderManifest para ativar o manifesto do decodificador. Para ter mais informações, consulte [Atualizar um manifesto do decodificador \(AWS CLI\)](#).
- Se você usa o FleetWise console de AWS IoT para criar um manifesto do decodificador, a AWS IoT ativa FleetWise automaticamente o manifesto do decodificador para você.

Os bancos de dados de barramento CAN são compatíveis com o formato de arquivo.dbc. Você pode fazer upload de arquivos.dbc para importar sinais e sinais de decodificador. Para obter um exemplo de arquivo .dbc, faça o seguinte.

Para obter um arquivo.dbc

1. Faça o download do [EngineSignalsarquivo.zip](#).

2. Navegue até o diretório em que você fez o download do arquivo `EngineSignals.zip`.
3. Descompacte o arquivo e salve-o localmente como `EngineSignals.dbc`.

## Tópicos

- [Criar e gerenciar catálogos de sinais](#)
- [Criar e gerenciar modelos de veículos](#)
- [Criar e gerenciar manifestos de decodificadores](#)

## Criar e gerenciar catálogos de sinais

### Note

Você pode baixar um [script de demonstração](#) para converter mensagens ROS 2 em arquivos VSS JSON compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

Um catálogo de sinais é uma coleção de sinais padronizados que podem ser reutilizados para criar modelos de veículos. AWS A IoT é FleetWise compatível com a [Especificação de Sinais de Veículos \(VSS\)](#) que você pode seguir para definir sinais. Um sinal pode ser de qualquer um dos seguintes tipos.

### Atributo

Os atributos representam informações estáticas que geralmente não mudam, como fabricante e data de fabricação.

### Ramificação

As ramificações representam sinais em uma estrutura aninhada. As ramificações demonstram hierarquias de sinais. Por exemplo, a ramificação `Vehicle` tem uma ramificação secundária, `Powertrain`. A ramificação `Powertrain` tem uma ramificação secundária, `combustionEngine`. Para localizar a ramificação `combustionEngine`, use a expressão `Vehicle.Powertrain.combustionEngine`.



## Sensor

Os dados do sensor informam o estado atual do veículo e mudam ao longo do tempo, à medida que o estado do veículo muda, como níveis de fluidos, temperaturas, vibrações ou tensão.

## Actuador

Os dados do atuador relatam o estado de um dispositivo do veículo, como motores, ar quente e fechaduras de portas. A alteração do estado de um dispositivo do veículo pode atualizar os dados do atuador. Por exemplo, você pode definir um atuador para representar o ar quente. O atuador recebe novos dados quando você liga ou desliga o ar quente.

## Estrutura personalizada

Uma estrutura personalizada (também conhecida como struct) representa uma estrutura de dados complexa ou de ordem superior. Ela facilita o agrupamento ou a vinculação lógica de dados originários da mesma fonte. Uma struct é usada quando os dados são lidos ou gravados em uma operação atômica, como para representar um tipo de dados complexo ou uma forma de ordem superior.

Um sinal do tipo struct é definido no catálogo de sinais usando uma referência a um tipo de dados de struct em vez de um tipo de dados primitivo. As structs podem ser usadas para todos os tipos de sinais, incluindo sensores, atributos, atuadores e tipos de dados de sistemas de visão. Se um sinal do tipo estrutura for enviado ou recebido, a AWS FleetWise IoT espera que todos os itens incluídos tenham valores válidos, portanto, todos os itens são obrigatórios. Por exemplo, caso uma struct contenha os itens `Vehicle.Camera.Image.height`, `Vehicle.Camera.Image.width` e `Vehicle.Camera.Image.data`, espera-se que o sinal enviado contenha valores para todos esses itens.

### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Propriedade personalizada

Uma propriedade personalizada representa um membro da estrutura de dados complexa. O tipo de dados da propriedade pode ser primitivo ou outra struct.

Ao representar uma forma de ordem superior usando uma struct e uma propriedade personalizada, a forma de ordem superior pretendida é sempre definida e visualizada como uma estrutura em árvore. A propriedade personalizada é usada para definir todos os nós da folha, enquanto a struct é usada para definir todos os nós que não são da folha.

### Note

- Se você usa o FleetWise console de AWS IoT para criar o primeiro modelo de veículo, não precisa criar manualmente um catálogo de sinais. Quando você cria seu primeiro modelo de veículo, a AWS IoT cria FleetWise automaticamente um catálogo de sinais para você. Para ter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- Se você usar o FleetWise console de AWS IoT para criar um modelo de veículo, poderá carregar arquivos.dbc para importar sinais. .dbc é um formato de arquivo suportado pelos bancos de dados da Controller Area Network (barramento CAN). Após a criação do modelo do veículo, novos sinais são adicionados automaticamente ao catálogo de sinais. Para ter mais informações, consulte [Criar um modelo de veículo \(console\)](#).
- AWS FleetWise Atualmente, a IoT oferece suporte a um catálogo de sinais para Conta da AWS cada região.

AWS A IoT FleetWise fornece as seguintes operações de API que você pode usar para criar e gerenciar catálogos de sinais.

- [CreateSignalCatalog](#)— Cria um novo catálogo de sinais.
- [ImportSignalCatalog](#)— Importa sinais para criar um catálogo de sinais fazendo o upload de um arquivo JSON. Os sinais devem ser definidos seguindo VSS e salvos no formato JSON.
- [UpdateSignalCatalog](#)— Atualiza um catálogo de sinais existente atualizando, removendo ou adicionando sinais.
- [DeleteSignalCatalog](#)— Exclui um catálogo de sinais existente.
- [ListSignalCatalogs](#)— Recupera uma lista paginada de resumos de todos os catálogos de sinais.
- [ListSignalCatalogNodes](#)— Recupera uma lista paginada de resumos de todos os sinais (nós) em um determinado catálogo de sinais.
- [GetSignalCatalog](#)— Recupera informações sobre um catálogo de sinais.

## Tutoriais

- [Configurar sinais](#)
- [Criar um catálogo de sinais \(AWS CLI\)](#)
- [Importar um catálogo de sinais](#)
- [Atualizar um catálogo de sinais \(AWS CLI\)](#)
- [Excluir um catálogo de sinais \(AWS CLI\)](#)
- [Obter informações do catálogo de sinal \(AWS CLI\)](#)

## Configurar sinais

Esta seção mostra como configurar ramificações, atributos, sensores e atuadores.

### Tópicos

- [Configurar ramificações](#)
- [Configurar atributos](#)
- [Configurar sensores ou atuadores](#)
- [Configurar tipos de dados complexos](#)

## Configurar ramificações

Para configurar uma nova conexão, especifique as seguintes informações:

- `fullyQualifiedName` — O nome totalmente qualificado da ramificação é o caminho para a ramificação além do nome da ramificação. Use um ponto (.) para se referir a uma ramificação secundária. Por exemplo, `Vehicle.Chassis.SteeringWheel` é o nome totalmente qualificado da ramificação `SteeringWheel`. `Vehicle.Chassis.` é o caminho para essa ramificação.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, dois pontos (:) e sublinhado (\_).

- (Opcional) `Description` — Descrição da ramificação.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou da ramificação que está sendo movida ou excluída.

A `deprecationMessage` pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre a ramificação, como a justificativa da ramificação ou referências a ramificações relacionadas.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

## Configurar atributos

Para configurar um atributo, especifique as informações a seguir.

- `dataType`— O tipo de dados do atributo deve ser um dos seguintes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, UINT16\_ARRAY, INT32\_ARRAY, INT32\_ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY, UNKNOWN ou uma estrutura personalizada definida na ramificação do tipo de dados. `fullyQualifiedName`
- `fullyQualifiedName`— O nome totalmente qualificado do atributo é o caminho para o atributo além do nome do atributo. Use um ponto (.) para se referir ao sinal secundário. Por exemplo, `Vehicle.Chassis.SteeringWheel.Diameter` é o nome totalmente qualificado do atributo `Diameter`. `Vehicle.Chassis.SteeringWheel.` é o caminho para esse atributo.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

- (Opcional) `Description` — A descrição do atributo.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `unit` — A unidade científica do atributo, como km ou Celsius.
- (Opcional) `min` — O valor mínimo do atributo.
- (Opcional) `max` — O valor máximo do atributo.
- (Opcional) `defaultValue` — O valor padrão do atributo.
- (Opcional) `assignedValue` — O valor atribuído ao atributo.

- (Opcional) `allowedValues` — Uma lista de valores que o atributo aceita.
- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

A `deprecationMessage` pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o atributo, como a justificativa para o atributo ou referências a atributos relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

## Configurar sensores ou atuadores

Para configurar um sensor ou atuador, especifique as seguintes informações.

- `dataType`— O tipo de dados do sinal deve ser um dos seguintes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, UINT16\_ARRAY, INT32\_ARRAY, INT32\_ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY, UNKNOWN ou uma estrutura personalizada definida na ramificação do tipo de dados. `fullyQualifiedName`
- `fullyQualifiedName` — O nome totalmente qualificado do sinal é o caminho para o sinal além do nome do sinal. Use um ponto (.) para se referir ao sinal secundário. Por exemplo, `Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` é o nome totalmente qualificado do atuador `HandsOffSteeringState`. `Vehicle.Chassis.SteeringWheel.HandsOff.` é o caminho para esse atuador.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `unit` — A unidade científica do sinal, como km ou Celsius.
- (Opcional) `min` — O valor mínimo do sinal.

- (Opcional) `max` — O valor máximo do sinal.
- (Opcional) `assignedValue` — O valor atribuído ao sinal.
- (Opcional) `allowedValues` — lista de valores que o sinal aceita.
- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

A `deprecationMessage` pode ter até 2.048 caracteres. Os caracteres válidos são `a-z`, `A-Z`, `0-9`, `:` (dois pontos), `_` (sublinhado) e `-` (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são `a-z`, `A-Z`, `0-9`, `:` (dois pontos), `_` (sublinhado) e `-` (hífen).

## Configurar tipos de dados complexos

Tipos de dados complexos são usados na modelagem de sistemas de visão. Além das ramificações, esses tipos de dados são compostos de estruturas (também conhecidas como `struct`) e propriedades. Uma `struct` é um sinal descrito por vários valores, como uma imagem. Uma propriedade representa um membro da `struct`, como um tipo de dados primitivo (por exemplo, `UINT8`) ou outra estrutura (como carimbo de data/hora). Por exemplo, `Vehicle.Cameras.Front` representa uma ramificação, `Vehicle.Cameras.Front.Image` representa uma `struct` e `Vehicle.Cameras.Timestamp` representa uma propriedade.

O exemplo de tipo de dados complexos a seguir demonstra como sinais e tipos de dados são exportados para um único arquivo JSON.

### Example tipo de dados complexos

```
{
  "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

Você pode baixar um [script de demonstração](#) para converter mensagens ROS 2 em arquivos VSS JSON compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Configurar struct

Para configurar uma estrutura personalizada (ou struct), especifique as informações a seguir.

- **fullyQualifiedName:** o nome totalmente qualificado da estrutura personalizada. Por exemplo, o nome totalmente qualificado de uma estrutura personalizada pode ser `ComplexDataTypes.VehicleDataTypes.SVMCamera`.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.

A `deprecationMessage` pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

## Configurar propriedade

Para configurar uma propriedade personalizada, especifique as informações a seguir.

- `dataType`: o tipo de dados do sinal deve ser um dos seguintes: `INT8`, `UINT8`, `INT16`, `UINT16`, `INT32`, `UINT32`, `INT64`, `UINT64`, `BOOLEAN`, `FLOAT`, `DOUBLE`, `STRING`, `UNIX_TIMESTAMP`, `INT8_ARRAY`, `UINT8_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`: o nome totalmente qualificado da propriedade personalizada. Por exemplo, o nome totalmente qualificado de uma propriedade personalizada pode ser `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`.

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado)

- (Opcional) `Description` — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `deprecationMessage` — A mensagem de descontinuação do nó ou ramificação que está sendo movido ou excluído.



A `deprecationMessage` pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `comment` — Um comentário adicional à descrição. Um comentário pode ser usado para fornecer informações adicionais sobre o sensor ou atuador, como justificativa ou referências a sensores ou atuadores relacionados.

O comentário pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

- (Opcional) `dataEncoding`: indica se a propriedade é um dado binário. A codificação de dados da propriedade personalizada deve ser uma das seguintes: BINARY ou TYPED.
- (Opcional) `structFullyQualifiedName` — O nome totalmente qualificado do nó de estrutura (estrutura) da propriedade personalizada se o tipo de dados da propriedade personalizada for Estrutura ou StructArray

O nome totalmente qualificado pode ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

## Criar um catálogo de sinais (AWS CLI)

Você pode usar a operação [CreateSignalCatalog](#) da API para criar um catálogo de sinais. O exemplo a seguir usa AWS CLI.

Para criar um catálogo de sinais, execute o comando a seguir.

*signal-catalog-configuration* Substitua pelo nome do arquivo JSON que contém a configuração.

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

- *signal-catalog-name* Substitua pelo nome do catálogo de sinais que você está criando.
- (Opcional) Substitua *description* por uma descrição que ajude a identificar o catálogo de sinais.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar sinais](#).

```
{
  "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

Você pode baixar um [script de demonstração](#) para converter mensagens ROS 2 em arquivos VSS JSON compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Importar um catálogo de sinais


Você pode usar o FleetWise console de AWS IoT ou a API para importar um catálogo de sinais.

### Tópicos

- [Importar um catálogo de sinais \(console\)](#)
- [Importar um catálogo de sinais \(AWS CLI\)](#)

### Importar um catálogo de sinais (console)

Você pode usar o FleetWise console de AWS IoT para importar um catálogo de sinais.

 Important

Você pode ter no máximo um catálogo de sinais. Se você já tem um catálogo de sinais, não verá a opção de importar um catálogo de sinais no console.

### Para importar um catálogo de sinais

1. Abra o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Catálogo de sinais.
3. Na página de resumo do catálogo de sinais, selecione Importar catálogo de sinais.
4. Importe o arquivo que contém os sinais.
  - Para fazer upload de um arquivo de um bucket S3:
    - a. Escolha Import from S3 (Importar do S3).
    - b. Selecione Browse S3 (Navegar no S3).

- c. Em Buckets, insira o nome do bucket ou objeto, selecione-o na lista e escolha o arquivo na lista. Selecione o botão Escolher arquivo.

Ou, para URI do S3, insira um URI do Serviço de Armazenamento Amazon Simple. Para obter mais informações, consulte [Métodos de acesso a um bucket](#) no Guia do usuário do Amazon S3.

- Para fazer upload de um arquivo do seu computador:
  - a. Selecione Importar arquivo.
  - b. Faça upload de um arquivo .json em um formato [Vehicle Signal Specification \(VSS\)](#).

5. Verifique o catálogo de sinais e selecione Importar arquivo.

## Importar um catálogo de sinais (AWS CLI)

Você pode usar a operação da [ImportSignalCatalog](#) API para fazer upload de um arquivo JSON que ajuda a criar um catálogo de sinais. Você deve seguir a [Vehicle Signal Specification \(VSS\)](#) para salvar os sinais no arquivo JSON. O exemplo a seguir usa AWS CLI.

Para importar um catálogo de sinais, execute o comando a seguir.

- *signal-catalog-name* Substitua pelo nome do catálogo de sinais que você está criando.
- (Opcional) Substitua *description* por uma descrição que ajude a identificar o catálogo de sinais.
- *signal-catalog-configuration-vss* Substitua pelo nome do arquivo de string JSON que contém sinais definidos no VSS.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar sinais](#).

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

O JSON deve ser codificado em string e passado pelo campo `vssJson`. Veja um exemplo de sinais definidos na 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"
    }
  }
}
}
```

O exemplo a seguir mostra os mesmos sinais definidos na VSS em uma string 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

Você pode baixar um [script de demonstração](#) para converter mensagens ROS 2 em arquivos VSS JSON compatíveis com o catálogo de sinais. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Atualizar um catálogo de sinais (AWS CLI)

Você pode usar a operação [UpdateSignalCatalog](#) da API para atualizar um catálogo de sinais existente. O exemplo a seguir usa AWS CLI.

Para atualizar um catálogo de sinais existente, execute o comando a seguir.

*signal-catalog-configuration* Substitua pelo nome do arquivo JSON que contém a configuração.

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

*signal-catalog-name* Substitua pelo nome do catálogo de sinais que você está atualizando.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar sinais](#).

### Important

Estruturas personalizadas são imutáveis. Se você precisar reordenar ou inserir propriedades em uma estrutura personalizada (struct) existente, exclua a estrutura e crie uma totalmente nova com a ordem de propriedades desejada.

Para excluir uma estrutura personalizada, adicione o nome totalmente qualificado da estrutura em `nodesToRemove`. Uma estrutura não poderá ser excluída se for referida por algum sinal. Todos os sinais que se referem à estrutura (seu tipo de dados é definido como a estrutura de destino) devem ser atualizados ou excluídos antes da solicitação de atualização do catálogo de sinais.

```
{
  "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
  }
}]
}
```

## Excluir um catálogo de sinais (AWS CLI)

Você pode usar a operação [DeleteSignalCatalog](#) da API para excluir um catálogo de sinais. O exemplo a seguir usa AWS CLI.


### Important

Antes de excluir um catálogo de sinais, verifique se ele não tem modelos de veículos, manifestos de decodificadores, veículos, frotas ou campanhas associados. Para obter instruções, consulte:

- [Excluir um modelo de veículo](#)
- [Excluir um manifesto do decodificador](#)
- [Excluir um veículo](#)
- [Excluir uma frota \(AWS CLI\)](#)
- [Excluir uma campanha](#)

Para excluir um catálogo de sinais existente, execute o comando a seguir. *signal-catalog-name* Substitua pelo nome do catálogo de sinais que você está excluindo.

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

 Note

Esse comando não gera resultados.

## Obter informações do catálogo de sinal (AWS CLI)

Você pode usar a operação [ListSignalCatalogs](#) da API para verificar se um catálogo de sinais foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os catálogos de sinais, execute o seguinte comando.

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

Você pode usar a operação [ListSignalCatalogNodes](#) da API para verificar se um catálogo de sinais foi atualizado. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais (nós) em um determinado catálogo de sinais, execute o seguinte comando.

*signal-catalog-name* Substitua pelo nome do catálogo de sinais que você está verificando.

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

Você pode usar a operação da [GetSignalCatalog](#) API para recuperar informações do catálogo de sinais. O exemplo a seguir usa AWS CLI.

Para recuperar informações sobre um catálogo de sinais, execute o comando a seguir.

*signal-catalog-name* Substitua pelo nome do catálogo de sinais que você deseja recuperar.

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

 Note


Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações no catálogo de sinais podem não ser refletidas imediatamente.

## Criar e gerenciar modelos de veículos

Você usa sinais para criar modelos de veículos que ajudam a padronizar o formato de seus veículos. Os modelos de veículos impõem informações consistentes em vários veículos do mesmo tipo, para que você possa processar dados de frotas de veículos. Veículos criados com base no mesmo modelo de veículo herdam o mesmo grupo de sinais. Para ter mais informações, consulte [Criar, provisionar e gerenciar veículos](#).

Cada modelo de veículo tem um campo de status que contém o estado do modelo do veículo. O estado pode ser um dos seguintes valores:

- ACTIVE — O modelo do veículo está ativo.
- DRAFT — A configuração do modelo do veículo será salva.

 Important

- Se você quiser usar a operação da API do `CreateModelManifest` para criar o primeiro modelo de veículo, antes deverá criar um catálogo de sinais. Para ter mais informações, consulte [Criar um catálogo de sinais \(AWS CLI\)](#).
- Se você usa o FleetWise console de AWS IoT para criar um modelo de veículo, a AWS IoT ativa FleetWise automaticamente o modelo de veículo para você.
- Se você usar a operação da API do `CreateModelManifest` para criar um modelo de veículo, o modelo do veículo permanecerá no estado DRAFT.
- Você não pode criar veículos usando modelos que estejam no estado DRAFT. Use a operação da API `UpdateModelManifest` para alterar os modelos de veículos para o estado ACTIVE.
- Não é possível editar modelos de veículos que estejam no estado ACTIVE.

## Tópicos

- [Criar um modelo de veículo](#)
- [Atualizar o modelo de um veículo \(AWS CLI\)](#)
- [Excluir um modelo de veículo](#)
- [Obter informações sobre o modelo do veículo \(AWS CLI\)](#)

## Criar um modelo de veículo

Você pode usar o FleetWise console de AWS IoT ou a API para criar modelos de veículos.

### Important

Você deve ter um catálogo de sinais antes de criar um modelo de veículo usando a operação de API `CreateModelManifest`.

### Tópicos

- [Criar um modelo de veículo \(console\)](#)
- [Criar um modelo de veículo \(AWS CLI\)](#)

## Criar um modelo de veículo (console)

No FleetWise console de AWS IoT, você pode criar um modelo de veículo das seguintes formas:

- [Use um modelo fornecido por AWS](#)
- [Criar manualmente um modelo de veículo](#)
- [Duplicar um modelo de veículo](#)

### Use um modelo fornecido por AWS

AWS FleetWise A IoT fornece um modelo de diagnóstico integrado (OBD) II, J1979 que cria automaticamente um catálogo de sinais, um modelo de veículo e um manifesto de decodificador para você. O modelo também adiciona interfaces de rede OBD ao manifesto do decodificador. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

Para criar um modelo de veículo usando um modelo

1. Navegue até o console [AWS de IoT. FleetWise](#)

2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Adicionar modelo fornecido.
4. Selecione On-board diagnostics (OBD) II .
5. Insira um nome para a interface de rede OBD que a AWS FleetWise IoT está criando.
6. Escolha Add.

### Criar manualmente um modelo de veículo

Você pode adicionar sinais do catálogo de sinais ou importar sinais fazendo upload de um ou mais arquivos .dbc. Um arquivo .dbc é um formato de arquivo compatível com os bancos de dados Controller Area Network (barramento CAN).

#### Important

Você não pode criar um modelo de veículo com sinais de dados do sistema de visão usando o console de AWS IoT FleetWise . Em vez disso, use o AWS CLI para criar um modelo de veículo.

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

### Para criar manualmente um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Criar modelo de veículo e faça o seguinte.

### Tópicos

- [Etapa 1: configurar o modelo do veículo](#)
- [Etapa 2: adicionar sinais](#)
- [Etapa 3: importar sinais](#)
- [\(Opcional\) Etapa 4: adicionar atributos](#)
- [Etapa 5: analisar e criar](#)

## Etapa 1: configurar o modelo do veículo

Em Informações gerais, faça o seguinte:

1. Insira um nome para o modelo do veículo.
2. (Opcional) Insira uma descrição.
3. Escolha Próximo.

## Etapa 2: adicionar sinais

### Note

- Se for a primeira vez que você usa a AWS IoT FleetWise, essa etapa não estará disponível até que você tenha um catálogo de sinais. Quando o primeiro modelo de veículo é criado, a AWS IoT cria FleetWise automaticamente um catálogo de sinais com sinais adicionados ao primeiro modelo de veículo.
- Se você tem experiência com AWS IoT FleetWise, pode adicionar sinais ao modelo do seu veículo selecionando sinais do catálogo de sinais ou carregando arquivos.dbc para importar sinais.
- Você deve ter pelo menos um sinal para criar um modelo de veículo.

Para adicionar sinais

1. Escolha um ou mais sinais do catálogo de sinais que você está adicionando ao modelo do veículo. Você pode revisar os sinais selecionados no painel direito.

### Note

Apenas os sinais selecionados serão adicionados ao modelo do veículo.

2. Escolha Próximo.

## Etapa 3: importar sinais

### Note

- Se for a primeira vez que você usa a AWS IoT FleetWise, você deve carregar pelo menos um arquivo.dbc para importar sinais.
- Se você tem experiência com AWS IoT FleetWise, pode adicionar sinais ao modelo do seu veículo selecionando sinais do catálogo de sinais ou carregando arquivos.dbc para importar sinais.
- Você deve ter pelo menos um sinal para criar um modelo de veículo.

### Para importar sinais

1. Selecione Selecionar arquivos.
2. Na caixa de diálogo, escolha o arquivo .dbc que contém sinais. Você pode fazer upload de vários arquivos .dbc.
3. AWS A IoT FleetWise analisa seus arquivos.dbc para recuperar sinais.

Na seção Sinais, especifique os seguintes metadados para cada sinal.

- Name — Nome do sinal.

O nome do sinal deve ser exclusivo. O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

- Data type — O tipo de dados do sinal deve ser um dos seguintes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, UINT16\_ARRAY, INT32\_ARRAY, UINT32\_ARRAY, INT64\_ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY ou UNKNOWN.
- Signal type — O tipo de sinal, que pode ser Sensor ou Atuador.
- (Opcional) Unit — A unidade científica do sinal, como km ou Celsius.
- (Opcional) Path — O caminho para o sinal. Semelhante ao JSONPath, use um ponto (.) para se referir a um sinal secundário. Por exemplo, **Vehicle.Engine.Light**.

O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado).

- (Opcional) Min — O valor mínimo do sinal.
- (Opcional) Max — O valor máximo do sinal.
- (Opcional) Description — A descrição do sinal.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

#### 4. Escolha Próximo.

(Opcional) Etapa 4: adicionar atributos

É possível adicionar até 100 atributos, incluindo os atributos existentes no catálogo de sinais.

Para adicionar atributos

##### 1. Em Adicionar atributos, especifique os seguintes metadados para cada atributo.

- Name — Nome do atributo.

O nome do sinal deve ser exclusivo. O nome e o caminho do sinal podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado)

- Data type — O tipo de dados do atributo deve ser um dos seguintes: INT8, UINT8, INT16, UINT16, INT32, UINT32, INT64, UINT64, BOOLEAN, FLOAT, DOUBLE, STRING, UNIX\_TIMESTAMP, INT8\_ARRAY, UINT8\_ARRAY, INT16\_ARRAY, UINT16\_ARRAY, INT32\_ARRAY, UINT32\_ARRAY, INT64\_ARRAY, UINT64\_ARRAY, BOOLEAN\_ARRAY, FLOAT\_ARRAY, DOUBLE\_ARRAY, STRING\_ARRAY, UNIX\_TIMESTAMP\_ARRAY ou UNKNOWN
- (Opcional) Unit — A unidade científica do atributo, como km ou Celsius.
- (Opcional) Path — O caminho para o sinal. Semelhante ao JSONPath, use um ponto (.) para se referir a um sinal secundário. Por exemplo, **Vehicle.Engine.Light**.

O nome do sinal mais o caminho podem ter até 150 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos) e \_ (sublinhado)

- (Opcional) Min — O valor mínimo do atributo.
- (Opcional) Max — O valor máximo do atributo.



- (Opcional) Description — A descrição do atributo.

A descrição pode ter até 2.048 caracteres. Os caracteres válidos são a–z, A–Z, 0–9, : (dois pontos), \_ (sublinhado) e - (hífen).

## 2. Escolha Próximo.

### Etapa 5: analisar e criar

Verifique as configurações do modelo do veículo e selecione Criar.

### Duplicar um modelo de veículo

AWS A IoT FleetWise pode copiar as configurações de um modelo de veículo existente para criar um novo modelo. Os sinais especificados no modelo de veículo selecionado são copiados para o novo modelo de veículo.

Para duplicar um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione um modelo na lista de modelos de veículos e depois Duplicar modelo.

Para configurar o modelo do veículo, siga o tutorial [Criar manualmente um modelo de veículo](#).

Pode levar alguns minutos para que a AWS IoT processe sua solicitação FleetWise para criar o modelo do veículo. Após a criação bem-sucedida do modelo do veículo, na página Modelos de veículos, a coluna Status mostra ATIVO. Quando o modelo do veículo se torna ativo, não é possível editá-lo.

### Criar um modelo de veículo (AWS CLI)

Você pode usar a operação da [CreateModelManifestAPI](#) para criar modelos de veículos (manifestos de modelos). O exemplo a seguir usa a AWS CLI.

**⚠ Important**

Se você quiser usar a FleetWise API de AWS IoT para criar o primeiro modelo de veículo, primeiro deverá criar um catálogo de sinais. Para obter mais informações sobre como criar um catálogo de sinais, consulte [Criar um catálogo de sinais \(AWS CLI\)](#).

Para criar um modelo de veículo, execute o seguinte comando.

*vehicle-model-configuration* Substitua pelo nome do arquivo JSON que contém a configuração.

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

- *vehicle-model-name* Substitua pelo nome do modelo do veículo que você está criando.
- Substitua *signal-catalog-ARN* pelo nome do recurso da Amazon (ARN) do catálogo de sinais.
- (Opcional) Substitua *description* por uma descrição que ajude a identificar o modelo do veículo.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar sinais](#).

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

## Atualizar o modelo de um veículo (AWS CLI)

Você pode usar a operação da [UpdateModelManifest](#) API para atualizar um modelo de veículo existente (manifestos de modelo). O exemplo a seguir usa a AWS CLI.

Para atualizar um modelo de veículo existente, execute o seguinte comando.

*update-vehicle-model-configuration* Substitua pelo nome do arquivo JSON que contém a configuração.

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

- *vehicle-model-name* Substitua pelo nome do modelo do veículo que você está atualizando.
- (Opcional) Para ativar o modelo do veículo, *vehicle-model-status* substitua por ACTIVE.

#### Important

Após a ativação do modelo do veículo, você não poderá alterá-lo.

- (Opcional) Substitua *description* por uma descrição atualizada para ajudar a identificar o modelo do veículo.

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

## Excluir um modelo de veículo

Você pode usar o FleetWise console de AWS IoT ou a API para excluir modelos de veículos.

#### Important

Os veículos e os manifestos do decodificador associados ao modelo do veículo devem ser excluídos primeiro. Para ter mais informações, consulte [Excluir um veículo](#) e [Excluir um manifesto do decodificador](#).

## Excluir um modelo de veículo (console)

Para excluir um modelo de veículo, use o console de AWS IoT. FleetWise

## Para excluir um modelo de veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, escolha o modelo do veículo de destino.
4. Escolha Delete.
5. Em Excluir **vehicle-model-name**?, insira o nome do modelo do veículo a ser excluído e escolha Confirmar.

## Excluir um modelo de veículo (AWS CLI)

Você pode usar a operação da [DeleteModelManifest](#) API para excluir um modelo de veículo existente (manifestos de modelo). O exemplo a seguir usa a AWS CLI.

Para excluir um modelo de veículo, execute o seguinte comando.

*model-manifest-name* Substitua pelo nome do modelo do veículo que você está excluindo.

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

### Note

Esse comando não gera resultados.

## Obter informações sobre o modelo do veículo (AWS CLI)

Você pode usar a operação [ListModelManifests](#) da API para verificar se um modelo de veículo foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os modelos de veículos, execute o seguinte comando.

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

Você pode usar a operação [ListModelManifestNodes](#) da API para verificar se o modelo de um veículo foi atualizado. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais (nós) em um determinado modelo de veículo, execute o seguinte comando.

*vehicle-model-name* Substitua pelo nome do modelo do veículo que você está verificando.

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

Para recuperar informações sobre um modelo de veículo, execute o comando a seguir.

Substitua *vehicle-model* pelo nome do modelo do veículo que você deseja recuperar.

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

#### Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações ao modelo do veículo podem não ser refletidas imediatamente.

## Criar e gerenciar manifestos de decodificadores

Os manifestos do decodificador contêm informações de decodificação que a AWS IoT FleetWise usa para transformar dados do veículo (dados binários) em valores legíveis por humanos e preparar seus dados para análises de dados. A interface de rede e os sinais do decodificador são os principais componentes com os quais você trabalha para configurar os manifestos do decodificador.

### Interface de rede

Contém informações sobre o protocolo que a rede do veículo usa. AWS IoT é FleetWise compatível com os seguintes protocolos.

#### Controller Area Network (barramento CAN)

Um protocolo que define como os dados são comunicados entre unidades de controle eletrônico (ECUs). As ECUs podem ser a unidade de controle do motor, airbags ou sistema de áudio.

## On-board diagnostic (OBD) II

Um protocolo mais desenvolvido que define como os dados de autodiagnóstico são comunicados entre as ECUs. Ele fornece uma série de códigos de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado no veículo.

## Middleware de veículo

O middleware de veículo definido como um tipo de interface de rede. Exemplos de middleware de veículo incluem Robot Operating System (ROS 2) e Scalable Service-Oriented Middleware over IP (SOME/IP).

### Note

AWS A IoT FleetWise oferece suporte ao middleware ROS 2 para dados do sistema de visão.

## Sinal do decodificador

Fornece informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um sinal de decodificador. Se o manifesto do decodificador contiver interfaces de rede CAN, ele deverá conter sinais do decodificador CAN. Se o manifesto do decodificador contiver interfaces de rede OBD, ele deverá conter sinais do decodificador OBD.

O manifesto do decodificador deverá conter sinais do decodificador de mensagens se também contiver interfaces de middleware do veículo.

Cada manifesto do decodificador deve estar associado a um modelo de veículo. AWS A IoT FleetWise usa o manifesto do decodificador associado para decodificar dados de veículos criados com base no modelo do veículo.

Cada manifesto do decodificador tem um campo de status que contém o estado do manifesto do decodificador. O estado pode ser um dos seguintes valores:

- ACTIVE — O manifesto do decodificador está ativo.
- DRAFT — A configuração do manifesto do decodificador não é salva.

- **VALIDATING:** o manifesto do decodificador está em validação quanto a sua elegibilidade. Isso se aplica somente aos manifestos do decodificador que contêm pelo menos um sinal de dados de sistemas de visão.
- **INVALID:** o manifesto do decodificador falhou na validação e ainda não pode ser ativado. Isso se aplica somente aos manifestos do decodificador que contêm pelo menos um sinal de dados de sistemas de visão. Você pode usar as `GetDecoderManifest` APIs `ListDecoderManifests` e para verificar o motivo da falha na validação.

#### Important

- Se você usa o FleetWise console de AWS IoT para criar um manifesto do decodificador, a AWS IoT ativa FleetWise automaticamente o manifesto do decodificador para você.
- Se você usar a operação da API `CreateDecoderManifest` para criar um manifesto do decodificador, o manifesto do decodificador permanecerá no estado `DRAFT`.
- Você não pode criar veículos usando modelos de veículos associados a um manifesto do decodificador da `DRAFT`. Use a operação da API `UpdateDecoderManifest` para alterar o manifesto do decodificador para o estado `ACTIVE`.
- Você não pode editar manifestos do decodificador que estejam no estado `ACTIVE`.

## Tópicos

- [Configurar interfaces de rede e sinais de decodificador](#)
- [Criar um manifesto do decodificador](#)
- [Atualizar um manifesto do decodificador \(AWS CLI\)](#)
- [Excluir um manifesto do decodificador](#)
- [Obter informações de manifesto do decodificador \(AWS CLI\)](#)

## Configurar interfaces de rede e sinais de decodificador

Cada manifesto de decodificador tem pelo menos uma interface de rede e sinais de decodificador emparelhados com sinais especificados no modelo de veículo associado.

Se o manifesto do decodificador contiver interfaces de rede CAN, ele deverá conter sinais do decodificador CAN. Se o manifesto do decodificador contiver interfaces de rede OBD, ele deverá conter sinais do decodificador OBD.

## Tópicos

- [Configurar interfaces de rede](#)
- [Configurar sinais do decodificador](#)

## Configurar interfaces de rede

Para configurar uma interface de rede CAN, especifique as informações a seguir.

- `name` — O nome da interface CAN.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- (Opcional) `protocolName` — O nome do protocolo.

Valores válidos: CAN-FD e CAN

- (Opcional) `protocolVersion` — FleetWise Atualmente, a AWS IoT suporta CAN-FD e CAN 2.0b.

Valores válidos: 1.0 e 2.0b

Para configurar uma interface de rede OBD, especifique as informações a seguir.

- `name` — O nome da interface OBD.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- `requestMessageId` — O ID da mensagem que está solicitando dados.
- (Opcional) `dtcRequestIntervalSeconds` — Com que frequência solicitar códigos de problemas de diagnóstico (DTCs) do veículo em segundos. Por exemplo, se o valor especificado for 120, o software Edge Agent coletará DTCs armazenados uma vez a cada 2 minutos.
- (Opcional) `hasTransmissionEcu` — Se o veículo possui módulo de controle de transmissão (TCM).

Valores válidos: `true` e `false`



- (Opcional) `obdStandard` — O padrão OBD compatível com a AWS FleetWise IoT. AWS FleetWise Atualmente, a IoT suporta o padrão ISO15765-4 do World Wide Harmonization On-Board Diagnostics (WWH-OBD).
- (Opcional) `pidRequestIntervalSeconds` — Com que frequência solicitar PIDs OBD II do veículo. Por exemplo, se o valor especificado for 120, o software Edge Agent coletará PIDs OBD II uma vez a cada 2 minutos.
- (Opcional) `useExtendedIds` — Caso utilize IDs ampliados na mensagem.

Valores válidos: `true` e `false`

Para configurar uma interface de rede de middleware, especifique as informações a seguir.

- `name`: o nome da interface de middleware do veículo.

O nome da interface deve ser exclusivo e pode ter de 1 a 100 caracteres.

- `protocolName`: o nome do protocolo.

Valores válidos: `R0S_2`

## Configurar sinais do decodificador

Para configurar um sinal de decodificador CAN, especifique as informações a seguir.

- `factor` — O multiplicador usado para decodificar a mensagem.
- `isBigEndian` — Caso a ordem de bytes de uma mensagem seja big endian. Se for big endian, o valor mais significativo na sequência será armazenado primeiro, no endereço de armazenamento mais baixo.
- `isSigned` — Caso a mensagem esteja assinada. Se estiver assinada, a mensagem poderá representar números positivos e negativos.
- `length` — O comprimento da mensagem em bytes.
- `messageId` — O ID da mensagem.
- `offset` — O deslocamento usado para calcular o valor do sinal. Combinado com o fator, o cálculo é `value = raw_value * factor + offset`.
- `startBit` — Indica a localização do primeiro bit da mensagem.
- (Opcional) `name` — Nome do sinal.

Para configurar um sinal de decodificador OBD, especifique as informações a seguir.

- `byteLength` — O comprimento da mensagem em bytes.
- `offset` — O deslocamento usado para calcular o valor do sinal. Combinado com a escala, o cálculo é  $value = raw\_value * scaling + offset$ .
- `pid` — O código de diagnóstico utilizado para solicitar uma mensagem de um veículo para este sinal.
- `pidResponseLength` — O tamanho da mensagem solicitada.
- `scaling` — O multiplicador utilizado para decodificar a mensagem.
- `serviceMode` — O modo de operação (serviço de diagnóstico) em uma mensagem.
- `startByte` — Indica o início da mensagem.
- (Opcional) `bitMaskLength` — O número de bits mascarados em uma mensagem.
- (Opcional) `bitRightShift` — O número de posições deslocadas para a direita.

Para configurar um sinal de decodificador de mensagens, especifique as informações a seguir.

- `topicName`: o nome do tópico para o sinal da mensagem. Corresponde aos tópicos do ROS 2. Para obter mais informações sobre o objeto de mensagem estruturada, consulte [StructuredMessage](#).
- `structuredMessage`: a mensagem estruturada para o sinal da mensagem. Ele pode ser definido com a `primitiveMessageDefinition`, `structuredMessageList` Definição ou `structuredMessageDefinition` recursivamente.

## Criar um manifesto do decodificador

Você pode usar o FleetWise console de AWS IoT ou a API para criar um manifesto de decodificador para o modelo do seu veículo.

### Important

Você deve ter um modelo de veículo antes de poder criar um manifesto do decodificador. Cada manifesto do decodificador deve estar associado a um modelo de veículo. Para ter mais informações, consulte [Criar e gerenciar modelos de veículos](#).

## Tópicos

- [Criar um manifesto do decodificador \(console\)](#)
- [Criar um manifesto do decodificador \(AWS CLI\)](#)

## Criar um manifesto do decodificador (console)

Você pode usar o FleetWise console de AWS IoT para criar um manifesto do decodificador associado ao modelo do seu veículo.

### Important

Você não pode configurar sinais de dados do sistema de visão em manifestos de decodificadores usando o console de IoT AWS . FleetWise Use a AWS CLI em vez disso. O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Para criar um manifesto do decodificador

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione o modelo de veículo de destino.
4. Na página de resumo do modelo do veículo, selecione Criar manifesto do decodificador e faça o seguinte.

## Tópicos

- [Etapa 1: configurar o manifesto do decodificador](#)
- [Etapa 2: adicionar interfaces de rede](#)
- [Etapa 3: revisar e criar](#)

## Etapa 1: configurar o manifesto do decodificador

Em Informações gerais, faça o seguinte:

1. Insira um nome exclusivo para o manifesto do decodificador.
2. (Opcional) Insira uma descrição.

### 3. Escolha Próximo.

#### Etapa 2: adicionar interfaces de rede

Cada manifesto do decodificador deve ter pelo menos uma interface de rede. Você pode adicionar várias interfaces de rede a um manifesto do decodificador.

#### Adicionar uma interface de rede

- Em Interface de rede, faça o seguinte.
  - a. Para o tipo de interface de rede, selecione o CAN\_INTERFACE ou OBD\_INTERFACE.
  - b. Insira um nome exclusivo para sua interface de rede.
  - c. Insira um ID de interface de rede exclusivo. Você pode usar o ID gerado pela AWS IoT FleetWise.
  - d. Selecione um ou mais sinais especificados no modelo do seu veículo para emparelhar com sinais do decodificador.
  - e. Para fornecer informações de decodificação, faça upload de um arquivo .dbc. AWS A IoT FleetWise analisa o arquivo.dbc para recuperar sinais do decodificador.
  - f. Na seção Sinais emparelhados, verifique se cada sinal está emparelhado com um sinal de decodificador.
  - g. Escolha Próximo.

#### Note

- Você pode carregar apenas um arquivo .dbc para cada interface de rede.
- Verifique se cada sinal especificado no modelo do seu veículo está emparelhado com um sinal de decodificador.
- Depois de optar por adicionar outra interface de rede, você não poderá editar aquela que está editando. Você pode excluir as interfaces de rede existentes.

#### Etapa 3: revisar e criar

Verifique as configurações do manifesto do decodificador e selecione Criar.

## Criar um manifesto do decodificador (AWS CLI)

Você pode usar a operação da [CreateDecoderManifest](#) API para criar manifestos do decodificador. O exemplo a seguir usa a AWS CLI.

### Important

Antes de criar um manifesto do decodificador, primeiro crie um modelo de veículo. Para ter mais informações, consulte [Criar um modelo de veículo](#).

Para criar um manifesto do decodificador, execute o seguinte comando.

*decoder-manifest-configuration* Substitua pelo nome do arquivo JSON que contém a configuração.

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

- *decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está criando.
- Substitua *vehicle-model-ARN* pelo o nome do recurso da Amazon (ARN) do modelo do veículo.
- (Opcional) Substitua *description* por uma descrição que ajude a identificar o manifesto do decodificador.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar interfaces de rede e sinais de decodificador](#).

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [
    {
      "canInterface": {
        "name": "myNetworkInterface",
        "protocolName": "CAN",
        "protocolVersion": "2.0b"
      }
    }
  ]
}
```

```

    },
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_INTERFACE"
  }
],
"signalDecoders": [
  {
    "canSignal": {
      "name": "Engine_Idle_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 16
    },
    "fullyQualified_name": "Vehicle.EngineIdleTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  },
  {
    "canSignal": {
      "name": "Engine_Run_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 40
    },
    "fullyQualified_name": "Vehicle.EngineRunTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  }
]
}

```

- *decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está criando.
- Substitua *vehicle-model-ARN* pelo o nome do recurso da Amazon (ARN) do modelo do veículo.

- (Opcional) Substitua *description* por uma descrição que ajude a identificar o manifesto do decodificador.

A ordem dos nós de propriedade dentro de uma estrutura (struct) deve permanecer consistente conforme definido no catálogo de sinais e no modelo do veículo (manifesto do modelo). Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte [Configurar interfaces de rede e sinais de decodificador](#).

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [{
    "canInterface": {
      "name": "myNetworkInterface",
      "protocolName": "CAN",
      "protocolVersion": "2.0b"
    },
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "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": "Qq1acaenBy0B3sSM39SYm",
    "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": "Qq1lacaenBy0B3sSM39SYm",
  "type": "CAN_SIGNAL"
},
{
  "fullyQualifiedName": "Vehicle.CompressedImageTopic",
  "type": "MESSAGE_SIGNAL",
  "interfaceId": "G1KzxxkdnmV5Hn7wkV3ZL9",
  "messageSignal": {
    "topicName": "CompressedImageTopic:sensor_msgs/msg/CompressedImage",
    "structuredMessage": {
      "structuredMessageDefinition": [{
        "fieldName": "header",
        "dataType": {
          "structuredMessageDefinition": [{
            "fieldName": "stamp",
            "dataType": {
              "structuredMessageDefinition": [{
                "fieldName": "sec",
                "dataType": {
                  "primitiveMessageDefinition": {
                    "ros2PrimitiveMessageDefinition": {
                      "primitiveType": "INT32"
                    }
                  }
                }
              ]
            }
          ],
          "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

Você pode baixar um [script de demonstração](#) para criar um manifesto do decodificador com sinais de sistemas de visão. Para ter mais informações, consulte o [Vision System Data Developer Guide](#).

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

## Atualizar um manifesto do decodificador (AWS CLI)

Você pode usar a operação da [UpdateDecoderManifest](#) API para atualizar um manifesto do decodificador. Você pode adicionar, remover e atualizar interfaces de rede e decodificadores de sinal. Você também pode alterar o status do manifesto do decodificador. O exemplo a seguir usa a AWS CLI.

Para atualizar um manifesto do decodificador, execute o comando a seguir.

*decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está atualizando.

```
aws iotfleetwise update-decoder-manifest /
    --name decoder-manifest-name /
    --status ACTIVE
```

### Important

Depois de ativar o manifesto do decodificador, você não poderá editá-lo.

## Excluir um manifesto do decodificador

Você pode usar o FleetWise console de AWS IoT ou a API para excluir um manifesto do decodificador.

### Important

Os veículos associados ao manifesto do decodificador devem ser excluídos primeiro. Para ter mais informações, consulte [Excluir um veículo](#).

### Tópicos

- [Excluir um manifesto do decodificador \(console\)](#)
- [Excluir um manifesto do decodificador \(AWS CLI\)](#)

### Excluir um manifesto do decodificador (console)

Você pode usar o FleetWise console de AWS IoT para excluir um manifesto do decodificador.

Para excluir um manifesto do decodificador

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Modelos de veículos.
3. Selecione o modelo de veículo de destino.
4. Na página de resumo do modelo do veículo, selecione a guia Manifestos do decodificador.
5. Selecione o manifesto do decodificador de destino e depois Excluir.
6. Em Excluir **decoder-manifest-name?**, insira o nome do manifesto do decodificador a ser excluído e selecione Confirmar.

### Excluir um manifesto do decodificador (AWS CLI)

Você pode usar a operação da [DeleteDecoderManifest](#) API para excluir um manifesto do decodificador. O exemplo a seguir usa AWS CLI.

**⚠ Important**

Antes de excluir o manifesto do decodificador, exclua primeiro os veículos associados. Para ter mais informações, consulte [Excluir um veículo](#).

Para excluir um manifesto do decodificador, execute o comando a seguir.

*decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está excluindo.

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

## Obter informações de manifesto do decodificador (AWS CLI)

Você pode usar a operação da [ListDecoderManifests](#) API para verificar se um manifesto do decodificador foi excluído. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os manifestos do decodificador, execute o comando a seguir.

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

Você pode usar a operação da [ListDecoderManifestSignals](#) API para verificar se os sinais do decodificador no manifesto do decodificador foram atualizados. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todos os sinais do decodificador (nós) em um determinado manifesto do decodificador, execute o comando a seguir.

*decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está verificando.

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

Você pode usar a operação da [ListDecoderManifestNetworkInterfaces](#) API para verificar se as interfaces de rede no manifesto do decodificador foram atualizadas. O exemplo a seguir usa AWS CLI.

Para recuperar uma lista paginada de resumos de todas as interfaces de rede em um determinado manifesto do decodificador, execute o comando a seguir.

*decoder-manifest-name* Substitua pelo nome do manifesto do decodificador que você está verificando.

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

Você pode usar a operação da [GetDecoderManifest](#) API para verificar se as interfaces de rede e os sinais do decodificador no manifesto do decodificador foram atualizados. O exemplo a seguir usa AWS CLI.

Para recuperar informações sobre um manifesto do decodificador, execute o comando a seguir.

Substitua *decoder-manifest* pelo nome do manifesto do decodificador que você quer recuperar.

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

#### Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações no manifesto do decodificador podem não ser refletidas imediatamente.

# Criar, provisionar e gerenciar veículos

Veículos são instâncias de modelos de veículos. Os veículos devem ser criados com base em um modelo de veículo e associados a um manifesto do decodificador. Os veículos carregam um ou mais fluxos de dados para a nuvem. Por exemplo, um veículo pode enviar dados de quilometragem, temperatura do motor e estado do ar quente para a nuvem. Cada veículo contém as seguintes informações:

## `vehicleName`

Um ID que identifica o veículo.

Não inclua informações de identificação pessoal (PII) nem outras informações confidenciais ou sigilosas em nomes de filas. Os nomes dos veículos podem ser acessados por outros AWS serviços, incluindo a Amazon CloudWatch. Os nomes dos veículos não se destinam a ser usados para dados privados ou confidenciais.

## `modelManifestARN`

O nome de recurso da Amazon (ARN) de um modelo de veículo (manifesto do modelo). Cada veículo é criado usando um modelo de veículo. Os veículos criados usando o mesmo modelo de veículo consistem no mesmo grupo de sinais herdados do modelo de veículo. Esses sinais são definidos e padronizados no catálogo de sinais.

## `decoderManifestArn`

O ARN do manifesto do decodificador. Um manifesto do decodificador fornece informações de decodificação que a AWS IoT FleetWise pode usar para transformar dados brutos do sinal (dados binários) em valores legíveis por humanos. Um manifesto do decodificador deve estar associado a um modelo de veículo. AWS A IoT FleetWise usa o mesmo manifesto do decodificador para decodificar dados brutos de veículos criados com base no mesmo modelo de veículo.

## `attributes`

Atributos são pares de valores-chave que contêm informações estáticas. Os veículos podem conter atributos herdados do modelo do veículo. Você pode adicionar outros atributos para distinguir um veículo individual de outros veículos criados usando o mesmo modelo de veículo. Por exemplo, se você tiver um carro preto, poderá especificar o seguinte valor para um atributo: `{"color": "black"}`.

**⚠ Important**

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

Para obter mais informações sobre modelos de veículos, manifestos de decodificadores e atributos, consulte [Modelagem de veículos](#).

AWS A IoT FleetWise fornece as seguintes operações de API que você pode usar para criar e gerenciar veículos.

- [CreateVehicle](#)— Cria um novo veículo.
- [BatchCreateVehicle](#)— Cria um ou mais veículos novos.
- [UpdateVehicle](#)— Atualiza um veículo existente.
- [BatchUpdateVehicle](#)— Atualiza um ou mais veículos existentes.
- [DeleteVehicle](#)— Exclui um veículo existente.
- [ListVehicles](#)— Recupera uma lista paginada de resumos de todos os veículos.
- [GetVehicle](#)— Recupera informações sobre um veículo.

## Tutoriais

- [Provisionar veículos](#)
- [Tópicos reservados](#)
- [Criar um veículo](#)
- [Atualizar um veículo \(AWS CLI\)](#)
- [Atualizar vários veículos \(AWS CLI\)](#)
- [Excluir um veículo](#)
- [Obter informações do veículo \(AWS CLI\)](#)

## Provisionar veículos

O FleetWise software Edge Agent para AWS IoT executado em seu veículo coleta e transfere dados para a nuvem. AWS A IoT FleetWise se integra AWS IoT Core para oferecer suporte à comunicação segura entre o software Edge Agent e a nuvem por meio do MQTT. Cada veículo corresponde a

AWS IoT alguma coisa. Você pode usar AWS IoT algo existente para criar um veículo ou configurar a AWS IoT FleetWise para criar automaticamente AWS IoT algo para seu veículo. Para ter mais informações, consulte [Criar um veículo \(AWS CLI\)](#).

AWS IoT Core oferece suporte à [autenticação](#) e [autorização](#) que ajudam a controlar com segurança o acesso aos recursos de IoT AWS . FleetWise Os veículos podem usar certificados X.509 para serem autenticados (conectados) para usar a AWS IoT FleetWise e AWS IoT Core políticas para serem autorizados (ter permissões) a realizar ações específicas.

## Autenticar veículos

Você pode criar AWS IoT Core políticas para autenticar seus veículos.

Para autenticar seu veículo

- Para criar uma AWS IoT Core política, execute o comando a seguir.
  - Substitua *policy-name* pelo nome da política que você quer criar.
  - Substitua *file-name* pelo nome do arquivo JSON que contém a política. AWS IoT Core

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

Antes de usar a política de exemplo, faça o seguinte:

- Substitua a *região* pela AWS região em que você criou recursos de AWS IoT. FleetWise
- Substitua o *AWS Account* pelo ID da sua AWS conta.

Este exemplo inclui tópicos reservados pela AWS IoT FleetWise. Você deve adicionar os tópicos à política. Para ter mais informações, consulte [Tópicos reservados](#).

```
{
  "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"
    ]
}
]

```

```
}
```

## Autorizar veículos

Você pode criar certificados X.509 para autorizar seus veículos.

Para autorizar seu veículo

### Important

Recomendamos que você crie um novo certificado para cada veículo.

1. Para criar um par de chaves RSA e emitir um certificado X.509, execute o comando a seguir.
  - Substitua *cert* pelo nome do arquivo que salva o conteúdo de saída do comando do `certificatePem`.
  - Substitua *public-key* pelo nome do arquivo que salva o conteúdo de saída do comando do `KeyPair.PublicKey`.
  - Substitua a *chave privada* pelo nome do arquivo que salva o conteúdo da saída do comando do `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. Copie o nome de recurso da Amazon (ARN) do certificado da saída.
3. Para anexar a política ao certificado, execute o comando a seguir.
  - Substitua *policy-name* pelo nome da AWS IoT Core política que você criou.
  - Substitua *certificate-arn* pelo ARN do certificado que você copiou.

```
aws iot attach-policy \  
  --policy-name policy-name
```

```
--target "certificate-arn"
```

4. Para anexar o certificado à coisa, execute o seguinte comando.

- *Substitua o nome* da coisa pelo nome da sua AWS IoT coisa ou pela ID do seu veículo.
- Substitua *certificate-arn* pelo ARN do certificado que você copiou.

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

## Tópicos reservados

AWS A IoT FleetWise reserva o uso dos seguintes tópicos. Se o tópico reservado permitir, você poderá se inscrever ou publicar nele. No entanto, não é possível criar novos tópicos que comecem com um cifrão (\$). Se você usar operações de publicação ou assinatura incompatível com tópicos reservados, isso poderá resultar no fim da conexão.

Tópico	Operação do cliente permitida	Descrição
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> /checkins	Publicar	<p>O software Edge Agent publica informações de status do veículo neste tópico.</p> <p>As informações de status do veículo são trocadas em formato de buffers de protocolo (protobuf). Para obter mais informações, consulte o Guia do <a href="#">desenvolvedor de FleetWise</a></p>

Tópico	Operação do cliente permitida	Descrição
		<a href="#">software do Edge Agent para AWS IoT.</a>
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /signals</code>	Publicar	<p>O software Edge Agent publica sinais para este tópico.</p> <p>As informações de sinal são trocadas no formato de buffers de protocolo (protobuf). Para obter mais informações, consulte o Guia do <a href="#">desenvolvedor de FleetWise software do Edge Agent para AWS IoT.</a></p>
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /collection_schemes</code>	Assinar	AWS A IoT FleetWise publica esquemas de coleta de dados sobre esse tópico. Os veículos consomem esses esquemas de coleta de dados.
<code>\$aws/iotfleetwise/vehicles/<i>vehicleName</i> /decoder_manifests</code>	Assinar	AWS A IoT FleetWise publica manifestos de decodificadores para esse tópico. Os veículos consomem esses manifestos do decodificador.

# Criar um veículo

Você pode usar o FleetWise console de AWS IoT ou a API para criar um veículo.

## Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar modelos de veículos](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

## Tópicos

- [Criar um veículo \(console\)](#)
- [Criar um veículo \(AWS CLI\)](#)
- [Criar vários veículos \(AWS CLI\)](#)

## Criar um veículo (console)

Você pode usar o FleetWise console de AWS IoT para criar um veículo.

## Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar modelos de veículos](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

## Para criar um veículo

1. Abra o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Veículos.
3. Na página de resumo do veículo, selecione Criar veículo e execute as etapas a seguir.

### Tópicos

- [Etapa 1: definir as propriedades do veículo](#)
- [Etapa 2: configurar o certificado do veículo](#)
- [Etapa 3: anexar políticas ao certificado](#)
- [Etapa 4: revisar e criar](#)

## Etapa 1: definir as propriedades do veículo

Nesta etapa, você dá um nome para o veículo e o associa ao manifesto do modelo e ao manifesto do decodificador.

1. Insira um nome para o veículo.

### Important

Um veículo corresponde a qualquer AWS IoT coisa. Se já existir alguma coisa com esse nome, selecione Associar o veículo a uma coisa de IoT para atualizar a coisa com o veículo. Ou escolha um nome de veículo diferente e a AWS IoT FleetWise criará automaticamente algo novo para o veículo.

2. Selecione um modelo de veículo (manifesto do modelo) na lista.
3. Selecione um manifesto de decodificador na lista. O manifesto do decodificador está associado ao modelo do veículo.
4. (Opcional) Para associar atributos do veículo, escolha Adicionar atributos. Se você ignorar esta etapa, deverá adicionar atributos após a criação do veículo antes de poder implementá-lo em campanhas.
5. (Opcional) Para associar etiquetas ao veículo, selecione Adicionar nova etiqueta. Você também poderá adicionar tags após a criação do veículo.
6. Escolha Próximo.

## Etapa 2: configurar o certificado do veículo

Para usar seu veículo como uma AWS IoT coisa, você deve configurar um certificado de veículo com uma política anexada. Se você ignorar esta etapa, deverá configurar um certificado após a criação do veículo antes de poder implantá-lo em campanhas.

1. Selecione Gerar automaticamente um novo certificado (recomendado).
2. Escolha Próximo.

## Etapa 3: anexar políticas ao certificado

Anexar uma política ao certificado que você configurou na etapa anterior.

1. Em Políticas, insira um nome de política existente. Para criar uma política, selecione Criar política.
2. Escolha Próximo.

## Etapa 4: revisar e criar

Verifique as configurações do veículo e selecione Criar veículo.

### Important

Depois que o veículo for criado, você deverá baixar o certificado e as chaves. Você usará o certificado e a chave privada para conectar o veículo no software Edge Agent for AWS IoT FleetWise .

## Criar um veículo (AWS CLI)

Ao criar um veículo, você deve usar um modelo de veículo associado a um manifesto do decodificador. Você pode usar a operação [CreateVehicle](#) da API para criar um veículo. O exemplo a seguir usa a AWS CLI.

### Important

Antes de começar, verifique o seguinte:

- Você deve ter um modelo de veículo, e o status dele deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar modelos de veículos](#).
- O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser ACTIVE. Para ter mais informações, consulte [Criar e gerenciar manifestos de decodificadores](#).

Para criar um veículo, execute o seguinte comando.

Substitua *file-name* pelo nome do arquivo JSON que contém a configuração do veículo.

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

Exemplo configuração do veículo

- (Opcional) O valor `associationBehavior` pode ser um dos seguintes:
  - `CreateIotThing`— Quando seu veículo é criado, AWS a IoT cria FleetWise automaticamente AWS IoT algo com o nome do ID do veículo.
  - `ValidateIotThingExists` — Use uma coisa da AWS IoT existente para criar um veículo.

Para criar AWS IoT algo, execute o comando a seguir. Substitua *thing-name* pelo nome da coisa que você quer criar.

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

Se não for especificado, a AWS IoT cria FleetWise automaticamente AWS IoT algo para o seu veículo.

#### Important

Certifique-se de que a AWS IoT coisa seja provisionada após a criação do veículo. Para ter mais informações, consulte [Provisionar veículos](#).

- Substitua *vehicle-name* por um dos seguintes.
  - O nome da sua AWS IoT coisa, se `associationBehavior` estiver configurado para `ValidateIotThingExists`.



- O ID do veículo a ser criado se `associationBehavior` estiver configurado como `CreateIotThing`.

O ID do veículo pode ter de 1 a 100 caracteres. Caracteres válidos: a–z, A–Z, 0–9, traço (-), sublinhado (\_) e dois pontos (:).

- Substitua `model-manifest-ARN` pelo ARN do modelo do veículo (manifesto do modelo).
- Substitua `decoder-manifest-ARN` pelo ARN do manifesto do decodificador associado ao modelo de veículo especificado.
- (Opcional) Você pode adicionar outros atributos para distinguir este veículo de outros veículos criados usando o mesmo modelo de veículo. Por exemplo, se você tiver um carro elétrico, poderá especificar o seguinte valor para um atributo: `{"fuelType": "electric"}`.

#### Important

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

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

## Criar vários veículos (AWS CLI)

Você pode usar a operação [BatchCreateVehicle](#) da API para criar vários veículos ao mesmo tempo. O exemplo a seguir usa a AWS CLI.

Para criar vários veículos, execute o seguinte comando.

Substitua `file-name` pelo nome do arquivo JSON que contém as configurações de vários veículos.

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

## Exemplo configurações de veículos

```
{
  "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"
      }
    }
  ]
}
```

Você pode criar até 10 veículos para cada operação em lote. Para obter mais informações sobre a configuração do veículo, consulte [Criar um veículo \(AWS CLI\)](#).

## Atualizar um veículo (AWS CLI)

Você pode usar a operação [UpdateVehicle](#) da API para atualizar um veículo existente. O exemplo a seguir usa a AWS CLI.

Para atualizar um veículo, execute o seguinte comando.

Substitua *file-name* pelo nome do arquivo JSON que contém a configuração do veículo.

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

### Exemplo configuração do veículo

- Substitua *vehicle-name* pelo ID do veículo que você quer atualizar.

- (Opcional) Substitua *model-manifest-ARN* pelo ARN do modelo de veículo (manifesto de modelo) que você usa para substituir o modelo de veículo em uso.
- (Opcional) Substitua *decoder-manifest-ARN* pelo ARN do manifesto do decodificador associado ao novo modelo de veículo que você especificou.
- (Opcional) *attribute-update-mode* Substitua pelos atributos do veículo.
  - Merge — Mesclar novos atributos em atributos existentes atualizando este último com novos valores e adicionando novos atributos se eles não existirem.

Por exemplo, se um veículo tiver os seguintes atributos: {"color": "black", "fuelType": "electric"}, e você atualizar o veículo com os seguintes atributos: {"color": "", "fuelType": "gasoline", "model": "x"}, o veículo atualizado terá os seguintes atributos: {"fuelType": "gasoline", "model": "x"}.

- Overwrite — Substitua os atributos existentes por novos.

Por exemplo, se um veículo tiver os seguintes atributos: {"color": "black", "fuelType": "electric"} e você atualizar o veículo pelo atributo {"model": "x"}, o veículo atualizado terá o atributo {"model": "x"}.

Isso será necessário se houver atributos na entrada.

- (Opcional) Para adicionar novos atributos ou atualizar os existentes com novos valores, configure *attributes*. Por exemplo, se você tiver um carro elétrico, poderá especificar o seguinte valor para um atributo: {"fuelType": "electric"}.

Para excluir atributos, configure *attributeUpdateMode* como Merge.

#### Important

Os atributos devem ser definidos no modelo de veículo associado antes de poder adicioná-los a veículos individuais.

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

```
}
```

## Atualizar vários veículos (AWS CLI)

Você pode usar a operação [BatchUpdateVehicle](#) da API para atualizar vários veículos existentes ao mesmo tempo. O exemplo a seguir usa a AWS CLI.

Para atualizar vários veículos, execute o seguinte comando.

Substitua *file-name* pelo nome do arquivo JSON que contém as configurações de vários veículos.

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

### Exemplo configurações de veículos

```
{
  "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"
      }
    }
  ]
}
```

Você pode atualizar até 10 veículos para cada operação em lote. Para obter mais informações sobre a configuração de cada veículo, consulte [Atualizar um veículo \(AWS CLI\)](#).

## Excluir um veículo

Você pode usar o FleetWise console de AWS IoT ou a API para excluir veículos.

### Important

Depois que um veículo é excluído, a AWS IoT remove FleetWise automaticamente o veículo das frotas e campanhas associadas. Para obter mais informações, consulte [Criar e gerenciar frotas](#) e [Coletar e transferir dados com campanhas](#). No entanto, o veículo ainda existe como uma coisa ou ainda está associado a uma coisa dentro AWS IoT Core. Para obter instruções sobre como excluir uma coisa, consulte [Excluir uma coisa](#) no Guia do desenvolvedor do AWS IoT Core .

## Excluir um veículo (console)

Você pode usar o FleetWise console de AWS IoT para excluir um veículo.

Para excluir um veículo

1. Navegue até o console [AWS de IoT. FleetWise](#)
2. No painel de navegação, selecione Veículos.
3. Na página Veículos, selecione o botão ao lado do veículo que você quer excluir.
4. Escolha Excluir.
5. Em Excluir **vehicle-name**, insira o nome do veículo e selecione Excluir.

## Excluir um veículo (AWS CLI)

Você pode usar a operação [DeleteVehicle](#) da API para excluir um veículo. O exemplo a seguir usa AWS CLI.

Para excluir um veículo, execute o seguinte comando.

Substitua *vehicle-name* pelo ID do veículo que você quer excluir.

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

## Obter informações do veículo (AWS CLI)

Você pode usar a operação [ListVehicles](#) da API para verificar se um veículo foi excluído. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todos os veículos, execute o seguinte comando.

```
aws iotfleetwise list-vehicles
```

Você pode usar a operação da [GetVehicle](#) API para recuperar as informações do veículo. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de um veículo, execute o seguinte comando.

Substitua *vehicle-name* pelo ID do veículo que você quer recuperar.

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

### Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações no veículo podem não ser refletidas imediatamente.

# Criar e gerenciar frotas

Uma frota representa um grupo de veículos. Uma frota sem veículos associados é uma entidade vazia. Antes de poder usar a frota para gerenciar vários veículos ao mesmo tempo, você deve associar os veículos à frota. Um veículo pode pertencer a várias frotas. Você pode controlar quais dados coletar de uma frota de veículos e quando coletar dados implantando uma campanha. Para obter mais informações, consulte [Coletar e transferir dados com campanhas](#).

Uma frota contém as seguintes informações.

`fleetId`

O ID da frota.

(Opcional) `description`

Uma descrição que ajuda você a encontrar a frota.

`signalCatalogArn`

O nome de recurso da Amazon (ARN) do catálogo de sinais.

O AWS IoT FleetWise fornece as seguintes operações de API que você pode usar para criar e gerenciar frotas.

- [CreateFleet](#) — Cria um grupo de veículos que contém o mesmo grupo de sinais.
- [AssociateVehicleFleet](#) — Associa um veículo a uma frota.
- [DisassociateVehicleFleet](#) — Desassocia um veículo de uma frota.
- [UpdateFleet](#) — Atualiza a descrição de uma frota existente.
- [DeleteFleet](#) — Exclui uma frota existente.
- [ListFleets](#) — Recupera uma lista paginada de resumos de todas as frotas.
- [ListFleetsForVehicle](#) — Recupera uma lista paginada de IDs de todas as frotas às quais o veículo pertence.
- [ListVehiclesInFleet](#) — Recupera uma lista paginada de resumos de todos os veículos de uma frota.
- [GetFleet](#) — Recupera informações sobre uma frota.

Tópicos

- [Criar uma frota \(AWS CLI\)](#)
- [Associar um veículo a uma frota \(AWS CLI\)](#)
- [Desassociar um veículo de uma frota \(AWS CLI\)](#)
- [Atualizar uma frota \(AWS CLI\)](#)
- [Excluir uma frota \(AWS CLI\)](#)
- [Obter informações da frota \(AWS CLI\)](#)

## Criar uma frota (AWS CLI)

Você pode usar a operação da API [CreateFleet](#) para criar uma frota de veículos. O exemplo a seguir usa AWS CLI.

### Important

Você deve ter um catálogo de sinais antes de poder criar uma frota. Para obter mais informações, consulte [Criar um catálogo de sinais \(AWS CLI\)](#).

Para criar uma frota, execute o seguinte comando.

- Substitua *fleet-id* pelo ID da frota que você está criando.

O ID da frota deve ser exclusivo e ter de 1 a 100 caracteres. Caracteres válidos: letras (A–Z e a–z), números (0–9), dois pontos (:), hífen (-) e sublinhados (\_).

- (Opcional) Substitua *description* por uma descrição.

A descrição pode conter de 1 a 2.048 caracteres.

- Substitua *signal-catalog-arn* pelo ARN do catálogo de sinais.

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



## Associar um veículo a uma frota (AWS CLI)

Você pode usar a operação da API [AssociateVehicleFleet](#) para associar um veículo a uma frota. O exemplo a seguir usa AWS CLI.

### Important

- Você deve ter um veículo e uma frota antes de poder associar um veículo a uma frota. Para obter mais informações, consulte [Criar, provisionar e gerenciar veículos](#).
- Se você associar um veículo a uma frota alvo de uma campanha, o AWS IoT FleetWise implantará automaticamente a campanha no veículo.

Para associar um veículo a uma frota, execute o seguinte comando.

- Substitua *fleet-id* pelo ID da frota.
- Substitua *vehicle-name* pelo ID do veículo.

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

## Desassociar um veículo de uma frota (AWS CLI)

Você pode usar a operação da API [DisassociateVehicleFleet](#) para desassociar um veículo de uma frota. O exemplo a seguir usa AWS CLI.

Para desassociar um veículo de uma frota, execute o seguinte comando.

- Substitua *fleet-id* pelo ID da frota.
- Substitua *vehicle-name* pelo ID do veículo.

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

## Atualizar uma frota (AWS CLI)

Você pode usar a operação da API [UpdateFleet](#) para atualizar a descrição de uma frota. O exemplo a seguir usa AWS CLI.

Para atualizar uma frota, execute o comando a seguir.

- Substitua *fleet-id* pelo ID da frota que você está atualizando.
- Substitua *description* por uma nova descrição.

A descrição pode conter de 1 a 2.048 caracteres.

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

## Excluir uma frota (AWS CLI)

Você pode usar a operação da API [DeleteFleet](#) para excluir uma frota. O exemplo a seguir usa AWS CLI.

### Important

Antes de excluir uma frota, confirme se ela não tem veículos associados. Para obter instruções sobre como desassociar um veículo de uma frota, consulte [Desassociar um veículo de uma frota \(AWS CLI\)](#).

Para excluir uma frota, execute o comando a seguir.

Substitua *fleet-id* pelo ID da frota que você está excluindo.

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

## Obter informações da frota (AWS CLI)

Você pode usar a operação da API [ListFleets](#) para verificar se uma frota foi excluída. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todas as frotas, execute o comando a seguir.

```
aws iotfleetwise list-fleets
```

Você pode usar a operação da API [ListFleetsForVehicle](#) para recuperar uma lista paginada de IDs de todas as frotas às quais o veículo pertence. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de IDs de todas as frotas às quais o veículo pertence, execute o seguinte comando.

Substitua *vehicle-name* pelo ID do veículo.

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

Você pode usar a operação da API [ListVehiclesInFleet](#) para recuperar uma lista paginada de resumos de todos os veículos em uma frota. O exemplo a seguir usa a AWS CLI.

Para recuperar uma lista paginada de resumos de todos os veículos de uma frota, execute o seguinte comando.

Substitua *fleet-id* pelo ID da frota.

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

Você pode usar a operação da API [GetFleet](#) para recuperar informações da frota. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de uma frota, execute o comando a seguir.

Substitua *fleet-id* pelo ID da frota.

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

#### Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações na frota podem não ser refletidas imediatamente.

# Coletar e transferir dados com campanhas

Uma campanha é uma orquestração de regras de coleta de dados. As campanhas fornecem ao software Edge Agent do AWS IoT FleetWise instruções sobre como selecionar, coletar e transferir dados para a nuvem.

Você cria as campanhas na nuvem. Após a aprovação das campanhas por você ou por sua equipe, o AWS IoT FleetWise realiza a implantação automática nos veículos. Você pode escolher implantar uma campanha em um veículo ou em uma frota de veículos. O software Edge Agent não inicia a coleta de dados até que uma campanha em execução seja implantada no veículo.

## Note

As campanhas não funcionarão até que você tenha o seguinte.

- O software Edge Agent está em execução no seu veículo. Para obter mais informações sobre como desenvolver, instalar e trabalhar com o software Edge Agent, faça o seguinte.
  1. Acesse o [console do AWS IoT FleetWise](#).
  2. Na página inicial do serviço, na seção Comece a usar o AWS IoT FleetWise, selecione Explorar o Edge Agent.
- Você configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte [Provisionar veículos](#).

Cada item da lista contém as seguintes informações.

`signalCatalogArn`

O nome de recurso da Amazon (ARN) do catálogo de sinais associado à campanha.

(Opcional) `tags`

Tags são metadados que podem ser usados para gerenciar a campanha. Você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados.

`TargetArn`

O ARN de um veículo ou de uma frota na qual a campanha é implantada.

## name

Um nome exclusivo que ajuda a identificar a campanha.

## collectionScheme

Os esquemas de coleta de dados fornecem instruções ao software Edge Agent sobre quais dados coletar ou quando coletá-los. Atualmente, o IoT FleetWise é compatível com o esquema de coleta baseado em condições e o esquema de coleta baseado em tempo.

## conditionBasedCollectionScheme

O esquema de coleta baseado em condições utiliza uma expressão lógica para reconhecer quais dados coletar. O software Edge Agent coleta dados quando a condição é atendida.

## expression

A expressão lógica usada para reconhecer quais dados devem ser coletados. Por exemplo, se a expressão `$variable.`myVehicle.InVehicleTemperature` > 50.0` for especificada, o software Edge Agent coletará valores de temperatura superiores a 50,0. Consulte instruções de como escrever expressões em [Expressões lógicas para campanhas](#).

(Opcional) `triggerMode` pode ser um dos valores a seguir.

- `RISING_EDGE` — O software Edge Agent coleta dados somente quando a condição é atendida pela primeira vez. Por exemplo, `$variable.`myVehicle.AirBagDeployed` == true`.
- `ALWAYS` — O software Edge Agent coleta dados sempre que a condição é atendida.

## (Opcional) `minimumTriggerIntervalMs`

A duração mínima entre dois eventos de coleta de dados, em milissegundos. Se um sinal mudar com frequência, você poderá coletar dados em uma taxa mais lenta.

## (Opcional) `conditionLanguageVersion`

A versão da linguagem de expressão condicional.

## timeBasedCollectionScheme

Ao definir um esquema de coleta baseado em tempo, especifique um período de tempo em milissegundos. O software Edge Agent usa o período de tempo para decidir com que frequência coletar dados. Por exemplo, se o período for de 120.000 milissegundos, o software Edge Agent coletará dados uma vez a cada dois minutos.

### (Opcional) `compression`

Para economizar largura de banda sem fio e reduzir o tráfego de rede, você pode especificar [SNAPPY](#) para compactar dados em veículos.

Por padrão (OFF), o software Edge Agent não compacta dados.

### `dataDestinationConfigs`

Selecione o destino para onde a campanha transferirá os dados do veículo. Você pode optar por armazenar dados no Amazon S3 ou no Amazon Timestream.

S3 é um mecanismo de armazenamento de dados econômico que oferece recursos duráveis de gerenciamento de dados e serviços de dados downstream. Você pode usar o S3 para dados relacionados ao comportamento de direção ou análise de manutenção de longo prazo.

Timestream é um mecanismo de persistência de dados que ajuda a identificar tendências e padrões quase em tempo real. Você pode usar o Timestream para dados de séries temporais, por exemplo, para analisar tendências passadas na velocidade ou frenagem do veículo.

### (Opcional) `dataExtraDimensions`

Você pode adicionar um ou mais atributos para fornecer informações adicionais para um sinal.

### (Opcional) `description`

Você pode adicionar uma descrição para ajudar a identificar o propósito da campanha.

### (Opcional) `diagnosticsMode`

Quando o modo de diagnóstico está configurado para `SEND_ACTIVE_DTCS`, a campanha envia códigos de problemas de diagnóstico (DTCs) padrão armazenados que ajudam a identificar o que há de errado em seu veículo. Por exemplo, P0097 indica que o módulo de controle do motor (ECM) determinou que a entrada do sensor de temperatura do ar de admissão 2 (IAT2) está abaixo da faixa normal do sensor.

Por padrão (OFF), o software Edge Agent não envia códigos de diagnóstico.

### (Opcional) `expiryTime`

Você pode definir a data de expiração da sua campanha. Quando a campanha expira, o software Edge Agent para de coletar dados conforme especificado na campanha. Se diversas campanhas forem implantadas no veículo, o software Edge Agent usará outras campanhas para coletar dados.

Valor padrão: 253402243200 (31 de dezembro de 9999, 00:00:00 UTC)

(Opcional) `postTriggerCollectionDuration`

Você pode definir uma duração de coleta pós-acionamento, para que o software Edge Agent continue coletando dados por um período especificado após um esquema ser invocado. Por exemplo, se um esquema de coleta baseado em condições com a seguinte expressão for invocado: `$variable.`myVehicle.Engine.RPM` > 7000.0`, o software Edge Agent continuará coletando valores de rotações por minuto (RPM) do motor. Mesmo que o RPM ultrapasse 7.000 apenas uma vez, isso poderá indicar que há um problema mecânico. Nesse caso, o software Edge Agent pode continuar coletando dados para ajudar a monitorar a condição.

Valor padrão: 0


(Opcional) `priority`

Você pode especificar um inteiro para indicar o nível de prioridade da campanha. Campanhas com um número menor têm prioridade mais alta. Se você implantar diversas campanhas em um veículo, as campanhas com maior prioridade serão iniciadas primeiro.

Valor padrão: 0

(Opcional) `signalsToCollect`

Uma lista de sinais dos quais os dados são coletados quando o esquema de coleta de dados é invocado.

 **Important**

Os sinais utilizados na expressão para o esquema de coleta baseado em condições devem ser especificados neste campo.

`name`

O nome do sinal do qual os dados são coletados quando o esquema de coleta de dados é invocado.

(Opcional) `maxSampleCount`

O número máximo de amostras de dados que o software Edge Agent coleta e transfere para a nuvem quando o esquema de coleta de dados é chamado.

### (Opcional) `minimumSamplingIntervalMs`

A duração mínima entre dois eventos de coleta de amostras de dados, em milissegundos. Se um sinal muda frequentemente, você pode usar este parâmetro para coletar dados em uma taxa mais lenta.

Intervalo válido: de 0 a 4294967295

### (Opcional) `spoolingMode`

Se `spoolingMode` estiver configurado para `T0_DISK`, o software Edge Agent armazenará temporariamente os dados localmente quando um veículo não estiver conectado à nuvem. Após o restabelecimento da conexão, os dados armazenados localmente serão transferidos automaticamente para a nuvem.

Valor padrão: `OFF`

### (Opcional) `startTime`

Uma campanha aprovada é ativada na hora de início.

Valor padrão: `0`

O status de uma campanha pode ser um dos valores a seguir.

- `CREATING` — AWS IoT FleetWise está processando sua solicitação para criar a campanha.
- `WAITING_FOR_APPROVAL` — Depois que uma campanha é criada, ela entra no estado `WAITING_FOR_APPROVAL`. Para aprovar a campanha, use a operação da API `UpdateCampaign`. Após a aprovação da campanha, o AWS IoT FleetWise implantará automaticamente a campanha no veículo ou frota alvo. Para obter mais informações, consulte [Atualizar uma campanha \(AWS CLI\)](#).
- `RUNNING` — A campanha está ativa.
- `SUSPENDED` — A campanha está suspensa. Para retomar a campanha, use a operação da API `UpdateCampaign`.

O AWS IoT FleetWise fornece as seguintes operações de API que você pode usar para criar e gerenciar campanhas.

- [CreateCampaign](#) — Cria uma campanha.



- [UpdateCampaign](#) — Atualiza uma campanha existente. Após a criação de uma campanha, você deverá usar esta operação de API para aprová-la.
- [DeleteCampaign](#) — Exclui uma campanha existente.
- [ListCampaigns](#) — Recupera uma lista paginada de resumos de todas as campanhas.
- [GetCampaign](#) — Recupera informações sobre uma campanha.

## Tutoriais

- [Criar uma campanha](#)
- [Atualizar uma campanha \(AWS CLI\)](#)
- [Excluir uma campanha](#)
- [Obter informações da campanha \(AWS CLI\)](#)

## Criar uma campanha

Você pode usar o console ou a API do AWS IoT FleetWise para criar campanhas para coletar dados do veículo.

### Important

Para que sua campanha funcione, você deve ter o seguinte:

- O software Edge Agent está em execução no seu veículo. Para obter mais informações sobre como desenvolver, instalar e trabalhar com o software Edge Agent, faça o seguinte:
  1. Acesse o [console do AWS IoT FleetWise](#).
  2. Na página inicial do serviço, na seção Comece a usar o AWS IoT FleetWise, selecione Explorar o Edge Agent.
- Você configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte [Provisionar veículos](#).

## Tópicos

- [Criar uma campanha \(console\)](#)
- [Criar uma campanha \(AWS CLI\)](#)
- [Expressões lógicas para campanhas](#)

## Criar uma campanha (console)

Você pode usar o console do AWS IoT FleetWise para criar uma campanha para selecionar, coletar e transferir dados de veículos para a nuvem.

Como criar uma campanha

1. Acesse [o console do AWS IoT FleetWise](#).
2. No painel de navegação, selecione Campanhas.
3. Na página Campanhas, selecione Criar campanha e, em seguida, conclua as etapas nos tópicos a seguir.

Tópicos

- [Etapa 1: configurar campanha](#)
- [Etapa 2: definir o destino do armazenamento](#)
- [Etapa 3: adicionar veículos](#)
- [Etapa 4: revisar e criar](#)
- [Etapa 5: implantar uma campanha](#)

### Important

- Você deve ter um catálogo de sinalização e um veículo antes de criar uma campanha. Para obter mais informações, consulte [Criar e gerenciar catálogos de sinais](#) e [Criar, provisionar e gerenciar veículos](#).
- Após a criação de uma campanha, você deverá aprová-la. Para obter mais informações, consulte [Etapa 5: implantar uma campanha](#).

## Etapa 1: configurar campanha

Em Informações gerais, faça o seguinte:

1. Insira um nome para a campanha.
2. (Opcional) Insira uma descrição.

Configure o esquema de coleta de dados da campanha. Um esquema de coleta de dados fornece instruções ao software Edge Agent sobre quais dados coletar ou quando coletá-los. No console do AWS IoT FleetWise, você pode configurar um esquema de coleta de dados das seguintes formas:


- Defina manualmente o esquema de coleta de dados.
- Faça upload de um arquivo para definir automaticamente o esquema de coleta de dados.

Em Opção de configuração, selecione uma das seguintes opções:

- Para especificar manualmente o tipo de esquema de coleta de dados e definir opções para personalizar o esquema, selecione Definir esquema de coleta de dados.

Especifique manualmente o tipo de esquema de coleta de dados e defina opções para personalizar o esquema.

1. Na seção Detalhes do esquema de coleta de dados, selecione o tipo de esquema de coleta de dados para usar nesta campanha. Para usar uma expressão lógica para reconhecer quais dados do veículo coletar, selecione Baseado em condição. Para usar um período específico para decidir com que frequência coletar dados do veículo, selecione Baseado em tempo.
2. Defina por quanto tempo a campanha coletará dados.

 Note

Por padrão, uma campanha aprovada é ativada imediatamente e não tem horário de término definido. Para evitar cobranças extras, você deve especificar um intervalo de tempo.

3. Se você tiver especificado um esquema de coleta de dados baseado em condições, deverá definir uma expressão lógica para reconhecer quais dados deverão ser coletados. AWS IoT FleetWise usa uma expressão lógica para reconhecer quais dados coletar para um esquema baseado em condições. A expressão deve especificar o nome completo de um sinal como uma variável, um operador de comparação e um valor de comparação.

Por exemplo, se você especificar a expressão


`$variable.`myVehicle.InVehicleTemperature` > 50.0`, o AWS IoT FleetWise coletará valores de temperatura maiores que 50,0. Consulte instruções de como escrever expressões em [Expressões lógicas para campanhas](#).

Insira a expressão lógica usada para reconhecer quais dados devem ser coletados.

4. (Opcional) Você pode especificar a versão do idioma da expressão condicional. O valor padrão é 1.
  5. (Opcional) Você pode especificar o intervalo mínimo de acionamento, que é a menor duração entre dois eventos de coleta de dados. Por exemplo, se um sinal muda frequentemente, você pode coletar dados em uma taxa mais lenta.
  6. Especifique a condição do Modo de acionamento para que o software Edge Agent colete dados. Por padrão, o software Edge Agent para AWS IoT FleetWise sempre coleta dados cada vez que a condição é atendida. Ou poderá coletar dados somente quando a condição for atendida pela primeira vez, no primeiro acionamento.
  7. Se você tiver especificado um esquema de coleta de dados baseado em tempo, deverá especificar um Período, em milissegundos, de 10.000 a 60.000 milissegundos. O software Edge Agent usa o período de tempo para decidir com que frequência coletar dados.
  8. (Opcional) Você pode editar as Opções avançadas do esquema.
    - a. Para economizar largura de banda sem fio e reduzir o tráfego de rede compactando dados, selecione Snappy.
    - b. (Opcional) Para definir por quanto tempo, em milissegundos, continuará coletando dados após um evento de coleta de dados, você pode especificar a Duração da coleta pós-acionamento.
    - c. (Opcional) Para indicar o nível de prioridade da campanha, você pode especificar a Prioridade da campanha. As campanhas com um número menor de prioridade são implantadas primeiro e são consideradas de prioridade mais alta.
    - d. O software Edge Agent pode armazenar dados localmente temporariamente quando um veículo não está conectado à nuvem. Após o restabelecimento da conexão, os dados armazenados localmente serão transferidos automaticamente para a nuvem. Especifique se você quer que o Edge Agent armazene dados localmente durante uma conexão perdida.
    - e. (Opcional) Para fornecer informações adicionais para um sinal, adicione até cinco atributos como dimensões de dados extras.
- Para fazer upload de um arquivo para definir o esquema de coleta de dados, selecione Carregar um arquivo.json do seu dispositivo local. AWS IoT FleetWise define automaticamente quais opções você pode definir no arquivo. Você pode revisar e atualizar as opções selecionadas.

Faça upload de um arquivo .json com detalhes sobre o esquema de coleta de dados.

1. Para importar informações sobre o esquema de coleta de dados, selecione Escolher arquivos. Para obter mais informações sobre o formato de arquivo necessário, consulte a documentação da API [CreateCampaign](#).


 Note

Atualmente, o AWS IoT FleetWise é compatível com a extensão de formato de arquivo .json.

2. O AWS IoT FleetWise define automaticamente o esquema de coleta de dados com base nas informações do seu arquivo. Analise as opções que o AWS IoT FleetWise selecionou para você. É possível atualizar as opções, se necessário.

Especifique os sinais


Você pode especificar os sinais dos quais coletar dados quando o esquema de coleta de dados for invocado.

 Important

Os sinais utilizados na expressão para o esquema de coleta baseado em condições devem ser especificados neste campo.

Para especificar os sinais dos quais coletar dados

1. Pesquise o nome totalmente qualificado do sinal.

 Note

O nome totalmente qualificado do sinal é o caminho para o sinal mais o nome do sinal. Use um ponto (.) para se referir ao sinal secundário.

Por exemplo,

`Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState`

é o nome totalmente qualificado do atuador `HandsOffSteeringState.Vehicle.Chassis.SteeringWheel.HandsOff`. é o caminho para esse atuador.

2. (Opcional) Em Contagem máxima de amostras, insira o número máximo de amostras de dados que o software Edge Agent coleta e transfere para a nuvem quando o esquema de coleta de dados é invocado.
3. (Opcional) Em Intervalo mínimo de amostragem, insira a duração mínima de tempo entre dois eventos de coleta de amostra de dados, em milissegundos. Se um sinal muda frequentemente, você pode usar este parâmetro para coletar dados em uma taxa mais lenta.
4. Para adicionar outro sinal, selecione Adicionar mais sinais. Você pode adicionar até 999 sinais.
5. Escolha Next (Próximo).

## Etapa 2: definir o destino do armazenamento

### Note

Você só poderá transferir dados do veículo para o Amazon S3 se a campanha contiver sinais de dados de sistemas de visão.

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Escolha o destino onde deseja armazenar os dados coletados pela campanha. É possível transferir dados de veículos para o Amazon S3 ou o Amazon Timestream.

Em Configurações de destino, faça o seguinte:

- Selecione S3 ou Timestream na lista suspensa.

Para armazenar dados de veículos em um bucket S3, selecione Amazon S3. S3 é um serviço de armazenamento de objetos que armazena dados como objetos em buckets. Para obter mais informações, consulte [Criar, configurar e trabalhar com buckets do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

O S3 otimiza o custo do armazenamento de dados e fornece mecanismos adicionais para usar dados de veículos, como data lakes, armazenamento centralizado de dados, pipelines de processamento de dados e análises. Você pode usar o S3 para armazenar dados para processamento e análise em lote. Por exemplo, você pode criar relatórios de eventos de frenagem

brusca para o modelo de machine learning (ML). Os dados recebidos do veículo são armazenados em buffer por 10 minutos antes da entrega.

## Amazon S3

### Important

Você só poderá transferir dados para o S3 se o AWS IoT FleetWise tiver permissões para gravar no bucket do S3. Para obter mais informações sobre como conceder acesso, consulte [Controle do acesso com o AWS IoT FleetWise](#).

Em Configurações de destino do S3, faça o seguinte:

1. Em bucket S3, selecione um bucket para o qual AWS IoT FleetWise tenha permissão.
2. (Opcional) Insira um prefixo personalizado que você possa usar para organizar os dados armazenados no bucket S3.
3. Selecione o formato de saída, que são os arquivos de formato salvos no bucket S3.
4. Selecione para compactar os dados armazenados no bucket S3 como um arquivo .gzip. Recomendamos a compactação de dados porque isso minimiza os custos de armazenamento.
5. As opções selecionadas em Configurações de destino do S3 alteram o URI do objeto S3 de exemplo. Este é um exemplo de quais arquivos são salvos no S3.

Para armazenar dados de veículos em uma tabela Timestream, selecione Amazon Timestream. Você pode usar o Timestream para consultar dados do veículo para identificar tendências e padrões. Por exemplo, você pode usar o Timestream para criar um alarme para o nível de combustível do veículo. Os dados do veículo recebido são transferidos para o Timestream quase que em tempo real. Para obter mais informações, consulte [O que é o Amazon Timestream?](#) no Guia do desenvolvedor do Amazon Timestream.

## Amazon Timestream

### Important

Você só poderá transferir dados para uma tabela se o AWS IoT FleetWise tiver permissões para gravar dados no Timestream. Para obter mais informações sobre como conceder acesso, consulte [Controle do acesso com o AWS IoT FleetWise](#).

Em Configurações da tabela Timestream, faça o seguinte:

1. Em Nome do banco de dados Timestream, selecione o nome do seu banco de dados Timestream na lista suspensa.
2. Em Nome da tabela Timestream, selecione o nome da tabela Timestream na lista suspensa.

Em Acesso ao serviço para Timestream, faça o seguinte:

- Escolha um perfil do IAM na lista suspensa.
- Escolha Next (Próximo).

### Etapa 3: adicionar veículos

Para escolher em quais veículos implantar sua campanha, selecione-os na lista de veículos. Filtre os veículos pesquisando os atributos e seus valores que você adicionou ao criar os veículos ou pelo nome do veículo.

Em Filtrar veículos, faça o seguinte:

1. Na caixa de pesquisa, encontre o atributo ou nome do veículo e escolha-o na lista.

#### Note

Cada atributo pode ser usado apenas uma vez.

2. Insira o valor do atributo ou o nome do veículo no qual você deseja implementar a campanha. Por exemplo, se o nome completo do atributo for `fuelType`, insira `gasoline` como o valor.
3. Para procurar outro atributo do veículo, repita as etapas anteriores. Você pode pesquisar até cinco atributos de veículos e um número ilimitado de nomes de veículos.
4. Os veículos que correspondem à sua pesquisa estão informados em Nome do veículo. Escolha os veículos nos quais deseja que a campanha seja implantada.

#### Note

Até 100 veículos são exibidos nos resultados da pesquisa. Escolha Seleccionar tudo para adicionar todos os veículos à campanha.



## 5. Escolha Next (Próximo).

### Etapa 4: revisar e criar

Verifique as configurações da campanha e selecione Criar campanha.

#### Note

Após a criação de uma campanha, você ou sua equipe deverão implantá-la nos veículos.

### Etapa 5: implantar uma campanha

Depois de criar uma campanha, você ou sua equipe deverão implantá-la nos veículos.

Para implantar uma campanha

1. Na página Resumo da campanha, selecione Implantar.
2. Revise e confirme que deseja iniciar a implantação e começar a coletar dados dos veículos conectados à campanha.
3. Escolha Implantar.

Para pausar a coleta de dados de veículos conectados à campanha, na página Resumo da campanha, selecione Suspende. Para retomar a coleta de dados dos veículos conectados à campanha, selecione Continuar.

## Criar uma campanha (AWS CLI)

Você também pode usar a operação da API [CreateCampaign](#) para criar um grupo de trabalho. O exemplo a seguir usa a AWS CLI.

Ao criar uma campanha, os dados coletados dos veículos podem ser armazenados no Amazon S3 (S3) ou no Amazon Timestream. Selecione Timestream para obter um banco de dados de séries temporais rápido, escalável e sem servidor, e para armazenar dados que precisam de processamento quase que em tempo real. Selecione o S3 para um armazenamento de objetos com escalabilidade, disponibilidade de dados, segurança e desempenho líderes do setor.

**⚠ Important**

Você só poderá transferir dados de veículos se o AWS IoT FleetWise tiver permissões para gravar dados no S3 ou no Timestream. Para obter mais informações sobre como conceder acesso, consulte [Controle do acesso com o AWS IoT FleetWise](#).

## Criar campanha

**⚠ Important**

- Você deve ter um catálogo de sinalização e um veículo ou frota antes de criar uma campanha. Para obter mais informações, consulte [Criar e gerenciar catálogos de sinais](#), [Criar, provisionar e gerenciar veículos](#) e [Criar e gerenciar frotas](#).
- Após a criação de uma campanha, você deverá usar a operação da API UpdateCampaign para aprovar a campanha. Para obter mais informações, consulte [Atualizar uma campanha \(AWS CLI\)](#).

Para criar uma campanha, execute o seguinte comando.

Substitua *file-name* pelo nome do arquivo JSON que contém a configuração da campanha.

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

- Substitua *campaign-name* pelo nome da campanha que você está criando.
- Substitua *signal-catalog-arn* pelo nome do recurso da Amazon (ARN) do catálogo de sinais.
- Substitua *target-arn* pelo ARN de uma frota ou veículo que você criou.
- Substitua *bucket-arn* pelo ARN do bucket do 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"
        }
    }
]
}

```

- Substitua *campaign-name* pelo nome da campanha que você está criando.
- Substitua *signal-catalog-arn* pelo nome do recurso da Amazon (ARN) do catálogo de sinais.
- Substitua *target-arn* pelo ARN de uma frota ou veículo que você criou.
- Substitua *role-arn* pelo ARN da função de execução de tarefas que concede permissão ao AWS IoT FleetWise para entregar dados à tabela Timestream.
- Substitua *table-arn* pelo ARN da tabela 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"
      }
    }
  ]
}
```

## Expressões lógicas para campanhas

O AWS IoT FleetWise usa uma expressão lógica para reconhecer quais dados devem ser coletados como parte de uma campanha. Para ter mais informações sobre expressões, consulte [Expressões](#) no Guia do desenvolvedor do AWS IoT Events.

A variável de expressão deve ser construída de acordo com as regras do tipo de dados que estão sendo coletados. Para dados do sistema de telemetria, a variável de expressão deve ser o nome totalmente qualificado do sinal. Para dados de sistemas de visão, a expressão combina o nome totalmente qualificado do sinal com o caminho que vai do tipo de dados do sinal até uma de suas propriedades.

Por exemplo, se o catálogo de sinais contém os seguintes nós:

```
{
  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"
}
```

Se os nós seguem a definição do ROS 2:

```
{
  Vehicle.ADAS.CameraStruct.msg:
    boolean obstaclesExists
    uint8[] image
    Obstacle[30] obstacles
}
{
  Vehicle.ADAS.Obstacle.msg:
    float32: probability
    uint8 o_type
    float32: distance
}
```

Estas são todas as variáveis possíveis de expressão de eventos:

```
{
  ...
  $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`
}
```

## Atualizar uma campanha (AWS CLI)

Você pode usar a operação da API [UpdateCampaign](#) para atualizar uma campanha existente. Use o seguinte comando AWS CLI.

- Substitua *campaign-name* pelo nome da campanha que você está atualizando.
- Substitua *action* por um dos seguintes:
  - APPROVE — Aprova a campanha para permitir que o AWS IoT FleetWise o implemente em um veículo ou frota.
  - SUSPEND — Suspende a campanha. A campanha é excluída dos veículos, e todos os veículos na campanha suspensa deixarão de enviar dados.
  - RESUME — Reativa a campanha SUSPEND. A campanha será reimplantada em todos os veículos, e os veículos retomarão o envio de dados.
  - UPDATE — Atualiza a campanha definindo atributos e associando-os a um sinal.

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

## Excluir uma campanha

Você pode usar o console do AWS IoT FleetWise ou a API para excluir campanhas.

### Excluir uma campanha (console)

Para excluir uma campanha, use o console do AWS IoT FleetWise.

Excluir uma campanha

1. Acesse o [console do AWS IoT FleetWise](#).
2. No painel de navegação, selecione Campanhas.
3. Na página Campanhas, selecione a campanha alvo.
4. Escolha Delete (Excluir).
5. Em Excluir **campaign-name?**, insira o nome da campanha a ser excluída e selecione Confirmar.

### Excluir uma campanha (AWS CLI)

Você pode usar a operação da API [DeleteCampaign](#) para excluir uma campanha. O exemplo a seguir usa AWS CLI.

Para excluir uma campanha, execute o seguinte comando.

Substitua *campaign-name* pelo nome do veículo que você está excluindo.

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

### Obter informações da campanha (AWS CLI)

Você pode usar a operação da API [ListCampaigns](#) para verificar se uma campanha foi excluída. O exemplo a seguir usa a AWS CLI.

Execute os seguintes comandos para recuperar uma lista paginada de resumos de todas as campanhas.


```
aws iotfleetwise list-campaigns
```

Você pode usar a operação da API [GetCampaign](#) para recuperar as informações do veículo. O exemplo a seguir usa a AWS CLI.

Para recuperar os metadados de uma campanha, execute o comando a seguir.

Substitua *campaign-name* pelo nome da campanha que você deseja recuperar.

```
aws iotfleetwise get-campaign --name campaign-name
```

 Note

Essa operação é [eventualmente consistente](#). Em outras palavras, as alterações na campanha podem não ser refletidas imediatamente.



# Processamento e visualização de dados do veículo

O software Edge Agent para AWS IoT FleetWise transfere dados de veículos selecionados para o Amazon Timestream ou Amazon Simple Storage Service (Amazon S3). Depois que os dados chegarem ao destino, você poderá usar outros serviços da AWS para visualizá-los e compartilhá-los.

## Processamento dos dados do veículo no Timestream

O Timestream é um banco de dados de séries temporais totalmente gerenciado que pode armazenar e analisar trilhões de pontos de dados de séries temporais por dia. Seus dados são armazenados em uma tabela Timestream gerenciada pelo cliente. Você pode usar o Timestream para consultar dados de veículos para obter insights sobre os veículos. Para obter mais informações, consulte [O que é o Amazon Timestream?](#)

O esquema padrão dos dados que são transferidos para o Timestream contém os seguintes campos:

Nome do campo	Tipo de dados	Descrição
eventId	varchar	O ID do evento de coleta de dados.
vehicleName	varchar	O ID do veículo do qual os dados foram coletados.
name	varchar	O nome da campanha que o software Edge Agent usa para coletar dados.
time	timestamp	A marcação de data/hora do ponto de dados.
measure_name	varchar	O nome do sinal.
measure_value::bigint	bigint	Valores de sinal do tipo Inteiro.

Nome do campo	Tipo de dados	Descrição
measure_value::double	duplo	Valores de sinal do tipo Duplo.
measure_value::boolean	boolean	Valores de sinal do tipo Booleano.

## Visualização de dados do veículo armazenados no Timestream

Depois que os dados do veículo forem transferidos para o Timestream, você poderá usar os seguintes serviços da AWS para visualizar, monitorar, analisar e compartilhar seus dados.

- Visualize e monitore dados em painéis usando o [Grafana ou Amazon Managed Grafana](#). Você pode visualizar dados de diversas fontes da AWS (como Amazon CloudWatch e Timestream) e outras fontes de dados com um único painel da Grafana.
- Analise e visualize dados em painéis usando o [Amazon QuickSight](#).

## Processamento de dados do veículo no S3

Amazon S3 é um serviço de armazenamento de objetos que armazena e protege qualquer quantidade de dados. É possível usar o S3 para diversos casos de uso, como data lakes, backup e restauração, arquivamento, aplicações corporativas, dispositivos de AWS IoT e análise de big data. Seus dados são armazenados no S3 como objetos em buckets. Para obter mais informações, consulte [O que é a Amazon S3?](#)

O esquema padrão de dados transferidos para o Amazon S3 contém os campos a seguir.

Nome do campo	Tipo de dados	Descrição
eventId	varchar	O ID do evento de coleta de dados.
vehicleName	varchar	O ID do veículo do qual os dados foram coletados.

Nome do campo	Tipo de dados	Descrição
name	varchar	O nome da campanha que o software Edge Agent usa para coletar dados.
time	timestamp	A marcação de data/hora do ponto de dados.
measure_name	varchar	O nome do sinal.
measure_value_BIGINT	bigint	Valores de sinal do tipo Inteiro.
measure_value_DOUBLE	duplo	Valores de sinal do tipo Duplo.
measure_value_BOOLEAN	boolean	Valores de sinal do tipo Booleano.
measure_value_STRUCT	struct	Valores de sinal do tipo Struct.

## Formato de objeto S3

O AWS IoT FleetWise transfere os dados do veículo para o S3, onde são salvos como um objeto. É possível usar o URI do objeto que identifica os dados de forma exclusiva para encontrar dados da campanha. O formato do URI do objeto do S3 depende se os dados coletados são processados ou não estruturados.

### Dados não estruturados

Os dados não estruturados são armazenados no S3 de uma forma não predefinida. Eles podem estar em vários formatos, como imagens ou vídeos.

As mensagens do veículo passadas para o AWS IoT FleetWise com dados de sinal dos arquivos do Amazon Ion são decodificadas e transferidas para o S3 como objetos. Os objetos do S3 representam cada sinal e são codificados em binário.

O URI do objeto do S3 de dados não estruturados usa o seguinte formato:

```
s3://bucket-name/prefix/unstructured-data/random-ID-yyyy-MM-dd-HH-mm-ss-SSS-vehicleName-signalName-fieldName
```

## Dados processados

Os dados processados são armazenados no S3 e passam por etapas de processamento que validam, enriquecem e transformam as mensagens. Listas de objetos e velocidade são exemplos de dados processados.

Os dados transferidos para o S3 são armazenados como objetos que representam registros que foram armazenados em buffer por um período de cerca de 10 minutos. Por padrão, o AWS IoT FleetWise adiciona um prefixo de hora UTC no formato `year=YYYY/month=MM/date=DD/hour=HH` antes de gravar objetos no S3. Esse prefixo cria uma hierarquia lógica no bucket, no qual cada barra (/) cria um nível na hierarquia. Os dados processados também contêm o URI do objeto do S3 para dados não estruturados.

O URI do objeto do S3 de dados processados usa o seguinte formato:

```
s3://bucket-name/prefix/processed-data/year=YYYY/month=MM/day=DD/hour=HH/part-0000-random-ID.gz.parquet
```

## Dados brutos

Os dados brutos, também conhecidos como dados primários, são dados coletados dos arquivos do Amazon Ion. É possível usar dados brutos para solucionar qualquer problema ou encontrar a causa raiz dos erros.

O URI do objeto do S3 de dados brutos usa o seguinte formato:

```
s3://bucket-name/prefix/raw-data/vehicle-name/eventID-timestamp.10n
```

## Analisar dados do veículo armazenados no S3

Depois que os dados do veículo forem transferidos para o S3, você poderá usar os serviços da AWS a seguir para monitorar, analisar e compartilhar dados.

Extraia e analise dados usando o Amazon SageMaker para fluxos de trabalho de rotulagem downstream e machine learning (ML).

Para ter mais informações, consulte os tópicos a seguir no Guia do desenvolvedor do Amazon SageMaker.

- [Processar dados](#)
- [Treinar modelos de machine learning](#)
- [Rotular imagens](#)

Catalogue seus dados usando Crawler do AWS Glue e analise-os no Amazon Athena. Por padrão, os objetos gravados no S3 têm partições de tempo no estilo Apache Hive, com caminhos de dados que contêm pares de valores-chave conectados por sinais de igual.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do Amazon Athena:

- [Particionamento de dados no Athena](#)
- [Use AWS Glue para se conectar a fontes de dados no Amazon S3](#)
- [Melhores práticas ao usar o Athena com AWS Glue](#)

Visualize dados usando o Amazon QuickSight lendo diretamente a tabela do Athena ou o bucket do S3.

### Tip

Se você estiver lendo diretamente do S3, confirme se os dados do veículo estão no formato JSON, pois o Amazon QuickSight não é compatível com o formato Apache Parquet.

Para obter mais informações, consulte os seguintes tópicos no Guia do usuário do Amazon QuickSight:

- [Fonte de dados compatíveis](#)

- [Criação de fonte de dados](#)

# AWS CLI e AWS SDKs

Esta seção fornece informações sobre como fazer solicitações da API do AWS IoT FleetWise.

Para obter mais informações sobre [operações e tipos de dados](#) do AWS IoT FleetWise, consulte a Referência da API do AWS IoT FleetWise.

Para usar o AWS IoT FleetWise com diversas linguagens de programação, use os [AWS SDKs](#), que contêm as seguintes funcionalidades automáticas:

- Assinar criptograficamente suas solicitações de serviço
- Recuperar solicitações
- Lidar com respostas de erro

Para acesso à linha de comando, use o AWS IoT FleetWise com o [AWS CLI](#). Você pode controlar o AWS IoT, o FleetWise e seus outros serviços usando a linha de comando e automatizá-los por meio de scripts.

# Solução de problemas do AWS IoT FleetWise

Use as informações e soluções de solução de problemas nesta seção para ajudar a resolver problemas no AWS IoT FleetWise.

As informações a seguir podem ajudar a solucionar problemas comuns com o AWS IoT FleetWise.

## Tópicos

- [Problemas com o manifesto do decodificador](#)
- [Problemas do software Edge Agent do AWS IoT FleetWise](#)

## Problemas com o manifesto do decodificador

Solucione problemas com manifesto do decodificador.

Diagnosticar chamadas de API do manifesto do decodificador

Erro	Diretrizes de solução de problemas
<code>UpdateOperationFailure.ConflictingDecoderUpdate</code>	O mesmo manifesto do decodificador tem várias solicitações de atualização. Espere e tente novamente.
<code>UpdateOperationFailure.InternalFailure</code>	<code>InternalFailure</code> é lançado como uma exceção encapsulada. O problema em si depende da exceção encapsulada.
<code>UpdateOperationFailure.ActiveDecoderUpdate</code>	O manifesto do decodificador está em um estado <code>Active</code> e não pode ser atualizado. Altere o estado do manifesto do decodificador para <code>DRAFT</code> e tente novamente.
<code>UpdateOperationFailure.ConflictingModelUpdate</code>	O AWS IoT FleetWise está tentando fazer a validação em relação a um modelo de veículo (manifesto do modelo) que está sendo modificado por outra pessoa. Espere e tente novamente.



Erro	Diretrizes de solução de problemas
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND</pre>	<p>O modelo do veículo não tem nenhum sinal associado a ele. Adicione sinais ao modelo do veículo e verifique se eles podem ser encontrados no catálogo de sinais associado.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE</pre>	<p>Atualize o modelo do veículo para que ele fique no estado ACTIVE e tente novamente.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse : FailureReason.MODEL_NOT_FOUND</pre>	<p>O AWS IoT FleetWise não conseguiu encontrar o modelo de veículo associado ao manifesto do decodificador. Verifique o nome do recurso da Amazon (ARN) do modelo do veículo e tente novamente.</p>
<pre>UpdateOperationFailure.Mode ManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE)</pre>	<p>A validação do modelo do veículo falhou porque os nomes dos sinais do modelo do veículo não foram encontrados no catálogo de sinais. Verifique se os sinais no modelo do veículo estão todos incluídos no catálogo de sinais associado.</p>
<pre>UpdateOperationFailure.ValidationFailure</pre>	<p>Sinais ou interfaces de rede que não são válidos foram encontrados na solicitação para atualizar o manifesto do decodificador. Verifique se todos os sinais e interfaces de rede retornados pela exceção existem, se todos os sinais usados estão associados a uma interface disponível e se você não removerá uma interface que tenha sinais associados a ela.</p>

Erro	Diretrizes de solução de problemas
<code>UpdateOperationFailure.KmsKeyAccessDenied</code>	Há um problema de permissão na chave do AWS Key Management Service (AWS KMS) usada para a operação. Verifique se você está usando um perfil que tenha acesso à chave e tente novamente.
<code>UpdateOperationFailure.DecoderDoesNotExist</code>	O manifesto do decodificador não existe. Verifique o nome manifesto do decodificador e tente novamente.

As mensagens de erro de dados de sistemas de visão com o motivo `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` incluirão uma dica na resposta que fornece informações sobre por que a solicitação falhou. Você pode usar a dica para determinar quais diretrizes de solução de problemas seguir.

#### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

### Diagnosticar a validação de dados de sistemas de visão do manifesto do decodificador

Erro	Diretrizes de solução de problemas
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</code>	O AWS IoT FleetWise não encontrou a estrutura do sinal raiz usada no decodificador de sinal usando o catálogo de sinais. Verifique se o sinal raiz da estrutura está definido corretamente no catálogo de sinais.
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</code>	Uma mensagem primitiva no catálogo de sinais não foi definida com o mesmo tipo de dados na solicitação de atualização do manifesto do decodificador. Verifique se as mensagens primitivas definidas na solicitação

Erro	Diretrizes de solução de problemas
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</code>	<p>correspondem à definição do catálogo de sinais correspondente.</p> <p>O número de propriedades definidas em uma struct no catálogo de sinais não corresponde ao número de propriedades que você está tentando decodificar no manifesto do decodificador. Verifique se você tem o número correto de sinais para decodificar comparando-o com os sinais definidos no catálogo de sinais.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>O AWS IoT FleetWise encontrou um sinal definido como STRUCT no catálogo de sinais sem uma <code>structuredMessageDefinition</code> definida na solicitação do manifesto do decodificador. Certifique-se de que cada struct esteja definida como uma <code>structuredMessageDefinition</code> na solicitação de atualização do manifesto do decodificador.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>O sinal raiz da estrutura usada no manifesto do decodificador não está definido adequadamente como uma estrutura no catálogo de sinais. A estrutura do sinal raiz usada no manifesto do decodificador deve ter o campo <code>structFullyQualifiedName</code> definido. Ela também precisa de um nó STRUCT com esse <code>fullyQualifiedName</code>.</p>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	<p>Uma das mensagens de folha usadas na solicitação do manifesto do decodificador não está definida como uma mensagem primitiva. Verifique se todos os objetos de folha na solicitação estão definidos como mensagens primitivas.</p>

Erro	Diretrizes de solução de problemas
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>Um objeto de matriz no catálogo de sinais não foi definido como <code>structuredMessageListDefinition</code> na solicitação de atualização do manifesto do decodificador. Verifique se todas as propriedades da matriz estão definidas como <code>structuredMessageListDefinition</code> na solicitação de atualização do manifesto do decodificador.</p>

## Problemas do software Edge Agent do AWS IoT FleetWise

Solucione problemas do software Edge Agent.

### Problemas

- [Problema: o software Edge Agent não inicia.](#)
- [Problema: \[ERROR\] \[IoT FleetWise Engine::connect\]: \[Failed to init persistency library\]](#)
- [Problema: o software Edge Agent não coleta PIDs de diagnóstico integrado \(OBD\) II e códigos de problemas de diagnóstico \(DTCs\).](#)
- [Problema: o software Edge Agent do AWS IoT FleetWise não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.](#)
- [Problema: \[ERROR\] \[AwsIotConnectivityModule::connect\]: \[Connection failed with error\] ou \[WARN\] \[AwsIotChannel::send\]: \[No alive MQTT Connection.\]](#)

### Problema: o software Edge Agent não inicia.

Pode ser que você veja os seguintes erros quando o software Edge Agent não inicia.

- ```
Error from reader: * Line 1, Column 1
Syntax error: value, object or array expected.
```

Solução: verifique se o arquivo de configuração do software Edge Agent do AWS IoT FleetWise está usando um formato JSON válido. Por exemplo, certifique-se de que as vírgulas estão sendo

usadas de forma correta. Para obter mais informações sobre o arquivo de configuração, faça o seguinte para baixar o Guia do desenvolvedor do software Edge Agent do AWS IoT FleetWise.

1. Acesse o [console do AWS IoT FleetWise](#).
2. Na página inicial do serviço, na seção Comece a usar o AWS IoT FleetWise, selecione Explorar o Edge Agent.

```
[ERROR] [SocketCANBusChannel::connect]: [ SocketCan with name xxx is not accessible]
[ERROR] [IoTFleetWiseEngine::connect]: [ Failed to Bind Consumers to Producers ]
```

Solução: esse erro pode ser exibido quando o software Edge Agent não consegue estabelecer comunicação de soquete com as interfaces de rede definidas no arquivo de configuração.

Para verificar se todas as interfaces de rede definidas na configuração estão disponíveis, execute o seguinte comando:

```
ip link show
```

Para colocar uma interface de rede online, execute o seguinte comando: Substitua *network-interface-id* pelo ID da interface de rede.

```
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]
```

Solução: esse erro pode ser exibido quando o software Edge Agent não consegue estabelecer conexão MQTT com AWS IoT Core. Verifique se os itens a seguir estão configurados corretamente e reinicie o software Edge Agent.

- `mqtConnection::endpointUrl`— Endpoint do dispositivo IoT da conta da AWS.
- `mqtConnection::clientId`— O ID do veículo no qual o software Edge Agent está em execução.
- `mqtConnection::certificateFilename`— O caminho para o arquivo de certificado do veículo.

- `mqttConnection::privateKeyFilename`— O caminho para o arquivo da chave privada do veículo.
- Você usou AWS IoT Core para provisionar o veículo. Para obter mais informações, consulte [Provisionar veículos](#).

Para obter mais informações, consulte [Perguntas frequentes sobre o AWS IoT Device SDK for C++](#).

## Problema: [ERROR] [IoTFleetWiseEngine::connect]: [Failed to init persistency library]

Solução: esse erro pode ser exibido quando o software Edge Agent não consegue localizar o armazenamento de persistência. Verifique se o seguinte está configurado corretamente e reinicie o software Edge Agent.

`persistency::persistencyPath` — Um caminho local usado para persistir esquemas de coleta, manifestos de decodificadores e instantâneos de dados.

## Problema: o software Edge Agent não coleta PIDs de diagnóstico integrado (OBD) II e códigos de problemas de diagnóstico (DTCs).

Solução: esse erro pode ser exibido quando `obdInterface::pidRequestIntervalSeconds` ou `obdInterface::dtcRequestIntervalSeconds` estão configurados como 0.

Se o software Edge Agent estiver sendo executado em um veículo com transmissão automática, verifique se `obdInterface::hasTransmissionEcu` está configurado para `true`.

Se o veículo aceitar IDs de arbitragem de Controller Area Network (barramento CAN) estendida, verifique se `obdInterface::useExtendedIds` está configurado para `true`.

## Problema: o software Edge Agent do AWS IoT FleetWise não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.

Solução: esse erro pode ser exibido quando as cotas padrão são violadas.

| Recurso                                                   | Cota                                           | Ajustável | Observação                                                                                                                                                                           |
|-----------------------------------------------------------|------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Valor do ID do sinal                                      | O ID do sinal deve ser menor ou igual a 50.000 | Sim       | O software Edge Agent não coletará dados de sinais que tenham um ID superior a 50.000. Recomendamos verificar quantos sinais o catálogo de sinais contém antes de alterar essa cota. |
| Número de esquemas ativos de recolha de dados por veículo | 256                                            | Sim       | Recomendamos que você verifique quantas campanhas criou na nuvem e quantos esquemas cada campanha contém antes de alterar essa cota.                                                 |
| Tamanho do buffer do histórico do sinal                   | 20 MB                                          | Sim       | Se a cota for violada, o software Edge Agent para de coletar novos dados.                                                                                                            |

**Problema:** [ERROR] [AwsIotConnectivityModule::connect]: [Connection failed with error] ou [WARN] [AwsIotChannel::send]: [No alive MQTT Connection.]

**Solução:** esse erro pode ser exibido quando o software Edge Agent não está conectado à nuvem. Por padrão, o software Edge Agent envia uma solicitação de ping a AWS IoT Core a cada minuto e aguarda por 3 minutos. Se não houver resposta, o software Edge Agent restabelecerá automaticamente a conexão com a nuvem.

# Segurança na AWS IoT FleetWise

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O [modelo de responsabilidade compartilhada](#) descreve isso como a segurança da nuvem e a segurança na nuvem:

- **Segurança da nuvem** — AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de [AWS](#) de . Para saber mais sobre os programas de conformidade que se aplicam à AWS IoT FleetWise, consulte [Serviços da AWS no escopo do programa de conformidade Serviços da](#) .
- **Segurança na nuvem** — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade dos dados, os requisitos da empresa e as leis e os regulamentos aplicáveis

Esta documentação ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a AWS IoT FleetWise. Ele mostra como configurar a AWS IoT para atender FleetWise aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos de AWS IoT FleetWise .

## Conteúdo

- [Proteção de dados na AWS IoT FleetWise](#)
- [Controlando o acesso com AWS IoT FleetWise](#)
- [Identity and Access Management para AWS IoT FleetWise](#)
- [Validação de conformidade para AWS IoT FleetWise](#)
- [Resiliência na IoT AWS FleetWise](#)
- [Segurança da infraestrutura na AWS IoT FleetWise](#)
- [Análise de configuração e vulnerabilidade na AWS IoT FleetWise](#)
- [Melhores práticas de segurança para AWS IoT FleetWise](#)



# Proteção de dados na AWS IoT FleetWise

O [modelo de responsabilidade AWS compartilhada](#) de se aplica à proteção de dados na AWS IoT FleetWise. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as [Perguntas frequentes sobre privacidade de dados](#). Para ter mais informações sobre a proteção de dados na Europa, consulte a [AWS postagem do blog Shared Responsibility Model and GDPR](#) no AWS Blog de segurança da.

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte [Federal Information Processing Standard \(FIPS\) 140-2](#).

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de email dos seus clientes, em marcações ou campos de formato livre, como um campo Name (Nome). Isso inclui quando você trabalha com a AWS IoT FleetWise ou outra coisa Serviços da AWS usando o console, a API ou AWS os AWS CLI SDKs. Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos

fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

AWS A IoT FleetWise deve ser usada com um Edge Agent que você desenvolve e instala no hardware compatível do veículo para transmitir dados do veículo para a AWS nuvem. A extração de dados de veículos pode estar sujeita a regulamentos de privacidade de dados em determinadas jurisdições. Antes de usar a AWS IoT FleetWise e instalar seu Edge Agent, é altamente recomendável que você avalie suas obrigações de conformidade de acordo com a lei aplicável. Isso inclui requisitos legais aplicáveis para fornecer avisos de privacidade legalmente adequados e obter os consentimentos necessários para extrair dados do veículo.

## Criptografia inativa

Os dados coletados de um veículo são transmitidos para a nuvem por meio de uma AWS IoT Core mensagem com o protocolo de mensagens MQTT. AWS A IoT FleetWise entrega os dados ao seu banco de dados Amazon Timestream. No Timestream, seus dados são criptografados. Todos Serviços da AWS criptografam dados em repouso por padrão.

A criptografia em repouso se integra com AWS Key Management Service (AWS KMS) para gerenciar a chave de criptografia usada para criptografar seus dados. Você pode optar por usar uma chave gerenciada pelo cliente para criptografar os dados coletados pela AWS IoT FleetWise. Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS KMS. Para obter mais informações, consulte [O que é AWS Key Management Service?](#) no Guia do AWS Key Management Service desenvolvedor.

## Criptografia em trânsito

Todos os dados trocados com os AWS IoT serviços são criptografados em trânsito usando o Transport Layer Security (TLS). Para obter mais informações, consulte [Segurança de transporte](#) no Guia do desenvolvedor do AWS IoT .

Além disso, AWS IoT Core oferece suporte à [autenticação](#) e [autorização](#) para ajudar a controlar com segurança o acesso aos recursos de IoT AWS . FleetWise Os veículos podem usar certificados X.509 para serem autenticados (conectados) para usar a AWS IoT FleetWise e usar AWS IoT Core políticas para serem autorizados (ter permissões) para realizar ações específicas. Para ter mais informações, consulte [the section called “Provisionar veículos”](#).

## Criptografia de dados

A criptografia de dados se refere à proteção de dados em trânsito (enquanto viajam de e para a AWS FleetWise IoT e entre gateways e servidores) e em repouso (enquanto são armazenados em dispositivos locais ou em). Serviços da AWS Você pode proteger os dados em repouso usando criptografia do lado do cliente.

### Note

AWS O processamento de FleetWise borda da IoT expõe APIs hospedadas em FleetWise gateways de AWS IoT e acessíveis pela rede local. Essas APIs são expostas por meio de uma conexão TLS apoiada por um certificado de servidor de propriedade do conector IoT Edge. AWS FleetWise Para autenticação do cliente, essas APIs utilizam uma senha de controle de acesso. A chave privada do certificado do servidor e a senha de controle de acesso são armazenadas em disco. AWS O processamento de FleetWise borda da IoT depende da criptografia do sistema de arquivos para a segurança dessas credenciais em repouso.

Para obter mais informações sobre criptografia do lado do servidor e criptografia do lado do cliente, consulte os tópicos a seguir.

### Conteúdo

- [Criptografia inativa](#)
- [Gerenciamento de chaves](#)

## Criptografia inativa

AWS A IoT FleetWise armazena seus dados na AWS nuvem e em gateways.

### Dados em repouso na AWS nuvem

AWS A IoT FleetWise armazena dados em outros Serviços da AWS que criptografam dados em repouso por padrão. A criptografia em repouso se integra com [AWS Key Management Service \(AWS KMS\)](#) para gerenciar a chave de criptografia usada para criptografar os valores das propriedades do seu ativo e agregar valores na IoT. AWS FleetWise Você pode optar por usar uma chave gerenciada pelo cliente para criptografar valores de propriedades de ativos e valores agregados na AWS IoT. FleetWise Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS KMS.

Você pode escolher uma chave gerenciada pelo cliente Chave pertencente à AWS ou uma chave gerenciada pelo cliente para criptografar seus dados.

### Como funciona

A criptografia em repouso se AWS KMS integra ao gerenciamento da chave de criptografia usada para criptografar seus dados.

- Chave pertencente à AWS — Chave de criptografia padrão. AWS A IoT FleetWise possui essa chave. Você não pode visualizar, gerenciar ou usar essa chave em Conta da AWS. Você também não pode ver as operações na chave nos AWS CloudTrail registros. Você pode usar essa chave sem nenhum custo adicional.
- Chave gerenciada pelo cliente — A chave é armazenada na conta que você cria, tem e gerencia. Você tem controle total sobre a chave KMS. AWS KMS Taxas adicionais se aplicam.

### Chaves pertencentes à AWS

Chaves pertencentes à AWS não estão armazenados em sua conta. Elas fazem parte de uma coleção de chaves KMS que AWS possui e gerencia para uso em várias Contas da AWS. Serviços da AWS pode ser usado Chaves pertencentes à AWS para proteger seus dados.

Você não pode visualizar, gerenciar Chaves pertencentes à AWS, usar ou auditar seu uso. No entanto, você não precisa realizar nenhuma ação nem alterar nenhum programa para proteger as chaves que criptografam seus dados.

Você não pagará nenhuma taxa se usar Chaves pertencentes à AWS, e elas não contam nas AWS KMS cotas da sua conta.

### Chaves gerenciadas pelo cliente

Chaves gerenciadas pelo cliente são chaves do KMS disponíveis na sua conta do que você cria, detém e gerencia. Você tem controle total sobre essas chaves KMS, como as seguintes:

- Estabelecer e manter as políticas de chave, políticas do IAM e concessões
- Habilitá-las e desabilitá-las
- Alternar o material criptográfico
- Adicionar etiquetas
- Criar aliases que se referem a elas

- Agendá-las para exclusão

Você também pode usar o CloudTrail Amazon CloudWatch Logs para rastrear as solicitações que a AWS IoT FleetWise envia AWS KMS em seu nome.

Se você estiver usando chaves gerenciadas pelo cliente, deverá conceder FleetWise acesso de AWS IoT à chave KMS armazenada em sua conta. A AWS IoT FleetWise usa criptografia de envelope e hierarquia de chaves para criptografar dados. A chave de criptografia do AWS KMS é usada para criptografar a chave raiz dessa hierarquia de chaves. Para obter mais informações, consulte [Criptografia envelopada](#) no Guia do desenvolvedor do AWS Key Management Service .

O exemplo de política a seguir concede FleetWise permissões de AWS IoT para criar uma chave gerenciada pelo cliente em seu nome.

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

#### Important

Ao adicionar as novas seções à sua política de chaves do KMS, não altere nenhuma seção existente na política. A AWS IoT não FleetWise pode realizar operações em seus dados se a criptografia estiver habilitada para AWS IoT FleetWise e se qualquer uma das afirmações a seguir for verdadeira:

- A chave KMS está desativada ou excluída.
- A política de chave KMS não está configurada corretamente para o serviço.

## Usar dados de sistemas de visão com criptografia em repouso

### Note

O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.

Se você tiver criptografia gerenciada pelo cliente com AWS KMS chaves habilitadas em sua FleetWise conta de AWS IoT e quiser usar dados do sistema de visão, redefina suas configurações de criptografia para serem compatíveis com tipos de dados complexos. Isso permite que a AWS IoT FleetWise estabeleça as permissões adicionais necessárias para os dados do sistema de visão.

### Note

O manifesto do decodificador poderá ficar preso em um status de validação se você não tiver redefinido as configurações de criptografia dos dados de sistemas de visão.

1. Use a operação [GetEncryptionConfiguration](#) da API para verificar se a AWS KMS criptografia está ativada. Nenhuma ação adicional será necessária se o tipo de criptografia for `FLEETWISE_DEFAULT_ENCRYPTION`.
2. Se o tipo de criptografia for `KMS_BASED_ENCRYPTION`, use a operação da [PutEncryptionConfiguration](#) API para redefinir o tipo de criptografia para `FLEETWISE_DEFAULT_ENCRYPTION`.

```
{
  aws iotfleetwise put-encryption-configuration --encryption-type
    FLEETWISE_DEFAULT_ENCRYPTION
}
```

3. Use a operação [PutEncryptionConfiguration](#) da API para reativar o tipo de criptografia para `KMS_BASED_ENCRYPTION`.

```
{
```

```
aws iotfleetwise put-encryption-configuration \
  --encryption-type "KMS_BASED_ENCRYPTION"
  --kms-key-id kms_key_id
}
```

Para ter mais informações sobre como habilitar a criptografia, consulte [Gerenciamento de chaves](#).

## Gerenciamento de chaves


### AWS Gerenciamento de chaves na FleetWise nuvem de IoT

Por padrão, a AWS IoT FleetWise usa Chaves gerenciadas pela AWS para proteger seus dados no. Nuvem AWS Você pode atualizar suas configurações para usar uma chave gerenciada pelo cliente para criptografar dados na AWS FleetWise IoT. Você pode criar, gerenciar e visualizar sua chave de criptografia por meio de AWS Key Management Service (AWS KMS).

AWS A IoT FleetWise oferece suporte à criptografia do lado do servidor com chaves gerenciadas pelo cliente armazenadas AWS KMS para criptografar dados para os seguintes recursos.

| AWS Recurso de IoT FleetWise            | Tipo de dados | Campos criptografados em repouso com chaves gerenciadas pelo cliente |
|-----------------------------------------|---------------|----------------------------------------------------------------------|
| Catálogo de sinais                      |               | description                                                          |
|                                         | Atributo      | description, allowedValues, defaultVa<br>lue, min, max               |
|                                         | Actuator      | description, allowedValues, min, max                                 |
|                                         | Sensor        | description, allowedValues, min, max                                 |
| Modelo do veículo (manifesto do modelo) |               | description                                                          |
| Manifesto do decodificador              |               | description                                                          |
|                                         | CanInterface  | protocolName, protocolVersion                                        |

| AWS Recurso de IoT FleetWise | Tipo de dados                   | Campos criptografados em repouso com chaves gerenciadas pelo cliente                                                                 |
|------------------------------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
|                              | ObdInterface                    | requestMessageId, dtcReques<br>tInterval Segundos hasTransm<br>issionEcu, Padrão OBD, Segundos,<br>pidRequestInterval useExtendedIds |
|                              | CanSignal                       | factor,, isSigned isBigEndian,<br>comprimento, messageID, offset,<br>startBit                                                        |
|                              | ObdSignal                       | ByteLength, offset, pid, escalabil<br>idade, modo de serviço pidRespon<br>seLength, startByte, bitMaskLength<br>bitRightShift        |
| Veículo                      |                                 | attributes                                                                                                                           |
| Campanha                     |                                 | description                                                                                                                          |
|                              | conditionBasedCollectionEsquema | expressão, minimumTriggerInte<br>rval Ms conditionLanguageVersion,<br>TriggerMode                                                    |
|                              | TimeBasedCollectionScheme       | periodMs                                                                                                                             |

 Note

Outros dados e recursos são criptografados usando a criptografia padrão com chaves gerenciadas pela AWS IoT FleetWise. Essa chave é criada e armazenada na conta de AWS IoT. FleetWise

Para obter mais informações, consulte [O que é AWS Key Management Service?](#) no Guia do AWS Key Management Service desenvolvedor.



## Habilitar criptografia usando chaves KMS (console)

Para usar chaves gerenciadas pelo cliente com a AWS IoT FleetWise, você deve atualizar suas configurações de IoT AWS . FleetWise

Para ativar a criptografia usando chaves KMS (console)

1. Abra o console [AWS de IoT. FleetWise](#)
2. Acesse Configurações.
3. Em Criptografia, selecione Editar para abrir a página Editar criptografia.
4. Em Tipo de chave de criptografia, escolha Escolher uma AWS KMS chave diferente. Isso permite a criptografia com chaves gerenciadas pelo cliente armazenadas na AWS KMS.

### Note

Você só pode usar a criptografia de chave gerenciada pelo cliente para recursos de AWS IoT FleetWise . Isso inclui o catálogo de sinais, modelo do veículo (manifesto do modelo), manifesto do decodificador, veículo, frota e campanha.

5. Selecione a chave KMS com uma das seguintes opções:
  - Para usar uma chave KMS existente — Selecione o alias da chave KMS na lista.
  - Para criar uma nova chave KMS — Escolha Criar uma AWS KMS chave.

### Note

Isso abre o AWS KMS console. Para obter mais informações sobre como criar uma chave KMS, consulte [Criação de chaves](#) no Guia do desenvolvedor do AWS Key Management Service .

6. Selecione Salvar para atualizar as configurações.

## Ativar criptografia usando chaves KMS (AWS CLI)

Você pode usar a operação da [PutEncryptionConfiguration](#) API para ativar a criptografia para sua conta de AWS IoT FleetWise . O exemplo a seguir usa AWS CLI.

Para ativar a criptografia, execute o seguinte comando.

- Substitua o *ID da chave KMS* pelo ID da chave KMS.

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

### Example Retorno

```
{  
  "kmsKeyId": "customer_kms_key_id",  
  "encryptionStatus": "PENDING",  
  "encryptionType": "KMS_BASED_ENCRYPTION"  
}
```

### Política de chaves do KMS

Depois de criar uma chave KMS, você deve, no mínimo, adicionar a seguinte declaração à sua política de chaves KMS para que ela funcione com a IoT AWS . 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": "*"   
}
```

Para obter mais informações sobre a edição de uma política de chaves do KMS para uso com AWS a FleetWise IoT, [consulte Alterando uma política de chaves](#) no Guia AWS Key Management Service do desenvolvedor.

### Important

Ao adicionar as novas seções à sua política de chaves do KMS, não altere nenhuma seção existente na política. A AWS IoT não FleetWise pode realizar operações em seus dados se a criptografia estiver habilitada para AWS IoT FleetWise e se qualquer uma das afirmações a seguir for verdadeira:

- A chave KMS está desativada ou excluída.
- A política de chave KMS não está configurada corretamente para o serviço.

## Controlando o acesso com AWS IoT FleetWise

As seções a seguir abordam como controlar o acesso de e para seus AWS IoT FleetWise recursos. As informações que eles abrangem incluem como conceder acesso ao seu aplicativo para que a AWS IoT FleetWise possa transferir dados do veículo durante as campanhas. Eles também descrevem como você pode conceder AWS IoT FleetWise acesso ao seu bucket Amazon S3 (S3) ou ao banco de dados e tabela do Amazon Timestream para armazenar dados.

A tecnologia para gerenciar todas essas formas de acesso é AWS Identity and Access Management (IAM). Para obter mais informações sobre o IAM, consulte [O que é o IAM?](#)

### Conteúdo

- [Conceder AWS IoT FleetWise acesso a um destino do Amazon S3](#)
- [Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream](#)

## Conceder AWS IoT FleetWise acesso a um destino do Amazon S3

Quando você usa um destino do Amazon S3, AWS IoT FleetWise entrega os dados do veículo para o seu bucket do S3 e, opcionalmente, pode usar uma AWS KMS chave que você possui para criptografia de dados. Se o registro de erros estiver ativado, AWS IoT FleetWise também enviará erros de entrega de dados para seu grupo de CloudWatch registros e fluxos. É necessário ter um perfil do IAM ao criar um fluxo de entrega.

AWS IoT FleetWise usa uma política de bucket com o principal de serviço para o destino S3. Para obter mais informações sobre como adicionar políticas de bucket, consulte [Adicionar uma política de bucket usando o console do Amazon S3](#) no Guia do usuário do Amazon Simple Storage Service.

Use a política de acesso a seguir para permitir o acesso AWS IoT FleetWise ao seu bucket do S3. Se você não tiver o bucket do S3, adicione `s3:PutObjectAcl` à lista de ações do Amazon S3. Isso concede ao proprietário do bucket acesso total aos objetos entregues por AWS IoT FleetWise. Para obter mais informações sobre como proteger o acesso a objetos em seus buckets, consulte [Exemplos de políticas de bucket](#) no Guia do usuário do 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"
        }
      }
    }
  ]
}
```

A política de bucket a seguir é para todas as campanhas em uma conta em uma AWS região.

```
{
  "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"
        }
      }
    }
  ]
}
```

Se você tiver uma chave KMS anexada ao seu bucket S3, a chave precisará da política a seguir. Para obter informações sobre gerenciamento de chaves, consulte [Proteção de dados usando criptografia do lado do servidor com AWS Key Management Service chaves \(SSE-KMS\)](#) no Guia do usuário do Amazon Simple Storage Service.

```
{
  "Version": "2012-10-17",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "key-arn"
}
```

### Important

Quando você cria um bucket, o S3 cria uma lista de controle de acesso (ACL) padrão que concede ao proprietário do recurso controle total sobre ele. Se a AWS IoT não FleetWise puder fornecer dados para o S3, certifique-se de desativar a ACL no bucket do S3. Para ter mais informações, consulte [Desabilitar ACLs de todos os novos buckets e aplicar Object Ownership](#) no Guia do usuário do Amazon Simple Storage Service.

## Conceder AWS IoT FleetWise acesso a um destino do Amazon Timestream

Quando você usa um destino Timestream, AWS IoT FleetWise entrega os dados do veículo para uma tabela Timestream. Você deve anexar as políticas à função do IAM para permitir o envio de dados AWS IoT FleetWise para o Timestream.

Se você usa o console para [criar uma campanha](#), a AWS IoT anexa FleetWise automaticamente a política necessária à função.

Antes de começar, verifique o seguinte:

### Important

- Você deve usar a mesma AWS região ao criar recursos de Timestream para IoT AWS . FleetWise Se você mudar de AWS região, poderá ter problemas para acessar os recursos do Timestream.

- AWS A IoT FleetWise está disponível no Leste dos EUA (Norte da Virgínia) e na Europa (Frankfurt).
- Para ver a lista de regiões compatíveis, consulte [Endpoints e cotas do Timestream](#) na Referência geral da AWS.

- É necessário ter um banco de dados do Timestream. Para ver um tutorial, consulte [Criar um banco de dados](#) no Guia do desenvolvedor do Amazon Timestream.
- Você deve ter uma tabela criada no banco de dados do Timestream especificado. Para ver um tutorial, consulte [Criar uma tabela](#) no Guia do desenvolvedor do Amazon Timestream.

Você pode usar o AWS CLI para criar uma função do IAM com uma política de confiança para o Timestream. Para criar um perfil do IAM, execute o comando a seguir.

Para criar um perfil do IAM com uma política de confiança

- *TimestreamExecutionRole* Substitua pelo nome da função que você está criando.
- Substituir *trust-policy* pelo arquivo JSON que contém a política de confiança.

```
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"
        ]
    }
}
]
}

```

Crie uma política de permissões para dar FleetWise permissões à AWS IoT para gravar dados no Timestream. Para criar uma política de permissões, execute o seguinte comando.

Para criar uma política de permissões

- *AWSIoTfleetwiseAccessTimestreamPermissionsPolicy* Substitua pelo nome da política que você está criando.
- Substitua *permissions-policy* pelo nome do arquivo JSON que contém a política de permissões.

```
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": "*"    
  }  
]  
}
```

Para anexar a política de permissões ao perfil do IAM

1. Na saída, copie o nome de recurso da Amazon (ARN) da política de permissões.
2. Para anexar a política de permissões do perfil do IAM, execute o comando a seguir.
  - *permissions-policy-arn* Substitua pelo ARN que você copiou na etapa anterior.
  - *TimestreamExecutionRole* Substitua pelo nome da função do IAM que você criou.

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-  
name TimestreamExecutionRole
```

Para obter mais informações, consulte [Gerenciamento de acesso para recursos do AWS](#) no Guia do usuário do IAM.

## Identity and Access Management para AWS IoT FleetWise

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos de IoT AWS . FleetWise O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

### Tópicos

- [Público](#)
- [Autenticando com identidades](#)
- [Gerenciamento do acesso usando políticas](#)
- [Como a AWS IoT FleetWise funciona com o IAM](#)
- [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

- [Solução de problemas AWS de FleetWise identidade e acesso à IoT](#)

## Público

A forma como você usa o AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz na AWS IoT FleetWise.

**Usuário do serviço** — Se você usa o FleetWise serviço de AWS IoT para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais FleetWise recursos de AWS IoT para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudar você a solicitar as permissões corretas ao seu administrador. Se você não conseguir acessar um recurso na AWS IoT FleetWise, consulte. [Solução de problemas AWS de FleetWise identidade e acesso à IoT](#)

**Administrador de serviços** — Se você é responsável pelos FleetWise recursos de AWS IoT em sua empresa, provavelmente tem acesso total à IoT AWS . FleetWise É seu trabalho determinar quais FleetWise recursos e recursos de AWS IoT seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com a AWS IoT FleetWise, consulte. [Como a AWS IoT FleetWise funciona com o IAM](#)

**Administrador do IAM** — Se você for administrador do IAM, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso à AWS IoT FleetWise. Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT que você pode usar no IAM, consulte. [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte [Como fazer login Conta da AWS](#) no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte [Assinatura de solicitações de AWS API](#) no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte [Autenticação multifator](#) no Guia do usuário do AWS IAM Identity Center . [Usar a autenticação multifator \(MFA\) na AWS](#) no Guia do usuário do IAM.

## Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário-raiz para tarefas diárias. Proteja as credenciais do usuário-raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário-raiz, consulte [Tarefas que exigem credenciais de usuário-raiz](#) no Guia do usuário do IAM.

## Identidade federada

Como prática recomendada, exija que usuários humanos, incluindo usuários que precisam de acesso de administrador, usem a federação com um provedor de identidade para acessar Serviços da AWS usando credenciais temporárias.

Uma identidade federada é um usuário do seu diretório de usuários corporativo, de um provedor de identidade da web AWS Directory Service, do diretório do Identity Center ou de qualquer usuário que acesse usando credenciais fornecidas Serviços da AWS por meio de uma fonte de identidade. Quando as identidades federadas são acessadas Contas da AWS, elas assumem funções, e as funções fornecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos usar o [AWS IAM Identity Center](#). Você pode criar usuários e grupos no IAM Identity Center ou pode se conectar e sincronizar com um conjunto de usuários e grupos em sua própria fonte de identidade para uso em todos os seus Contas da AWS aplicativos. Para obter mais informações sobre o Centro de Identidade do IAM, consulte [“What is IAM Identity Center?” \(O que é o Centro de Identidade do IAM?\)](#) no [AWS IAM Identity Center Guia do usuário](#) do [AWS IAM Identity Center](#).

## Grupos e usuários do IAM

Um [usuário do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para obter mais informações, consulte [Altere as chaves de acesso regularmente para casos de uso que exijam credenciais](#) de longo prazo no [Guia do usuário do IAM](#).

Um [grupo do IAM](#) é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar atributos do IAM.

Usuários são diferentes de perfis. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte [Quando criar um usuário do IAM \(em vez de um perfil\)](#) no [Guia do usuário do IAM](#).

## Perfis do IAM

Uma [função do IAM](#) é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console [trocando de funções](#). Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando uma URL personalizada. Para obter mais informações sobre métodos para o uso de perfis, consulte [Usar perfis do IAM](#) no [Guia do usuário do IAM](#).

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- **Acesso de usuário federado:** para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para obter mais informações sobre perfis para federação, consulte [Criar um perfil para um provedor de identidades de terceiros](#) no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte [Conjuntos de permissões](#) no AWS IAM Identity Center Guia do usuário do .
- **Permissões temporárias para usuários do IAM:** um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- **Acesso entre contas:** é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre perfis e políticas baseadas em atributo para acesso entre contas, consulte [Como os perfis do IAM diferem das políticas baseadas em atributo](#) no Guia do usuário do IAM.
- **Acesso entre serviços —** Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões da entidade principal de chamada, usando um perfil de serviço ou uma função vinculada ao serviço.
- **Sessões de acesso direto (FAS) —** Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).
- **Perfil de serviço:** um perfil de serviço é um perfil do IAM [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais

informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

- **Função vinculada ao serviço** — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.
- **Aplicativos em execução no Amazon EC2** — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. É preferível fazer isso e armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém a função e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para mais informações, consulte [Usar um perfil do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2](#) no Guia do usuário do IAM.

Para saber se deseja usar as funções do IAM, consulte [Quando criar um perfil do IAM \(em vez de um usuário\)](#) no Guia do usuário do IAM.

## Gerenciamento do acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada na AWS como documentos JSON. Para obter mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte [Visão geral das políticas JSON](#) no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

## Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário do IAM, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte [Escolher entre políticas gerenciadas e políticas em linha](#) no Guia do usuário do IAM.

## Políticas baseadas em recursos

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em atributos são políticas em linha que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

## Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte [Visão geral da lista de controle de acesso \(ACL\)](#) no Guia do desenvolvedor do Amazon Simple Storage Service.

## Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- **Limites de permissões:** um limite de permissões é um recurso avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (perfil ou usuário do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em atributo que especificam o usuário ou o perfil no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações sobre limites de permissões, consulte [Limites de permissões para identidades do IAM](#) no Guia do usuário do IAM.
- **Políticas de controle de serviço (SCPs)** — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente várias Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para obter mais informações sobre o Organizações e SCPs, consulte [Como os SCPs funcionam](#) no Guia do usuário do AWS Organizations .
- **Políticas de sessão:** são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em atributo. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para obter mais informações, consulte [Políticas de sessão](#) no Guia do usuário do IAM.



## Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte [Lógica de avaliação de políticas](#) no Guia do usuário do IAM.

## Como a AWS IoT FleetWise funciona com o IAM

Antes de usar o IAM para gerenciar o acesso à AWS IoT FleetWise, saiba quais recursos do IAM estão disponíveis para uso com a IoT AWS . FleetWise

Recursos do IAM que você pode usar com a AWS IoT FleetWise

| Atributo do IAM                                  | AWS Suporte à IoT FleetWise |
|--------------------------------------------------|-----------------------------|
| <a href="#">Políticas baseadas em identidade</a> | Sim                         |
| <a href="#">Políticas baseadas em recursos</a>   | Não                         |
| <a href="#">Ações de políticas</a>               | Sim                         |
| <a href="#">atributos de políticas</a>           | Sim                         |
| <a href="#">Chaves de condição de políticas</a>  | Sim                         |
| <a href="#">ACLs</a>                             | Não                         |
| <a href="#">ABAC (tags em políticas)</a>         | Parcial                     |
| <a href="#">Credenciais temporárias</a>          | Sim                         |
| <a href="#">Permissões de entidade principal</a> | Sim                         |
| <a href="#">Perfis de serviço</a>                | Não                         |
| <a href="#">Funções vinculadas ao serviço</a>    | Não                         |

Para ter uma visão de alto nível de como a AWS FleetWise IoT e AWS outros serviços funcionam com a maioria dos recursos do IAM, [AWS consulte os serviços que funcionam com o IAM no Guia](#) do usuário do IAM.

## Políticas baseadas em identidade para IoT AWS FleetWise

|                                                   |     |
|---------------------------------------------------|-----|
| É compatível com políticas baseadas em identidade | Sim |
|---------------------------------------------------|-----|

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte [Criar políticas do IAM](#) no Guia do usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou atributos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. Não é possível especificar a entidade principal em uma política baseada em identidade porque ela se aplica ao usuário ou função à qual ela está anexado. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte [Referência de elementos da política JSON do IAM](#) no Guia do Usuário do IAM.

### Exemplos de políticas baseadas em identidade para IoT AWS FleetWise

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte. [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Políticas baseadas em recursos na IoT AWS FleetWise

|                                                  |     |
|--------------------------------------------------|-----|
| Oferece suporte a políticas baseadas em recursos | Não |
|--------------------------------------------------|-----|

Políticas baseadas em recursos são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico.

Para o atributo ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse atributo e em que condições. Você deve [especificar uma entidade principal](#) em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Para permitir o acesso entre contas, você pode especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em atributo. Adicionar uma entidade principal entre contas à política baseada em atributo é apenas metade da tarefa de estabelecimento da relação de confiança. Quando o principal e o recurso são diferentes Contas da AWS, um administrador do IAM na conta confiável também deve conceder permissão à entidade principal (usuário ou função) para acessar o recurso. Eles concedem permissão ao anexar uma política baseada em identidade para a entidade. No entanto, se uma política baseada em atributo conceder acesso a uma entidade principal na mesma conta, nenhuma política baseada em identidade adicional será necessária. Para obter mais informações, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

## Ações políticas para AWS IoT FleetWise

|                                      |     |
|--------------------------------------|-----|
| Oferece suporte a ações de políticas | Sim |
|--------------------------------------|-----|

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

Para ver uma lista de FleetWise ações de AWS IoT, consulte [Ações definidas pela AWS IoT FleetWise](#) na Referência de autorização de serviço.

As ações de política na AWS IoT FleetWise usam o seguinte prefixo antes da ação:

```
iotfleetwise
```

Para especificar várias ações em uma única instrução, separe-as com vírgulas.

```
"Action": [  
  "iotfleetwise:action1",  
  "iotfleetwise:action2"  
]
```

Você também pode especificar várias ações usando caracteres-curinga (\*). Por exemplo, para especificar todas as ações que começam com a palavra `List`, inclua a seguinte ação:

```
"Action": "iotfleetwise:List*"
```

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Recursos de políticas para AWS IoT FleetWise

|                                          |     |
|------------------------------------------|-----|
| Oferece suporte a atributos de políticas | Sim |
|------------------------------------------|-----|

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Resource` de política JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando [Nome do recurso da Amazon \(ARN\)](#). Isso pode ser feito para ações que oferecem suporte a um tipo de atributo específico, conhecido como permissões em nível de atributo.

Para ações não compatíveis com permissões no nível de recurso, como operações de listagem, use um curinga (\*) para indicar que a instrução se aplica a todos os recursos.

```
"Resource": "*"
```

Para ver uma lista dos tipos de FleetWise recursos de AWS IoT e seus ARNs, consulte [Recursos definidos pela AWS IoT FleetWise](#) na Referência de autorização de serviço. Para saber com quais

ações você pode especificar o ARN de cada recurso, consulte [Ações definidas pela IoT AWS](#).

## FleetWise

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Chaves de condição de política para AWS IoT FleetWise

|                                                                      |     |
|----------------------------------------------------------------------|-----|
| Compatível com chaves de condição de política específicas do serviço | Sim |
|----------------------------------------------------------------------|-----|

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual principal pode executar ações em quais recursos, e em que condições.

O elemento `Condition` (ou `Condition` bloco de) permite que você especifique condições nas quais uma instrução está em vigor. O elemento `Condition` é opcional. É possível criar expressões condicionais que usam [agentes de condição](#), como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação.

Se você especificar vários elementos `Condition` em uma instrução ou várias chaves em um único `Condition` elemento, a AWS os avaliará usando uma operação lógica AND. Se você especificar vários valores para uma única chave de condição, AWS avalia a condição usando uma OR operação lógica. Todas as condições devem ser atendidas para que as permissões da instrução sejam concedidas.

Você também pode usar variáveis de espaço reservado ao especificar as condições. Por exemplo, é possível conceder a um usuário do IAM permissão para acessar um atributo somente se ele estiver marcado com seu nome de usuário do IAM. Para obter mais informações, consulte [Elementos de política do IAM: variáveis e tags](#) no Guia do usuário do IAM.

AWS suporta chaves de condição globais e chaves de condição específicas do serviço. Para ver todas as chaves de condição AWS globais, consulte as [chaves de contexto de condição AWS global](#) no Guia do usuário do IAM.

Para ver uma lista de chaves de FleetWise condição de AWS IoT, consulte [Chaves de condição para AWS IoT FleetWise](#) na Referência de autorização de serviço. Para saber com quais ações e recursos você pode usar uma chave de condição, consulte [Ações definidas pela AWS IoT FleetWise](#)

Para ver exemplos de políticas FleetWise baseadas em identidade de AWS IoT, consulte [Exemplos de políticas baseadas em identidade para IoT AWS FleetWise](#)

## Listas de controle de acesso (ACLs) na IoT AWS FleetWise

Oferece suporte a ACLs

Não

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

## Controle de acesso baseado em atributos (ABAC) com IoT AWS FleetWise

Oferece suporte a ABAC (tags em políticas)

Parcial

O controle de acesso baseado em recurso (ABAC) é uma estratégia de autorização que define permissões com base em recursos. Em AWS, esses atributos são chamados de tags. Você pode anexar tags a entidades do IAM (usuários ou funções) e a vários AWS recursos. A marcação de entidades e atributos é a primeira etapa do ABAC. Em seguida, você cria políticas de ABAC para permitir operações quando a tag da entidade principal corresponder à tag do atributo que ela está tentando acessar.

O ABAC é útil em ambientes que estão crescendo rapidamente e ajuda em situações em que o gerenciamento de políticas se torna um problema.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no [elemento de condição](#) de uma política usando as chaves de condição `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou `aws:TagKeys`.

Se um serviço oferecer suporte às três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço oferecer suporte às três chaves de condição somente para alguns tipos de recursos, o valor será Parcial.

Para obter mais informações sobre o ABAC, consulte [O que é ABAC?](#) no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte [Usar controle de acesso baseado em atributos \(ABAC\)](#) no Guia do usuário do IAM.

**Note**

AWS FleetWise Só há suporte para `IoTiam:PassRole`, o que é necessário para a operação da `CreateCampaign` API.

## Usando credenciais temporárias com AWS a IoT FleetWise

|                                           |     |
|-------------------------------------------|-----|
| Oferece suporte a credenciais temporárias | Sim |
|-------------------------------------------|-----|

Alguns Serviços da AWS não funcionam quando você faz login usando credenciais temporárias. Para obter informações adicionais, incluindo quais Serviços da AWS funcionam com credenciais temporárias, consulte Serviços da AWS “[Trabalhe com o IAM](#)” no Guia do usuário do IAM.

Você está usando credenciais temporárias se fizer login AWS Management Console usando qualquer método, exceto um nome de usuário e senha. Por exemplo, quando você acessa AWS usando o link de login único (SSO) da sua empresa, esse processo cria automaticamente credenciais temporárias. Você também cria automaticamente credenciais temporárias quando faz login no console como usuário e, em seguida, alterna perfis. Para obter mais informações sobre como alternar perfis, consulte [Alternar para um perfil \(console\)](#) no Guia do usuário do IAM.

Você pode criar manualmente credenciais temporárias usando a AWS API AWS CLI ou. Em seguida, você pode usar essas credenciais temporárias para acessar AWS. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para mais informações, consulte [Credenciais de segurança temporárias no IAM](#).

## Permissões principais entre serviços para IoT AWS FleetWise

|                                                                  |     |
|------------------------------------------------------------------|-----|
| Suporte para o recurso Encaminhamento de sessões de acesso (FAS) | Sim |
|------------------------------------------------------------------|-----|

Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado um principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos

serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte [Encaminhar sessões de acesso](#).

## Funções de serviço para AWS IoT FleetWise

Oferece suporte a perfis de serviço Não

O perfil de serviço é um perfil do IAM [https://docs.aws.amazon.com/IAM/latest/UserGuide/id\\_roles.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles.html) que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte [Criar um perfil para delegar permissões a um AWS service \(Serviço da AWS\)](#) no Guia do usuário do IAM.

### Warning

Alterar as permissões de uma função de serviço pode interromper a funcionalidade AWS da IoT FleetWise. Edite funções de serviço somente quando a AWS IoT FleetWise fornecer orientação para fazer isso.

## Funções vinculadas a serviços para IoT AWS FleetWise

É compatível com perfis vinculados ao serviço Não

Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um AWS service (Serviço da AWS). O serviço pode assumir o perfil de executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte [Serviços do AWS que funcionam com o IAM](#). Encontre um serviço na tabela que inclua um Yes na coluna Função vinculada ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado ao serviço desse serviço.



## Usando funções vinculadas a serviços para o AWS IoT FleetWise

O AWS IoT FleetWise usa funções vinculadas a [serviços AWS Identity and Access Management](#) (IAM). Uma função vinculada a serviços é um tipo exclusivo de função do IAM vinculada diretamente ao AWS IoT. FleetWise As funções vinculadas ao serviço são predefinidas pelo AWS IoT FleetWise e incluem as permissões que o AWS IoT precisa FleetWise para enviar métricas para a Amazon. CloudWatch Para ter mais informações, consulte [Monitoramento do AWS IoT FleetWise com Amazon CloudWatch](#).

Uma função vinculada ao serviço agiliza a configuração do AWS FleetWise IoT porque você não precisa adicionar manualmente as permissões necessárias. O AWS IoT FleetWise define as permissões de suas funções vinculadas a serviços e, a menos que seja definido de outra forma, somente o AWS FleetWise IoT pode assumir suas funções. As permissões definidas incluem a política de confiança e a política de permissões. Essa política de permissões não pode ser anexada a nenhuma outra entidade do IAM.

Um perfil vinculado ao serviço poderá ser excluído somente após excluir seus atributos relacionados. Isso protege seus FleetWise recursos do AWS IoT porque você não pode remover inadvertidamente a permissão para acessar os recursos.

Para obter informações sobre outros serviços compatíveis com perfis vinculados a serviços, consulte [Serviços da AWS compatíveis com o IAM](#) e procure os serviços que mostram Sim na coluna Perfis vinculados a serviços. Para visualizar a documentação da função vinculada ao serviço desse serviço, escolha Sim com um link.

### Permissões de função vinculadas a serviços para AWS IoT FleetWise

O AWS IoT FleetWise usa a função vinculada ao serviço chamada — `AWSServiceRoleForIoT` FleetWise Uma política gerenciada pela AWS que é usada para todas as permissões do out-of-the-box AWS IoT. FleetWise

A função `AWSServiceRoleForIoT` FleetWise vinculada ao serviço confia nos seguintes serviços para assumir a função:

- `IoT` FleetWise

A política de permissões de função nomeada `AWSIoT` FleetWise `ServiceRolePolicy` permite que o AWS IoT FleetWise conclua as seguintes ações nos recursos especificados:

- Ação: `cloudwatch:PutMetricData` no recurso \*

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua uma função vinculada a serviço. Para obter mais informações, consulte [Permissões de perfil vinculado a serviços no Guia do usuário do IAM](#).

### Criação de uma função vinculada a serviços para o AWS IoT FleetWise

Não é necessário criar manualmente uma função vinculada a serviço. Quando você registra uma conta no FleetWise console do AWS IoT, no ou na AWS API AWS CLI, o AWS FleetWise IoT cria a função vinculada ao serviço para você. Para ter mais informações, consulte [Configuração de definições](#).

### Criação de uma função vinculada ao serviço no AWS FleetWise IoT (console)

Não é necessário criar manualmente uma função vinculada a serviço. Quando você registra uma conta no FleetWise console do AWS IoT, na AWS CLI ou na API, o AWS AWS IoT FleetWise cria a função vinculada ao serviço para você.

### Editando uma função vinculada a serviços para o AWS IoT FleetWise

Você não pode editar a função `AWSServiceRoleForIoT FleetWise` vinculada ao serviço no AWS IoT. FleetWise Como várias entidades podem fazer referência a qualquer perfil vinculado ao serviço criado, não é possível alterar o nome do perfil. No entanto, será possível editar a descrição da função usando o IAM. Para ter mais informações, consulte [Editar um perfil vinculado ao serviço](#) no Guia do usuário do IAM.

### Limpar um perfil vinculado ao serviço

Antes de usar o IAM para excluir um perfil vinculado ao serviço, você deverá excluir qualquer recurso usado pelo perfil.

#### Note

Se o AWS IoT FleetWise estiver usando a função quando você tentar excluir os recursos, a exclusão poderá falhar. Se isso acontecer, espere alguns minutos e tente a operação novamente. Para saber como excluir o `service-linked-role` por meio do console, da AWS CLI ou da AWS API, consulte Como [usar funções vinculadas ao serviço](#) no Guia do usuário do IAM.

Se você excluir essa função vinculada ao serviço e precisar criá-la novamente, poderá registrar uma conta no AWS IoT. FleetWise FleetWise Em seguida, o AWS IoT cria a função vinculada ao serviço para você novamente.

## Exemplos de políticas baseadas em identidade para IoT AWS FleetWise

Por padrão, usuários e funções não têm permissão para criar ou modificar recursos de AWS IoT FleetWise. Eles também não podem realizar tarefas usando a AWS API AWS Management Console, AWS Command Line Interface (AWS CLI) ou. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do IAM. O administrador pode então adicionar as políticas do IAM aos perfis, e os usuários podem assumir os perfis.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documento de política JSON, consulte [Criação de políticas do IAM](#) no Guia do Usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pela AWS IoT FleetWise, incluindo o formato dos ARNs para cada um dos tipos de recursos, consulte [Ações, recursos e chaves de condição para AWS IoT FleetWise](#) na Referência de Autorização de Serviço.

### Tópicos

- [Melhores práticas de política](#)
- [Usando o console AWS de IoT FleetWise](#)
- [Permitir que os usuários visualizem suas próprias permissões](#)
- [Acessar recursos no Amazon Timestream](#)

## Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir FleetWise recursos de AWS IoT em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso.

Para obter mais informações, consulte [Políticas gerenciadas pela AWS](#) ou [Políticas gerenciadas pela AWS para perfis de trabalho](#) no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em atributos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para obter mais informações sobre como usar o IAM para aplicar permissões, consulte [Políticas e permissões no IAM](#) no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e atributos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte [Elementos de política JSON do IAM: condições](#) no Manual do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para obter mais informações, consulte [Validação de políticas do IAM Access Analyzer](#) no Guia do usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para obter mais informações, consulte [Configuração de acesso](#) à API protegido por MFA no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte [Práticas recomendadas de segurança no IAM](#) no Guia do usuário do IAM.

## Usando o console AWS de IoT FleetWise

Para acessar o FleetWise console de AWS IoT, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os FleetWise recursos de AWS IoT em seu. Conta da AWS Se você criar uma política baseada em identidade que seja mais restritiva do que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis) com essa política.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente a ações que correspondam a operação de API que estiverem tentando executar.

Para garantir que usuários e funções ainda possam usar o FleetWise console de AWS IoT, anexe também a AWS IoT FleetWise ConsoleAccess ou a política ReadOnly AWS gerenciada às entidades. Para obter mais informações, consulte [Adicionando Permissões a um Usuário](#) no Guia do Usuário do IAM.

## Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "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": "*"
    }
  ]
}
```

## Acessar recursos no Amazon Timestream

Antes de usar a AWS IoT FleetWise, você deve registrar sua AWS conta, o IAM e os recursos do Amazon Timestream para AWS conceder permissão à FleetWise IoT para enviar dados do veículo em seu nome. Nuvem AWS Veja os requisitos de registro:

- Um banco de dados do Amazon Timestream.
- Uma tabela criada no banco de dados do Amazon Timestream especificado.
- Uma função do IAM que permite que a AWS IoT envie dados FleetWise para o Amazon Timestream.

Para obter mais informações, incluindo procedimentos e exemplos de políticas, consulte [Definir configurações](#).

## Solução de problemas AWS de FleetWise identidade e acesso à IoT

Use as informações a seguir para ajudá-lo a diagnosticar e corrigir problemas comuns que você pode encontrar ao trabalhar com AWS FleetWise IoT e IAM.

### Tópicos

- [Não estou autorizado a realizar uma ação na AWS IoT FleetWise](#)
- [Não estou autorizado a realizar iam: PassRole](#)
- [Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de AWS IoT FleetWise](#)

## Não estou autorizado a realizar uma ação na AWS IoT FleetWise

Se isso AWS Management Console indicar que você não está autorizado a realizar uma ação, entre em contato com o administrador para obter ajuda. Caso seu administrador seja a pessoa que forneceu suas credenciais de início de sessão.

O erro do exemplo a seguir ocorre quando o usuário do IAM `mateojackson` tenta usar o console para ver detalhes sobre um recurso fictício do `myVehicle`, mas não tem as permissões do `iotfleetwise:GetVehicleStatus`.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotfleetwise:GetVehicleStatus on resource: myVehicle
```

Neste caso, Mateo pede ao administrador para atualizar suas políticas para permitir a ele o acesso ao recurso `myVehicle` usando a ação `iotfleetwise:GetVehicleStatus`.

## Não estou autorizado a realizar iam: PassRole

Se você receber um erro informando que não está autorizado a realizar a `iam:PassRole` ação, suas políticas devem ser atualizadas para permitir que você passe uma função para a AWS IoT FleetWise.

Alguns Serviços da AWS permitem que você passe uma função existente para esse serviço em vez de criar uma nova função de serviço ou uma função vinculada ao serviço. Para fazer isso, é preciso ter permissões para passar o perfil para o serviço.

O exemplo de erro a seguir ocorre quando um usuário do IAM chamado `marymajor` tenta usar o console para realizar uma ação na AWS IoT FleetWise. No entanto, a ação exige que o serviço tenha permissões concedidas por um perfil de serviço. Mary não tem permissões para passar o perfil para o serviço.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Nesse caso, as políticas de Mary devem ser atualizadas para permitir que ela realize a ação `iam:PassRole`.

Se precisar de ajuda, entre em contato com seu AWS administrador. Seu administrador é a pessoa que forneceu suas credenciais de login.

## Quero permitir que pessoas de fora da minha Conta da AWS acessem meus recursos de AWS IoT FleetWise

Você pode criar uma função que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. Você pode especificar quem é confiável para assumir o perfil. Para serviços compatíveis com políticas baseadas em recursos ou listas de controle de acesso (ACLs), você pode usar essas políticas para conceder às pessoas acesso aos seus recursos.

Para saber mais, consulte:

- Para saber se a AWS IoT é FleetWise compatível com esses recursos, consulte [Como a AWS IoT FleetWise funciona com o IAM](#)
- Para saber como fornecer acesso aos seus recursos em todos os Contas da AWS que você possui, consulte Como [fornecer acesso a um usuário do IAM em outro Conta da AWS que você possui](#) no Guia do usuário do IAM.
- Para saber como fornecer acesso aos seus recursos a terceiros Contas da AWS, consulte Como [fornecer acesso Contas da AWS a terceiros](#) no Guia do usuário do IAM.
- Para saber como conceder acesso por meio da federação de identidades, consulte [Conceder acesso a usuários autenticados externamente \(federação de identidades\)](#) no Guia do usuário do IAM.
- Para saber a diferença entre usar perfis e políticas baseadas em recursos para acesso entre contas, consulte [Como os perfis do IAM diferem de políticas baseadas em recursos](#) no Guia do usuário do IAM.

## Validação de conformidade para AWS IoT FleetWise

### Note

AWS A IoT FleetWise não está no escopo de nenhum programa de AWS conformidade.


Para saber se um AWS service (Serviço da AWS) está dentro do escopo de programas de conformidade específicos, consulte [Serviços da AWS Escopo por Programa de Conformidade](#) [Serviços da AWS](#) e escolha o programa de conformidade em que você está interessado. Para obter informações gerais, consulte Programas de [AWS conformidade Programas AWS](#) de .



Você pode baixar relatórios de auditoria de terceiros usando AWS Artifact. Para obter mais informações, consulte [Baixar relatórios em AWS Artifact](#).

Sua responsabilidade de conformidade ao usar Serviços da AWS é determinada pela confidencialidade de seus dados, pelos objetivos de conformidade de sua empresa e pelas leis e regulamentações aplicáveis. AWS fornece os seguintes recursos para ajudar na conformidade:

- [Guias de início rápido sobre segurança e conformidade](#) — Esses guias de implantação discutem considerações arquitetônicas e fornecem etapas para a implantação de ambientes básicos AWS focados em segurança e conformidade.
- [Arquitetura para segurança e conformidade com a HIPAA na Amazon Web Services](#) — Este whitepaper descreve como as empresas podem usar AWS para criar aplicativos qualificados para a HIPAA.

 Note

Nem todos Serviços da AWS são elegíveis para a HIPAA. Para obter mais informações, consulte a [Referência dos serviços qualificados pela HIPAA](#).

- AWS Recursos de <https://aws.amazon.com/compliance/resources/> de conformidade — Essa coleção de pastas de trabalho e guias pode ser aplicada ao seu setor e local.
- [AWS Guias de conformidade do cliente](#) — Entenda o modelo de responsabilidade compartilhada sob a ótica da conformidade. Os guias resumem as melhores práticas de proteção Serviços da AWS e mapeiam as diretrizes para controles de segurança em várias estruturas (incluindo o Instituto Nacional de Padrões e Tecnologia (NIST), o Conselho de Padrões de Segurança do Setor de Cartões de Pagamento (PCI) e a Organização Internacional de Padronização (ISO)).
- [Avaliação de recursos com regras](#) no Guia do AWS Config desenvolvedor — O AWS Config serviço avalia o quão bem suas configurações de recursos estão em conformidade com as práticas internas, as diretrizes e os regulamentos do setor.
- [AWS Security Hub](#) — Isso AWS service (Serviço da AWS) fornece uma visão abrangente do seu estado de segurança interno AWS. O Security Hub usa controles de segurança para avaliar os atributos da AWS e verificar a conformidade com os padrões e as práticas recomendadas do setor de segurança. Para obter uma lista dos serviços e controles aceitos, consulte a [Referência de controles do Security Hub](#).
- [Amazon GuardDuty](#) — Isso AWS service (Serviço da AWS) detecta possíveis ameaças às suas cargas de trabalho Contas da AWS, contêineres e dados monitorando seu ambiente em busca de atividades suspeitas e maliciosas. GuardDuty pode ajudá-lo a atender a vários requisitos de

conformidade, como o PCI DSS, atendendo aos requisitos de detecção de intrusões exigidos por determinadas estruturas de conformidade.

- [AWS Audit Manager](#)— Isso AWS service (Serviço da AWS) ajuda você a auditar continuamente seu AWS uso para simplificar a forma como você gerencia o risco e a conformidade com as regulamentações e os padrões do setor.

## Resiliência na IoT AWS FleetWise

A infraestrutura AWS global é construída em torno de AWS regiões e zonas de disponibilidade. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte [Infraestrutura AWS global](#).

### Note

Os dados processados pela AWS IoT são FleetWise armazenados em um banco de dados Amazon Timestream. O Timestream oferece suporte a backups em outras zonas ou regiões de AWS disponibilidade. No entanto, você pode escrever seu próprio aplicativo usando o Timestream SDK para consultar dados e salvá-los no destino de sua escolha.

Para obter mais informações sobre o Amazon Timestream, consulte o [Guia do desenvolvedor do Amazon Timestream](#).

## Segurança da infraestrutura na AWS IoT FleetWise

Como um serviço gerenciado, a AWS IoT FleetWise é protegida pela segurança de rede AWS global. Para obter informações sobre serviços AWS de segurança e como AWS proteger a infraestrutura, consulte [AWS Cloud Security](#). Para projetar seu AWS ambiente usando as melhores práticas de segurança de infraestrutura, consulte [Proteção](#) de infraestrutura no Security Pillar AWS Well-Architected Framework.

Você usa chamadas de API AWS publicadas para acessar a AWS IoT FleetWise pela rede. Os clientes devem ser compatíveis com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com sigilo de encaminhamento perfeito (perfect forward secrecy, ou PFS) como DHE (Ephemeral Diffie-Hellman, ou Efêmero Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman, ou Curva elíptica efêmera Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, são compatíveis com esses modos.

Além disso, as solicitações devem ser assinadas utilizando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o [AWS Security Token Service](#) (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Você pode chamar essas operações de API de qualquer local de rede, mas a AWS IoT FleetWise oferece suporte a políticas de acesso baseadas em recursos, que podem incluir restrições com base no endereço IP de origem. Você também pode usar FleetWise políticas de AWS IoT para controlar o acesso de endpoints específicos da Amazon Virtual Private Cloud (Amazon VPC) ou VPCs específicas. Efetivamente, isso isola o acesso à rede a um determinado recurso de AWS FleetWise IoT somente da VPC específica dentro da rede. AWS

## Tópicos

- [Conectando-se à AWS IoT FleetWise por meio de uma interface VPC endpoint](#)

## Conectando-se à AWS IoT FleetWise por meio de uma interface VPC endpoint

Você pode se conectar diretamente à AWS IoT FleetWise usando uma interface [VPC endpoint \(AWS PrivateLink\) na sua Virtual Private Cloud \(VPC\)](#), em vez de se conectar pela Internet. Quando você usa uma interface VPC endpoint, a comunicação entre sua VPC e a AWS FleetWise IoT é conduzida inteiramente dentro da rede. AWS Cada endpoint da VPC é representado por uma ou mais [interfaces de rede elástica](#) (ENIs) com endereços IP privados nas sub-redes da VPC.

A interface VPC endpoint conecta sua VPC diretamente à AWS FleetWise IoT sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão. AWS Direct Connect As instâncias na sua VPC não precisam de endereços IP públicos para se comunicarem com a API de IoT AWS . FleetWise

Para usar a AWS IoT FleetWise por meio de sua VPC, você deve se conectar a partir de uma instância que esteja dentro da VPC ou conectar sua rede privada à sua VPC usando uma (VPN) ou. AWS Virtual Private Network AWS Direct Connect Para obter informações sobre o Amazon VPN, consulte [Conexões VPN](#) no Guia do usuário do Amazon Virtual Private Cloud. Para obter informações sobre AWS Direct Connect, consulte [Criação de uma conexão](#) no Guia AWS Direct Connect do usuário.

Você pode criar uma interface VPC endpoint para se conectar à AWS IoT FleetWise usando o AWS console ou os comandos (). AWS Command Line Interface AWS CLI Para obter mais informações, consulte [Creating an interface endpoint](#) (Criação de um endpoint de interface).

Depois de criar uma interface VPC endpoint, se você habilitar nomes de host DNS privados para o endpoint, o endpoint de AWS IoT FleetWise padrão será resolvido para seu endpoint VPC. O endpoint de nome de serviço padrão para AWS FleetWise IoT está no formato a seguir.

```
iotfleetwise.Region.amazonaws.com
```

Se você não ativar nomes de host DNS privados, o Amazon VPC fornecerá um nome de endpoint do DNS que você pode usar no formato a seguir.

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

Para obter mais informações, consulte [Interface VPC endpoints \(AWS PrivateLink\)](#) no Guia do usuário da Amazon VPC.

AWS A IoT FleetWise oferece suporte para fazer chamadas para todas as [ações de API](#) dentro da sua VPC.

Você pode anexar políticas de endpoint da VPC a um endpoint da VPC para controlar o acesso de entidades principais do IAM. Também é possível associar grupos de segurança a um VPC endpoint para controlar o acesso de entrada e saída com base na origem e no destino do tráfego de rede, como um intervalo de endereços IP. Para obter mais informações, consulte [Controlling access to services with VPC endpoints](#).

## Criação de uma política de VPC endpoint para IoT AWS FleetWise

Você pode criar uma política para endpoints Amazon VPC para AWS IoT para especificar o seguinte FleetWise :

- O principal que pode ou não executar ações
- As ações que podem ou não ser executadas

Para mais informações, consulte [Controlar o acesso a serviços com VPC endpoints](#) no Guia do usuário da Amazon VPC.

Example — Política de VPC endpoint para negar todo o acesso de uma conta especificada AWS

A política de VPC endpoint a seguir nega à AWS conta **123456789012** todas as chamadas de API usando o endpoint.

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Example – Política de endpoint da VPC para permitir o acesso à VPC somente a uma entidade principal do IAM (usuário) especificada

*A política de VPC endpoint a seguir permite acesso total somente ao usuário **lijuan na conta 123456789012. AWS** Todos as outras entidades principais do IAM têm acesso negado usando o endpoint.*

```
{
  "Statement": [
```

```
{
  "Action": "*",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": [
      "arn:aws:iam::123456789012:user/Lijuan"
    ]
  }
}]
}
```

### Example — Política de endpoint de VPC para ações de IoT AWS FleetWise

Veja a seguir um exemplo de uma política de endpoint para AWS IoT FleetWise. *Quando anexada a um endpoint, essa política concede acesso às ações de AWS FleetWise IoT listadas para o usuário do IAM FleetWise no 123456789012. Conta da AWS*

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/fleetWise"
        ]
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:ListFleets",
        "iotfleetwise:ListCampaigns",
        "iotfleetwise:CreateVehicle",
      ]
    }
  ]
}
```

## Análise de configuração e vulnerabilidade na AWS IoT FleetWise

Ambientes de IoT consistem em grandes quantidades de dispositivos com capacidades diversas, duradouros e geograficamente distribuídos. Essas características tornam a configuração do dispositivo complexa e propensa a erros. Além disso, os dispositivos quase sempre têm restrições

quanto à capacidade computacional, memória e capacidade de armazenamento, o uso de criptografia e outras formas de segurança nos dispositivos é limitado. Os dispositivos geralmente usam software com vulnerabilidades conhecidas. Esses fatores tornam os dispositivos de IoT, incluindo veículos que coletam dados AWS para IoT FleetWise, um alvo atraente para hackers e dificultam sua proteção contínua.

A configuração e os controles de TI são uma responsabilidade compartilhada entre você AWS e você, nosso cliente. Para obter mais informações, consulte o [modelo de responsabilidade AWS compartilhada](#).

## Melhores práticas de segurança para AWS IoT FleetWise

AWS A IoT FleetWise fornece vários recursos de segurança a serem considerados ao desenvolver e implementar suas próprias políticas de segurança. As melhores práticas a seguir são diretrizes gerais e não representam uma solução completa de segurança. Como essas melhores práticas podem não ser adequadas ou suficientes para o seu ambiente, trate-as como considerações úteis em vez de prescrições.

Para saber mais sobre segurança em, AWS IoT consulte [Práticas recomendadas de segurança AWS IoT Core no](#) Guia do AWS IoT desenvolvedor.

### Conceder o mínimo possível de permissões

Siga o princípio de privilégio mínimo usando o conjunto mínimo de permissões em funções do IAM. Limite o uso do caractere curinga \* para as propriedades Action e Resource em suas políticas do IAM. Em vez disso, declare um conjunto finito de ações e recursos quando possível. Para obter mais informações sobre as melhores práticas de privilégio mínimo e outras de políticas, consulte [the section called “Melhores práticas de política”](#).

### Não registrar em log informações confidenciais

Você deve impedir o registro de credenciais e outras informações de identificação pessoal (PII). Recomendamos que você implemente as seguintes proteções:

- Não use informações confidenciais nos nomes dos dispositivos.
- Não use informações confidenciais nos nomes e IDs dos FleetWise recursos de AWS IoT, por exemplo, nos nomes de campanhas, manifestos de decodificadores, modelos de veículos e catálogos de sinais, ou nos IDs de veículos e frotas.

## Use AWS CloudTrail para visualizar o histórico de chamadas da API

Você pode ver um histórico das chamadas de FleetWise API de AWS IoT feitas em sua conta para fins de análise de segurança e solução de problemas operacionais. Para receber um histórico das chamadas da FleetWise API de AWS IoT feitas em sua conta, basta ativar CloudTrail o. AWS Management Console Para ter mais informações, consulte [the section called “Logs do CloudTrail”](#).

## Manter o relógio do dispositivo sincronizado

É importante ter a hora exata no seu dispositivo. Os certificados X.509 têm data e hora de expiração. O relógio em seu dispositivo é usado para verificar se um certificado de servidor ainda é válido. Os relógios do dispositivo podem atrasar ao longo do tempo ou as baterias podem descarregar.

Para obter mais informações, consulte a melhor prática [Manter o relógio do dispositivo sincronizado](#) no Guia do desenvolvedor do AWS IoT Core .



# Monitoramento do AWS IoT FleetWise

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e a performance do AWS IoT FleetWise e das outras soluções da AWS. A AWS fornece as seguintes ferramentas de monitoramento para observar o AWS IoT FleetWise, informar quando algo está errado e tomar ações automáticas quando apropriado:

- O Amazon CloudWatch monitora os recursos da AWS e as aplicações que você executa na AWS em tempo real. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que notificam você ou que realizam ações quando uma métrica atinge um limite especificado. Por exemplo, você pode fazer o CloudWatch acompanhar o uso da CPU ou outras métricas das instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).
- O Amazon CloudWatch Logs permite monitorar, armazenar e acessar os arquivos de log de instâncias do Amazon EC2, do CloudTrail e de outras fontes. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. Você também pode arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).
- AWS CloudTrail captura chamadas de API e eventos relacionados realizados pela conta da Conta da AWS ou em nome dela. Em seguida, ele disponibiliza os arquivos de log a um bucket do Amazon S3 especificado por você. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem no qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o [AWS CloudTrail Guia do Usuário](#).

## Monitoramento do AWS IoT FleetWise com Amazon CloudWatch

As métricas do Amazon CloudWatch são uma forma de monitorar os recursos da AWS e a performance. AWS IoT FleetWise envia métricas para o CloudWatch. Você pode usar o AWS Management Console, a AWS CLI ou uma API para listar as métricas que o AWS IoT FleetWise envia para o CloudWatch. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch](#).

**⚠ Important**

Você deve definir as configurações para que o AWS IoT FleetWise possa enviar métricas para o CloudWatch. Para obter mais informações, consulte [Configuração de definições](#).

O namespace `AWS/IoTFleetWise` inclui as métricas a seguir.

## Métricas de sinal

| Métrica                             | Descrição                                                                                                                                                                                                                                                                             |
|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>IllegalMessageFromEdge</code> | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise não correspondia ao formato exigido.</p> <p>Unidades: contagem</p> <p>Dimensões: <code>VehicleName</code></p> <p>Estatística válida: soma</p>                                                                    |
| <code>MessageThrottled</code>       | <p>Uma mensagem enviada do veículo para o AWS IoT FleetWise foi limitada. Isso ocorre porque você excedeu os <a href="#">limites de serviço</a> dessa conta na região atual.</p> <p>Unidades: contagem</p> <p>Dimensões: <code>VehicleName</code></p> <p>Estatística válida: soma</p> |
| <code>ModelingError</code>          | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo.</p> <p>Unidades: contagem</p> <p>Dimensões: <code>ModelManifestName</code></p>                                                              |

| Métrica       | Descrição                                                                                                                                                                                                                                                                  |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | Estatística válida: soma                                                                                                                                                                                                                                                   |
| DecodingError | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise contém sinais que apresentam falha na decodificação com relação ao manifesto de decodificador do veículo.</p> <p>Unidades: contagem</p> <p>Dimensões: DecoderName</p> <p>Estatística válida: soma</p> |

### Métricas de campanha

| Métrica         | Descrição                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VehicleNotFound | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que o veículo é desconhecido.</p> <p>Unidades: contagem</p> <p>Dimensões: VehicleName</p> <p>Estatística válida: soma</p> |
| CampaignInvalid | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que a campanha não é válida.</p> <p>Unidades: contagem</p> <p>Dimensões: CampaignName</p> <p>Estatística válida: soma</p> |

| Métrica          | Descrição                                                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CampaignNotFound | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que a campanha é desconhecida.</p> <p>Unidades: contagem</p> <p>Dimensões: CampaignName</p> <p>Estatística válida: soma</p> |

### Métricas de destino dos dados da campanha

| Métrica              | Descrição                                                                                                                                                                                                              |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TimestreamWriteError | <p>O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo na tabela do Amazon Timestream.</p> <p>Unidades: contagem</p> <p>Dimensões: DatabaseName, TableName</p> <p>Estatística válida: soma</p>            |
| S3WriteError         | <p>O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p> <p>Unidades: contagem</p> <p>Dimensões: BucketName</p> <p>Estatística válida: soma</p> |
| S3ReadError          | <p>O AWS IoT FleetWise não conseguiu ler a chave de um objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3).</p>                                                                                   |

| Métrica | Descrição                |
|---------|--------------------------|
|         | Unidades: contagem       |
|         | Dimensões: BucketName    |
|         | Estatística válida: soma |

### Principais métricas do AWS KMS gerenciadas pelo cliente

| Métrica            | Descrição                                                                                                                                                                                                                                                         |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMSKeyAccessDenied | <p>O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo na tabela Timestream ou no bucket do Amazon S3 devido a um erro de acesso negado à chave do AWS KMS.</p> <p>Unidades: contagem</p> <p>Dimensões: KMSKeyId</p> <p>Estatística válida: soma</p> |

## Monitoramento do AWS IoT FleetWise com Amazon CloudWatch Logs

O Amazon CloudWatch Logs monitora os eventos que ocorrem nos recursos e alerta quando há algum problema. Se receber um alerta, você poderá acessar os arquivos de log para obter informações sobre o evento específico. Para obter mais informações, consulte o [Guia do usuário do Amazon CloudWatch Logs](#).

### Visualizar logs do AWS IoT FleetWise no console do CloudWatch

#### Important

Antes de ver o grupo de logs do AWS IoT FleetWise no console do CloudWatch, verifique se o seguinte é verdadeiro:

- Você habilitou o registro em log no AWS IoT FleetWise. Para obter mais informações sobre registro em log, consulte [Configurar o registro em log do AWS IoT FleetWise](#).
- Já existem entradas de log escritas por operações da AWS IoT.

Para visualizar logs do AWS IoT FleetWise no console do CloudWatch

1. Abra o [console do CloudWatch](#).
2. No painel de navegação, escolha Logs > Grupos de logs.
3. Escolha o grupo de logs .
4. Escolha Pesquisar grupo de logs. Você verá uma lista completa dos eventos de log gerados para sua conta.
5. Escolha o ícone de expansão para examinar um fluxo individual e encontrar todos os logs que tem um nível de log ERROR.

Você também pode inserir uma consulta na caixa de pesquisa Filtrar eventos. Por exemplo, você pode tentar a seguinte consulta:

```
{ $.logLevel = "ERROR" }
```

Para obter mais informações sobre como criar expressões de filtro, consulte [Sintaxe de padrões de filtros](#) no Guia do usuário do Amazon CloudWatch Logs.

Example entrada de registro

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

## Tipos de eventos de sinal

| Tipo de evento            | Descrição                                                                                                                                                                                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MODELING_ERROR            | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo.</p> <p>Atributos: vehicleName, campaignName, signalCatalogName, signalId, signalValue, signalValueRangeMin, signalValueRangeMax, modelManifestName</p>   |
| ILLEGAL_MESSAGE_FROM_EDGE | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise não correspondia ao formato exigido.</p> <p>Atributos: vehicleName, campaignName, signalCatalogName</p>                                                                                                                       |
| DECODING_ERROR            | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise contém sinais que apresentam falha na decodificação com relação ao manifesto de decodificador do veículo.</p> <p>Atributos: campaignName, signalCatalogName, decoderManifestName, (opcional) signalName, (opcional) s3URI</p> |

## Tipos de evento de campanha

| Tipo de evento    | Descrição                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------|
| VEHICLE_NOT_FOUND | <p>Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que o veículo era desconhecido.</p> |

| Tipo de evento     | Descrição                                                                                                                                                     |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | Atributos: vehicleName, campaignName                                                                                                                          |
| CAMPAIGN_NOT_FOUND | Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que a campanha era desconhecida.<br><br>Atributos: vehicleName (opcional), campaignName |
| CAMPAIGN_INVALID   | Uma mensagem enviada do veículo e recebida pelo AWS IoT FleetWise, em que a campanha não era válida.<br><br>Atributos: vehicleName (opcional), campaignName   |

#### Tipos de eventos de destino de dados da campanha

| Tipo de evento         | Descrição                                                                                                                                                                                 |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TIMESTREAM_WRITE_ERROR | O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo na tabela do Amazon Timestream.<br><br>Atributos: vehicleName, campaignName, timestreamDatabaseName, timestreamTableName |
| S3_WRITE_ERROR         | O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3).<br><br>Atributos: campaignName, destinationName                  |
| S3_READ_ERROR          | O AWS IoT FleetWise não conseguiu ler a chave de um objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3).                                                             |



| Tipo de evento | Descrição                                |
|----------------|------------------------------------------|
|                | Atributos: campaignName, destinationName |

Principais tipos de eventos do AWS KMS gerenciados pelo cliente

| Tipo de evento        | Descrição                                                                                                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| KMS_KEY_ACCESS_DENIED | O AWS IoT FleetWise não conseguiu gravar uma mensagem do veículo na tabela Timestream ou no bucket do Amazon S3 devido a um erro de acesso negado à chave do AWS KMS. |

## Atributos

Todas as entradas do CloudWatch Logs incluem estes atributos:

`accountId`

Seu ID da Conta da AWS.

`eventType`

O tipo de evento para o qual o log foi gerado. O valor do tipo de evento depende do evento que gerou a entrada de log. Cada descrição de entrada de log inclui o valor de `eventType` para essa entrada de log.

`logLevel`

O nível de log que está sendo usado. Para obter mais informações, consulte [Níveis de log](#) no Guia do desenvolvedor do AWS IoT Core.

`message`

Contém detalhes específicos sobre o log.

`timestamp`

A marcação de data/hora em milissegundos de quando o AWS IoT FleetWise processou o log.

## Atributos opcionais

As entradas do CloudWatch Logs incluem opcionalmente esses atributos, dependendo do `eventType`:

### `decoderManifestName`

O nome do manifesto do decodificador que contém o sinal.

### `destinationName`

O nome do destino dos dados do veículo. Por exemplo, o nome do bucket do Amazon S3.

### `campaignName`

O nome da campanha.

### `signalCatalogName`

O nome do catálogo de sinais que contém o sinal.

### `signalId`

O ID do sinal de erro.

### `signalIds`

Uma lista de IDs de sinal de erro.

### `signalName`

O nome do sinal.

### `signalTimestampEpochMs`

A marcação de data/hora do sinal de erro.

### `signalValue`

O valor do sinal de erro.

### `signalValueRangeMax`

O alcance máximo do sinal de erro.

### `signalValueRangeMin`

O alcance mínimo do sinal de erro.

## s3URI

O identificador exclusivo do Amazon S3 de um arquivo do Amazon Ion de uma mensagem do veículo.

## timestreamDatabaseName

O nome do banco de dados do Timestream.

## timestreamTableName

O nome da tabela do Timestream.

## vehicleName

O nome do veículo.

## Configurar o registro em log do AWS IoT FleetWise

Você pode enviar dados de log do AWS IoT FleetWise para um grupo de logs do CloudWatch. O CloudWatch Logs oferece visibilidade caso o AWS IoT FleetWise falhe no processamento de mensagens de veículos. Por exemplo, isso pode acontecer devido a uma configuração incorreta ou a outros erros do cliente. Você será notificado sobre quaisquer erros para poder identificar e mitigar problemas.

Antes de enviar logs para o CloudWatch, você deve criar um grupo de logs do CloudWatch. Configure o grupo de logs com a mesma conta e na mesma região que você usou com o AWS IoT FleetWise. Ao ativar o registro em log no AWS IoT FleetWise, forneça o nome do grupo de logs. Depois que o registro em log é ativado, o AWS IoT FleetWise envia os registros para o grupo de logs do CloudWatch em fluxos de log.

Você pode visualizar os dados de log enviados do AWS IoT FleetWise no console do CloudWatch. Para obter mais informações sobre como configurar um grupo de log do CloudWatch e visualizar dados de log, consulte [Trabalhar com grupos de log](#).

## Permissões para publicar logs no CloudWatch

Configurar o registro em log de um grupo de logs do CloudWatch requer as configurações de permissões descritas nesta seção. Para obter informações sobre como gerenciar permissões, consulte [Gerenciamento de acesso a recursos da AWS](#) no Guia do usuário do IAM.

Com essas permissões, você pode alterar a configuração de registro em log, configurar a entrega de log para o CloudWatch e recuperar informações sobre seu grupo de logs.

```
{
  "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"
    }
  ]
}
```

Quando as ações são permitidas em todos os recursos da AWS, é indicado na política com uma configuração "Resource" de "\*". Isso significa que as ações são permitidas em todos os recursos da AWS compatíveis com cada ação.

## Configurar o registro em log no AWS IoT FleetWise (console)

Esta seção descreve como usar o console do AWS IoT FleetWise para configurar o registro em log.

Para usar o console do AWS IoT FleetWise para configurar o registro em log

1. Abra o [console do AWS IoT FleetWise](#).
2. No painel esquerdo, escolha Settings (Configurações).
3. Na seção Registro em log da página Configurações, selecione Editar.
4. Na seção Registro em log do CloudWatch, insira o Grupo de logs.
5. Selecione Enviar para salvar as alterações.

Depois de habilitar o registro em log, você poderá visualizar os dados de log no console do [CloudWatch](#).

## Configurar o registro em log padrão no AWS IoT FleetWise (CLI)

Esta seção descreve como configurar o registro em log para o AWS IoT FleetWise usando a CLI.

Também é possível executar esse procedimento com a API usando os métodos na API da AWS que correspondam aos comandos da CLI mostrados aqui. Você pode usar a operação da API [GetLoggingOptions](#) para buscar a configuração atual e a operação da API [PutLoggingOptions](#) para modificar a configuração.

Para usar a CLI para configurar o log para o AWS IoT FleetWise

1. Para obter as opções de registro em log da sua conta, use o comando get-logging-options.

```
aws iotfleetwise get-logging-options
```

2. Para ativar o registro em log, use o comando put-logging-options.

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery  
logType=ERROR,logGroupName=MyLogGroup
```

onde:

logType

O tipo de log para enviar dados ao CloudWatch Logs. Para desativar o registro em log, altere o valor para OFF.

## logGroupName

O grupo do CloudWatch Logs para o qual a operação envia dados. Crie o nome do grupo de logs antes de ativar o registro em log para o AWS IoT FleetWise.

Depois de ativar o registro em log, consulte [Pesquisar entradas de log usando a AWS CLI](#).

## Registrar chamadas da API do AWS IoT FleetWise usando AWS CloudTrail

O AWS IoT FleetWise está integrado ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou serviço da AWS no AWS IoT FleetWise. O CloudTrail captura as chamadas de API do AWS IoT FleetWise como eventos. As chamadas capturadas incluem chamadas do console do AWS IoT FleetWise e chamadas de código para as operações da API do AWS IoT FleetWise. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos do AWS IoT FleetWise. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Ao usar as informações coletadas pelo CloudTrail, você pode determinar a solicitação feita ao AWS IoT FleetWise, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o [Guia do usuário do AWS CloudTrail](#).

## Informações sobre o AWS IoT FleetWise no CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre atividade no AWS IoT FleetWise, essa atividade é registrada em um evento do CloudTrail junto com outros eventos de serviço da AWS em Histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte [Como visualizar eventos com o histórico de eventos do CloudTrail](#).

Para um registro contínuo de eventos em sua conta da AWS, incluindo eventos do AWS IoT FleetWise, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível

configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- [Visão geral da criação de uma trilha](#)
- [Serviços e integrações compatíveis com o CloudTrail](#)
- [Configurar notificações do Amazon SNS para o CloudTrail](#)
- [Recebimento de arquivos de log do CloudTrail de várias regiões](#)
- [Recebimento de arquivos de log do CloudTrail de várias contas](#)

Todas as ações do AWS IoT FleetWise são registradas pelo CloudTrail e documentadas na [Referência da API do AWS IoT FleetWise](#). Por exemplo, as chamadas para as APIs `CreateCampaign`, `AssociateVehicleFleet` e `GetModelManifest` geram entradas nos arquivos de log do CloudTrail.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do usuário do IAM.
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte [Elemento `userIdentity` do CloudTrail](#).

## Noções básicas sobre entradas do arquivo de log do AWS IoT FleetWise

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a operação *`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"
}
```



# Histórico de documentos do Guia do desenvolvedor do AWS IoT FleetWise

A tabela a seguir descreve as versões de documentação do AWS IoT FleetWise.

| Alteração                                                  | Descrição                                                                                                                                                                                                                                                                                                                                                                                      | Data                   |
|------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#">Prévia dos dados de sistemas de visão</a>      | É possível usar a prévia dos dados de sistemas de visão do AWS IoT FleetWise para coletar e organizar dados dos sistemas de visão do veículo, inclusive de câmeras, radares e lidars. Ele mantém os dados de sistemas de visão estruturados e não estruturados, os metadados (ID do evento, campanha, veículo) e o sensor padrão (dados de telemetria) sincronizados automaticamente na nuvem. | 26 de novembro de 2023 |
| <a href="#">Chaves gerenciadas pelo cliente do AWS KMS</a> | O AWS IoT FleetWise agora é compatível com as chaves gerenciadas pelo cliente do AWS KMS. É possível usar a chave do KMS para criptografar dados do lado do servidor relacionados aos recursos do AWS IoT FleetWise (catálogo de sinais, modelo do veículo, manifesto do decodificador, veículos e configurações da campanha de coleta de dados) armazenados na Nuvem AWS.                     | 16 de outubro de 2023  |

---

|                                                              |                                                                                                                                                                                                                         |                        |
|--------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| <a href="#"><u>Armazenamento de objetos no Amazon S3</u></a> | O AWS IoT FleetWise agora é compatível com o armazenamento de dados usando o Amazon Simple Storage Service (Amazon S3). É possível armazenar dados coletados durante campanhas no Amazon S3, além do Amazon Timestream. | 1.º de junho de 2023   |
| <a href="#"><u>Disponibilidade geral</u></a>                 | Esta é a versão pública do AWS IoT FleetWise.                                                                                                                                                                           | 27 de setembro de 2022 |
| <a href="#"><u>Lançamento inicial</u></a>                    | Esta é a versão prévia do Guia do desenvolvedor do AWS IoT FleetWise.                                                                                                                                                   | 30 de novembro de 2021 |

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.