Guia do desenvolvedor

AWS IoT FleetWise

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS IoT FleetWise: Guia do desenvolvedor

Copyright © 2026 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 desprestigie a Amazon. Todas as outras marcas comerciais que não pertencem à Amazon pertencem a seus respectivos proprietários, que podem ou não ser afiliados, patrocinados pela Amazon ou ter conexão com ela.

Table of Contents

O que é AWS IoT FleetWise? 1
Benefícios 2
Casos de uso 3
Aviso importante 3
Você é novato na AWS IoT FleetWise? 4
Acessando a AWS IoT FleetWise 4
Preços para AWS IoT FleetWise 4
Serviços relacionados 5
Principais conceitos 5
Principais conceitos 6
Características da AWS IoT FleetWise 11
AWS Regiões suportadas 11
Configurar a AWS IoT FleetWise 14
Configure seu Conta da AWS 14
Inscreva-se para um Conta da AWS 14
Criar um usuário com acesso administrativo 15
Comece pelo console 16
Defina suas configurações 17
Definir configurações (console) 17
Definir configurações (AWS CLI) 18
Usando IPv6 com a AWS IoT FleetWise 20
IPv6 pré-requisitos para pontos finais do plano de controle 20
IPv6 suporte para AWS PrivateLink endpoints 20
Testando a compatibilidade IPv6 de endereços 20
Usando IPv6 endereços nas políticas do IAM 21
Usar endpoints de pilha dupla 22
Conceitos básicos 24
Introdução 24
Pré-requisitos 25
Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise 26
Etapa 2: criar um modelo de veículo 27
Etapa 3: criar um manifesto do decodificador 29
Etapa 4: configurar um manifesto do decodificador 30
Etapa 5: criar um veículo 31

Etapa 6: criar uma campanha 32
Etapa 7: limpar 34
Próximas etapas 34
Ingerir dados 35
Veículos modelo 39
Catálogos de sinais 42
Configurar sinais 45
Crie um catálogo de sinais 51
Importar um catálogo de sinais 56
Atualizar um catálogo de sinais 66
Excluir um catálogo de sinais 70
Obtenha informações do catálogo de sinais 71
Modelos de veículos 72
Criar um modelo de veículo 73
Atualizar o modelo de um veículo 80
Excluir um modelo de veículo 82
Obtenha informações sobre o modelo do veículo 84
Manifestos de decodificadores 85
Configurar interfaces e sinais 88
Criar um manifesto do decodificador 91
Atualizar um manifesto do decodificador 101
Excluir um manifesto do decodificador 104
Obtenha informações do manifesto do decodificador 107
Gerenciar veículos 108
Provisionar veículos 109
Autenticar veículos 110
Autorizar veículos 112
Tópicos reservados 113
Criar um veículo 118
Criar um veículo (console) 118
Criar um veículo (AWS CLI) 120
Crie vários veículos 123
Atualizar um veículo 125
Atualize vários veículos 128
Excluir um veículo 129
Excluir um veículo (console) 129

Excluir um veículo (AWS CLI) 130
Obtenha informações sobre o veículo 131
Gerenciar frotas 133
Criar uma frota 134
Associar um veículo a uma frota 135
Desassociar um veículo de uma frota 136
Atualizar uma frota 137
Excluir uma frota 138
Verificar a exclusão da frota 139
Obtenha informações sobre a frota 139
Gerencie dados com campanhas 143
Criar uma campanha 149
Criar uma campanha (console) 150
Criar uma campanha (AWS CLI) 158
Expressões lógicas para campanhas AWS de IoT FleetWise 164
Atualizar uma campanha 165
Excluir uma campanha 167
Excluir uma campanha (console) 167
Excluir uma campanha (AWS CLI) 167
Verificar a exclusão da campanha 167
Obtenha informações sobre a campanha 168
Armazenar e encaminhar 169
Crie partições de dados 169
Carregar dados da campanha 173
Carregar dados usando AWS IoT Jobs 174
Colete dados do código de problemas de diagnóstico 175
Palavras-chave do código de problema de diagnóstico 177
Crie uma campanha de coleta de dados para códigos de problemas de diagnóstico 179
Casos de uso do código de problema de diagnóstico 182
Visualize dados do veículo 185
Processando dados do veículo enviados para um tópico do MQTT 186
Processar dados do veículo no Timestream 187
Visualize os dados do veículo armazenados no Timestream 188
Processar dados do veículo no Amazon S3 188
Formato de objeto Amazon S3 189
Analise os dados do veículo armazenados no Amazon S3 191

Comandos 193
Conceitos de comandos 194
Conceitos de chaves de comandos 195
Status de execução do comando 198
Veículos e comandos 204
Visão geral do fluxo de trabalho 204
fluxo de trabalho do veículo 206
Fluxo de trabalho de comandos 208
(Opcional) Notificações de comandos 210
Criar e gerenciar comandos 211
Criar um recurso de comando 212
Recuperar informações sobre um comando 214
Listar comandos na sua conta 215
Atualizar ou descontinuar um recurso de comando 216
Excluir um recurso de comando 217
Iniciar e monitorar execuções de comando 219
Atualizar o resultado da execução do comando 223
Obtenha a execução do comando 225
Listar as execuções de comandos em sua conta 226
Excluir uma execução de comando 228
Exemplo: uso de comandos 229
Visão geral do exemplo do modo de direção do veículo 229
Pré-requisitos 230
Política do IAM para usar comandos remotos 230
Executar AWS IoT comandos (AWS CLI) 233
Limpeza 238
Cenários de uso de comandos 240
Criando um comando sem parâmetros 240
Criação de um comando com valores padrão para parâmetros 242
Criação de um comando com valores de parâmetros 243
Usando comandos com modelos de estado 244
Último estado conhecido 247
Crie um modelo de estado 248
Associar um modelo de FleetWise estado de AWS IoT a um veículo 250
Atualizar um modelo de estado 251
Excluir um modelo de estado 252

Obtenha informações do modelo de estado ... 253
Operações de modelo estadual ... 254
Ativar e desativar a coleta de dados de estado ... 255
Obtenha um instantâneo do estado do veículo ... 260
Processe os últimos dados do veículo estadual conhecido usando mensagens MQTT ... 263
Configurar a coleta de dados independente da rede ... 267
Introdução ... 267
Configuração do ambiente ... 267
Modelos de dados ... 267
Atualizações do catálogo do Signal ... 268
Modelo e decodificador do veículo ... 270
Enviar comando ... 272
AWS CLI and SDKs ... 274
Solução de problemas ... 275
Problemas com o manifesto do decodificador ... 275
Problemas com o agente Edge ... 279
Problema: o software Edge Agent não inicia. ... 279
Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [Falha ao iniciar a biblioteca de persistência] ... 281
Problema: O software Edge Agent não coleta diagnósticos integrados (OBD) II PIDs e códigos de problemas de diagnóstico (). DTCs ... 281
Problema: o FleetWise software Edge Agent para AWS IoT não coleta dados da rede ou não consegue aplicar regras de inspeção de dados. ... 281
Problema: [ERROR] [AwsIotConnectivityModule: :connect]: [Falha na conexão com erro] ou [WARN] [AwsIotChannel: :send]: [Nenhuma conexão MQTT ativa.] ... 282
Armazenar e encaminhar problemas ... 283
Problema: receber um `AccessDeniedException` com todas as permissões necessárias do IAM ... 283
Problema: os dados enviados para AWS IoT Jobs ignoram o `endTime` ... 283
Problema: o upload de dados para AWS IoT Jobs tem um status de `REJECTED` execução. ... 283
Segurança ... 284
Proteção de dados ... 285
Criptografia em repouso na AWS IoT FleetWise ... 286
Criptografia em trânsito ... 286
Criptografia de dados na AWS IoT FleetWise ... 287

Controlar o acesso 299
Conceder AWS IoT FleetWise permissão para enviar e receber dados sobre um tópico do MQTT 300
Conceder AWS IoT FleetWise acesso a um destino do Amazon S3 303
Conceda AWS IoT FleetWise acesso a um destino do Amazon Timestream 306
Conceda AWS IoT Device Management permissão para gerar a carga útil para comandos com AWS IoT FleetWise 309
Gerenciamento de Identidade e Acesso 314
Público 315
Autenticação com identidades 315
Gerenciar o acesso usando políticas 317
Como a AWS IoT FleetWise funciona com o IAM 318
Exemplos de políticas baseadas em identidade 327
Solução de problemas 330
Referência de permissões da API 332
Atualizações da política gerenciada 342
AWSIoTFleetwiseServiceRolePolicy 343
Validação de conformidade 343
Resiliência 344
Segurança da infraestrutura 344
Conectando-se à AWS IoT FleetWise por meio de uma interface VPC endpoint 345
Análise de configuração e vulnerabilidade 348
Práticas recomendadas de segurança 349
Conceder o mínimo possível de permissões 349
Não registrar em log informações confidenciais 349
Use AWS CloudTrail para visualizar o histórico de chamadas da API 350
Manter o relógio do dispositivo sincronizado 350
Monitorando a AWS IoT FleetWise 351
Monitoramento com CloudWatch 351
Monitore com CloudWatch registros 358
Exibir FleetWise registros AWS de IoT no console CloudWatch 358
Configurar o registro em log da 365
CloudTrail troncos 368
AWS Informações de IoT em FleetWise CloudTrail 369
Compreensão das entradas do arquivo de log do 370
Histórico do documento 372

..... ccclxxvi

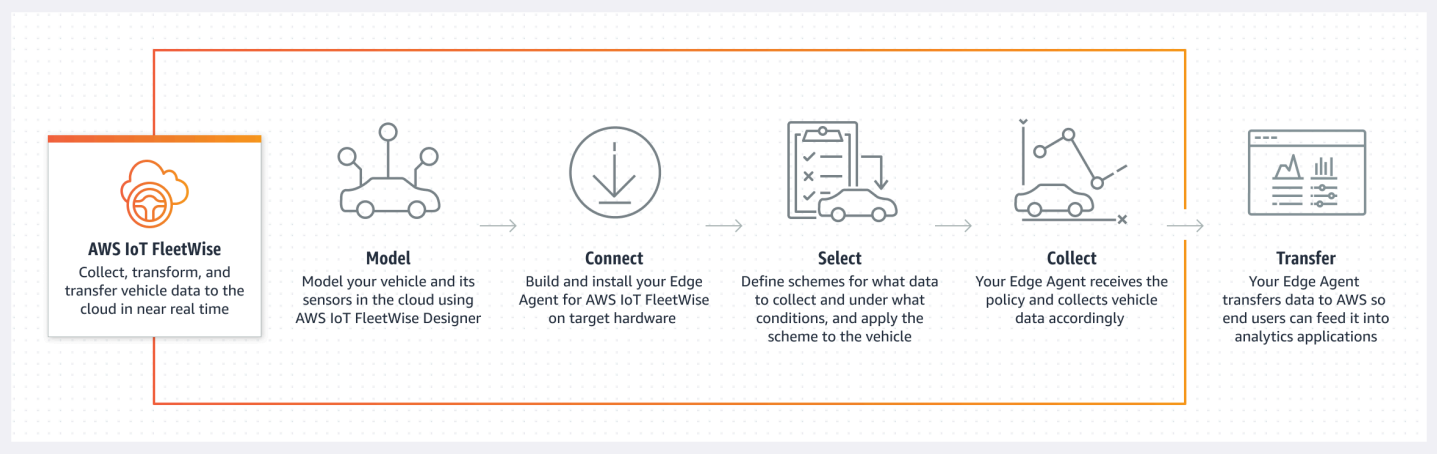
O que é AWS IoT FleetWise?

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

AWS A 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 a AWS IoT FleetWise, você pode coletar e organizar dados de veículos que usam protocolos e formatos de dados diferentes. AWS FleetWise A IoT ajuda a transformar mensagens de baixo nível em valores legíveis por humanos e a padronizar o formato de dados na nuvem para análises 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 infotenimento 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 da AWS IoT FleetWise.



Tópicos

- Benefícios
- Casos de uso
- Aviso importante
- Você é novato na AWS IoT FleetWise?
- Acessando a AWS IoT FleetWise
- Preços para AWS IoT FleetWise
- Serviços relacionados
- Principais conceitos e recursos da AWS IoT FleetWise
- AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise

Benefícios

Os principais benefícios da AWS IoT FleetWise são:

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

: Obtenha 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 automaticamente sincronizados na nuvem. AWS A IoT FleetWise mantém os dados estruturados e não estruturados do sistema de visão, metadados e dados de sensores padrão sincronizados automaticamente na nuvem. Isso simplifica o processo de montar uma visão geral dos eventos e receber informações.

Armazene dados no Edge e encaminhe-os em condições ideais

: Reduza os custos de transmissão armazenando temporariamente dados nos veículos. Você pode encaminhar dados selecionados para a nuvem sob condições ideais e especificadas, como quando os veículos se conectam ao Wi-Fi.

> **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 você pode usar a AWS IoT FleetWise incluem o seguinte:

AI/ML Modelos de trem

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.

Crie e gerencie comandos

Use comandos para executar comandos em um veículo a partir da nuvem. Você pode enviar comandos remotamente para um veículo e, em alguns segundos, o veículo executará o comando. Por exemplo, você pode configurar comandos para trancar a porta de um veículo ou definir a temperatura.

Crie e gerencie modelos de estado

Os modelos de estado fornecem um mecanismo para que os proprietários de veículos rastreiem o estado de seus veículos. O AWS IoT FleetWise Edge Agent executado no veículo coleta e envia atualizações de sinal para a nuvem.

Aviso importante

Os dados do veículo coletados por meio do uso da AWS IoT FleetWise são destinados apenas para fins informativos (inclusive para ajudá-lo a treinar modelos de inteligência artificial e aprendizado de

máquina baseados em nuvem), e você não pode usar a AWS FleetWise IoT para controlar ou operar as funções do veículo. Você é o único responsável por todas as responsabilidades que possam surgir em relação a qualquer uso fora da finalidade pretendida FleetWise da AWS IoT e de qualquer forma contrária às regulamentações de veículos aplicáveis.

Os dados do veículo coletados por meio do uso da AWS IoT FleetWise devem ser avaliados quanto à precisão, conforme apropriado para seu caso de uso, inclusive para fins de cumprimento de quaisquer obrigações de conformidade que você possa ter de acordo com os regulamentos de segurança veicular aplicáveis (como obrigações de monitoramento e emissão de relatórios de segurança). 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). Você e seus Usuários Finais são os únicos responsáveis por todas as decisões tomadas, conselhos dados, ações tomadas e falhas em tomar medidas com base no uso da AWS IoT FleetWise.

Você é novato na AWS IoT FleetWise?

Se você é iniciante na AWS IoT FleetWise, recomendamos que comece lendo as seguintes seções:

- Principais conceitos e recursos da AWS IoT FleetWise
- Configurar a AWS IoT FleetWise
- Tutorial: Comece a usar a AWS IoT FleetWise
- Ingira dados AWS de FleetWise IoT na nuvem

Acessando a AWS IoT FleetWise

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

Preços para 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 na AWS IoT FleetWise. Você também paga pelas mensagens coletadas dos veículos. Para obter informações atuais sobre preços, consulte a página de FleetWise preços de AWS IoT. Para saber mais sobre o protocolo de mensagens MQTT, consulte MQTT no Guia do desenvolvedor do AWS IoT Core .

Serviços relacionados

AWS A IoT se FleetWise integra aos seguintes AWS serviços para melhorar a disponibilidade e a escalabilidade de suas soluções em nuvem.

- AWS IoT Core— Registre e controle AWS IoT dispositivos que carregam dados do veículo para a AWS IoT FleetWise e enviam comandos remotamente para um veículo. 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.

Principais conceitos e recursos da AWS IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

Depois de ler essa introdução, consulte a Configurar a AWS IoT FleetWise seção para saber como configurar a AWS FleetWise IoT.

Tópicos

- Principais conceitos
- Características da AWS IoT FleetWise

Principais conceitos

AWS A IoT FleetWise fornece uma estrutura de modelagem de veículos para você modelar seu veículo e seus sensores e atuadores na nuvem. Para permitir a comunicação segura entre seu veículo e a nuvem, a AWS IoT FleetWise também fornece uma implementação de referência para ajudá-lo 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.

A seguir estão os principais conceitos da 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 Gerencie AWS catálogos de sinais de IoT FleetWise.

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. 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.

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 Gerencie AWS catálogos de sinais de IoT FleetWise.

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 Gerencie modelos AWS de veículos de IoT FleetWise.

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 obter mais informações, consulte Gerencie manifestos AWS de decodificadores de IoT FleetWise.

Interface de rede

Contém informações sobre o protocolo que a rede do veículo usa. AWS A 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). ECUs pode 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 eles. ECUs Ele fornece vários códigos de problemas de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado com seu 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.

Interfaces personalizadas

Você também pode usar sua própria interface para decodificar sinais no Edge. Isso pode economizar seu tempo, pois você não precisa criar regras de decodificação na nuvem.

Decodificador de sinal

Fornece informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um decodificador de sinal. 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 decodificadores de sinal OBD.

O manifesto do decodificador deve conter decodificadores de sinal de mensagem se também contiver interfaces de middleware do veículo. Ou, se o manifesto do decodificador contiver interfaces de decodificação personalizadas, ele também deverá conter sinais de decodificação personalizados.

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.

Campaign

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.

Partição de dados

Configure dados particionados em uma campanha para armazenar temporariamente os dados do sinal. Você configura quando e como encaminhar os 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, a AWS IoT FleetWise suporta 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.

Comandos

Os comandos executam comandos em um veículo a partir da nuvem. Você pode enviar comandos remotamente para um veículo e, em alguns segundos, o veículo executará o comando. Por exemplo, você pode configurar comandos para trancar a porta de um veículo ou definir a temperatura.

O comando é um recurso gerenciado pelo AWS IoT Device Management. Ele contém configurações reutilizáveis que são aplicadas ao enviar uma execução de comando para o veículo. Para obter mais informações, consulte AWS IoT os comandos no Guia do AWS IoT Core desenvolvedor.

Modelos de estado

Os modelos de estado fornecem um mecanismo para que os proprietários de veículos rastreiem o estado de seus veículos. O agente de software Edge Agent que é executado no veículo coleta e envia atualizações de sinal para a nuvem. Cada modelo de estado contém uma lista de sinais dos quais os dados são coletados.

Características da AWS IoT FleetWise

A seguir estão os principais recursos da AWS IoT FleetWise.

Modelagem de veículos

Crie representações virtuais de seus veículos e aplique um formato comum para organizar os sinais dos veículos. AWS A IoT é FleetWise compatível com a Especificação de Sinais de Veículos (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.

Edge Agent para software AWS de 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.

AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise

Para ver uma lista de AWS regiões que oferecem suporte à AWS IoT FleetWise, consulte endpoints e cotas de AWS IoT FleetWise . AWS Os FleetWise recursos de IoT diferem em seu suporte regional.

> **Note**
>
> O acesso à região Ásia-Pacífico (Mumbai) e alguns recursos de AWS FleetWise IoT estão atualmente bloqueados. Para solicitar acesso a essa AWS região e a todos os recursos fechados, entre em contato com seu gerente de conta ou com o AWS Support Center.

A tabela a seguir mostra o suporte a recursos por região:

Características/Regiões	Leste dos EUA (Norte da Virgínia)	Europa (Frankfurt)	Ásia-Pacífico (Mumbai) NOTA: somente acesso fechado
Catálogos de sinais	Sim	Sim	Fechado
Modelos de veículos	Sim	Sim	Fechado
Manifestos do decodificador	Sim	Sim	Fechado
Veículos	Sim	Sim	Fechado
Frotas	Sim	Sim	Fechado
Campanhas	Sim	Sim	Fechado
Dados do sistema de visão (na versão prévia)	Sim	Sim	Fechado
Tópico do MQTT como destino de dados da campanha	Fechado	Fechado	Fechado
Armazenar e encaminhar	Fechado	Fechado	Fechado
Comandos	Fechado	Fechado	Fechado
Último estado conhecido	Fechado	Fechado	Fechado
Coleta de dados independente de rede usando uma interface de decodificação personalizada	Fechado	Fechado	Fechado

Características/Regiões	Leste dos EUA (Norte da Virgínia)	Europa (Frankfurt)	Ásia-Pacífico (Mumbai) NOTA: somente acesso fechado
Obtenção do código de problema de diagnóstico (DTC) *	Fechado	Fechado	Fechado

*A busca do DTC oferece uma variedade de recursos que vão além da recuperação básica de dados do DTC. Essa funcionalidade inclui recursos personalizados que permitem definir funções na borda e invocá-las pelo nome em expressões de campanha baseadas em condições. Além disso, ele suporta a coleta de cadeias de caracteres ilimitadas, fornecendo manipulação flexível de tipos de dados de cadeias de caracteres. O Edge Agent pode buscar dados periodicamente ou acionado por condições específicas, aumentando sua adaptabilidade e eficiência nos processos de coleta de dados. Para obter mais informações, consulte o guia de funções personalizadas e a implementação da referência de coleta de dados do DTC no Guia do Desenvolvedor do Edge Agent.

Configurar 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
- Defina suas configurações AWS de IoT FleetWise
- Fazendo solicitações para a AWS IoT FleetWise usando IPv6

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 a https://portal.aws.amazon.com/billing/inscrição.
2. Siga as instruções online.

   Parte do procedimento de inscrição envolve receber uma chamada telefônica ou uma mensagem de texto e inserir um código de verificação pelo teclado do telefone.

   Quando você se inscreve em um Conta da AWS, um Usuário raiz da conta da AWSé 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, você pode visualizar a atividade atual da sua conta e gerenciar sua conta acessando https://aws.amazon.com/e escolhendo 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 Centro de Identidade do AWS IAM, 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 Console de gerenciamento da AWScomo proprietário da conta escolhendo Usuário raiz e inserindo seu endereço de Conta da AWS e-mail. Na próxima página, insira a senha.

   Para obter ajuda ao fazer login usando o usuário-raiz, consulte 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. Habilita o Centro de Identidade do IAM.

   Para obter instruções, consulte Habilitar o Centro de Identidade do AWS IAM no Guia do usuário do Centro de Identidade do AWS IAM .
2. No Centro de Identidade do IAM, conceda o acesso administrativo a 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 Centro de Identidade do AWS IAM do usuário.

Iniciar sessão como o usuário com acesso administrativo

- Para fazer login com o seu usuário do Centro de Identidade do IAM, use o URL de login enviado ao seu endereço de e-mail quando o usuário do Centro de Identidade do IAM foi criado.

  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 Criar um conjunto de permissões no Guia do usuário do Centro de Identidade do AWS IAM .
2. Atribua usuários a um grupo e, em seguida, atribua o acesso de logon único ao grupo.

   Para obter instruções, consulte Adicionar grupos no Guia do usuário do Centro de Identidade do AWS IAM .

> **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 obter 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. Abra 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 Crie um modelo de AWS veículo de IoT FleetWise.

Defina suas configurações AWS de IoT FleetWise

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 rastrear métricas, como para determinar se há um limite de serviço excedido. Para obter mais informações sobre CloudWatch métricas, consulteMonitore AWS a IoT com a Amazon FleetWise CloudWatch.

Com o CloudWatch Logs, AWS a 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, consulteConfigurar o AWS registro de 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 na AWS IoT FleetWise.

Definir configurações (console)

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

1. Abra o console AWS de IoT. FleetWise
2. No painel esquerdo, selecione Configurações.
3. Em Métricas, escolha Ativar. AWS A 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 obter mais informações, consulte Gerenciamento de chaves na AWS IoT FleetWise.
        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.

Configuração de permissão do IAM para registro de conta

Para invocar a `RegisterAccount` API com sucesso, você precisa incluí-la `iam:CreateServiceLinkedRole` em seu documento de política do IAM. Essa API cria uma função vinculada ao serviço em sua conta que é usada para publicar métricas de AWS FleetWise IoT em sua. CloudWatch Para verificar se a conta foi registrada com sucesso, invoque a `GetRegisterAccountStatus` API e verifique se o status do registro é`REGISTRATION_SUCCESS`.

O exemplo a seguir mostra um exemplo de documento de política para configurar permissões para `RegisterAccount` e`GetRegisterAccountStatus`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:RegisterAccount",
        "iotfleetwise:GetRegisterAccountStatus",
        "iam:CreateServiceLinkedRole"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

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 é usada somente para publicar métricas de AWS FleetWise IoT em. CloudWatch Para obter mais informações, consulte Usando funções vinculadas a serviços para o AWS IoT FleetWise.

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

Example resposta

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

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.

Fazendo solicitações para a AWS IoT FleetWise usando IPv6

Você pode se comunicar com o AWS IoT via FleetWise Internet Protocol versão 6 (IPv6) e IPv4 gerenciar seus recursos. Os endpoints de pilha dupla oferecem suporte a solicitações de IoT AWS repetidamente. FleetWise APIs IPv6 IPv4 Não há cobranças adicionais pela comunicação IPv6.

O IPv6 protocolo é o padrão IP da próxima geração com recursos de segurança adicionais. Ele oferece espaço de endereço longo de 128 bits, enquanto IPv4 tem um endereço longo de 32 bits. IPv4 pode gerar 4,29 x 10^9 endereços, enquanto IPv6 pode ter 3,4 x 10^38 endereços.

IPv6 pré-requisitos para pontos finais do plano de controle

IPv6 o suporte ao protocolo é ativado automaticamente para pontos finais do plano de controle. Ao usar os endpoints para clientes do ambiente de gerenciamento, é necessário fornecer a extensão SNI (Indicação de nome do servidor). Os clientes podem usar a extensão SNI para indicar o nome do servidor que está sendo contatado e se ele está usando os endpoints regulares ou os endpoints de pilha dupla. Consulte Usar endpoints de pilha dupla.

IPv6 suporte para AWS PrivateLink endpoints

AWS A IoT FleetWise oferece suporte à IPv6 comunicação para fazer interface com endpoints de VPC usando. AWS PrivateLink

Testando a compatibilidade IPv6 de endereços

Se você estiver usando use Linux/Unix ou Mac OS X, pode testar se é possível acessar um endpoint de pilha dupla IPv6 usando o comando curl, conforme mostrado no exemplo a seguir:

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

Você recebe de volta informações semelhantes ao exemplo a seguir. Se você estiver conectado IPv6, o endereço IP conectado será um IPv6 endereço.

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

Se você estiver usando o Microsoft Windows 7 ou o Windows 10, poderá testar se é possível acessar um endpoint de pilha dupla usando IPv6 ou IPv4 usando o comando ping, conforme mostrado no exemplo a seguir.

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

Usando IPv6 endereços nas políticas do IAM

Antes de usar IPv6 seus recursos, você deve garantir que todas as políticas do IAM usadas para filtragem de endereços IP incluam intervalos de IPv6 endereços. Para obter mais informações sobre como gerenciar permissões de acesso com o IAM, consulte Identity and Access Management para AWS IoT FleetWise.

As políticas do IAM que filtram endereços IP usam Operadores de condição de endereço IP.
A política a seguir identifica o `54.240.143.*` intervalo de IPv4 endereços permitidos usando operadores de condição de endereço IP. Como todos os IPv6 endereços estão fora do intervalo permitido, essa política impede a comunicação usando IPv6 endereços.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "iotfleetwise:*",
      "Resource": "arn:aws:iotfleetwise:us-east-1:111122223333:*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "54.240.143.0/24"}
      }
    }
  ]
}
```

Para incluir IPv6 endereços, você pode modificar o elemento Condição da política para permitir intervalos de endereços IPv4 (54.240.143.0/24) e IPv6 (2001:: 1234:5678DB8: :/64), conforme mostrado no exemplo a seguir.

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

Usar endpoints de pilha dupla

AWS Os endpoints de duas FleetWise pilhas de IoT oferecem suporte a solicitações de IoT em e. AWS FleetWise APIs IPv6 IPv4 Quando você faz uma solicitação para um endpoint de pilha dupla, ela é automaticamente resolvida para um endereço ou. IPv4 IPv6 No modo de pilha dupla, tanto as IPv4 conexões quanto as IPv6 do cliente são aceitas.

Se você estiver usando a API REST, poderá acessar diretamente um endpoint de AWS IoT usando o nome do FleetWise endpoint (URI). AWS A IoT FleetWise suporta somente nomes de endpoints regionais de pilha dupla, o que significa que você deve especificar o como parte do Região da AWS nome.

A tabela a seguir mostra o formato dos endpoints do plano de controle para AWS FleetWise IoT durante o IPv4 uso e os modos de pilha dupla. Para obter mais informações sobre esses endpoints, consulte Endpoints de AWS FleetWise IoT.

Endpoint	IPv4 endereço	Modo de pilha dupla
Ambiente de gerenciamento	Fleetwise. *<region>*.amazonaws.com	Fleetwise. *<region>*.api.aws

Ao usar o AWS CLI and AWS SDKs, você pode usar uma variável de `AWS_USE_DUALSTACK_ENDPOINT` ambiente ou o `use_dualstack_endpoint` parâmetro, que é uma configuração de arquivo de configuração compartilhada, para mudar para um endpoint de pilha dupla. Você também pode especificar o endpoint de pilha dupla diretamente como uma substituição do endpoint de AWS FleetWise IoT no arquivo de configuração. Para obter mais informações, consulte Dual-stack and FIPS endpoints.

Ao usar o AWS CLI, você pode definir o valor da configuração `use_dualstack_endpoint` como `true` em um perfil no seu AWS arquivo Config. Isso direcionará todas as FleetWise solicitações de AWS IoT feitas pelos comandos para o endpoint de pilha dupla da região especificada. Especifique a região no arquivo de configuração ou em um comando usando a opção `--region`.

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

Em vez de usar os endpoints de pilha dupla para todos os comandos, use esses endpoints para comandos específicos:

- Você pode usar o endpoint de pilha dupla para comandos específicos definindo o `--endpoint-url` parâmetro para esses comandos. Por exemplo, no comando a seguir, você pode substituir o *<endpoint-url>* para`iotfleetwise.`*<region>*`.api.aws`.

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

- Você pode configurar perfis separados no seu AWS arquivo Config. Por exemplo, crie um perfil definido `use_dualstack_endpoint` como verdadeiro e um perfil que não seja definido`use_dualstack_endpoint`. Quando executar um comando, especifique qual perfil deseja usar, dependendo de querer ou não usar o endpoint de pilha dupla.

Tutorial: Comece a usar a AWS IoT FleetWise

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

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

- Ingira dados AWS de FleetWise IoT na nuvem
- Modelos de AWS veículos de IoT FleetWise
- Gerencie AWS veículos de IoT FleetWise
- Gerencie frotas na IoT AWS FleetWise
- Colete FleetWise dados de AWS IoT com campanhas

Introdução

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. Principais conceitos e recursos da 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.

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 Gerenciamento de contas 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). Você pode usar o seletor de região no Console de gerenciamento da AWS para alternar para uma dessas regiões. Para obter mais informações, consulte FleetWise endpoints e cotas de AWS IoT.
- 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.
- A demonstração do software Edge Agent. (As instruções para configurar a demonstração estão na próxima etapa.)
  - 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 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 baixar a demonstração, navegue até o console de AWS IoT. FleetWise Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.

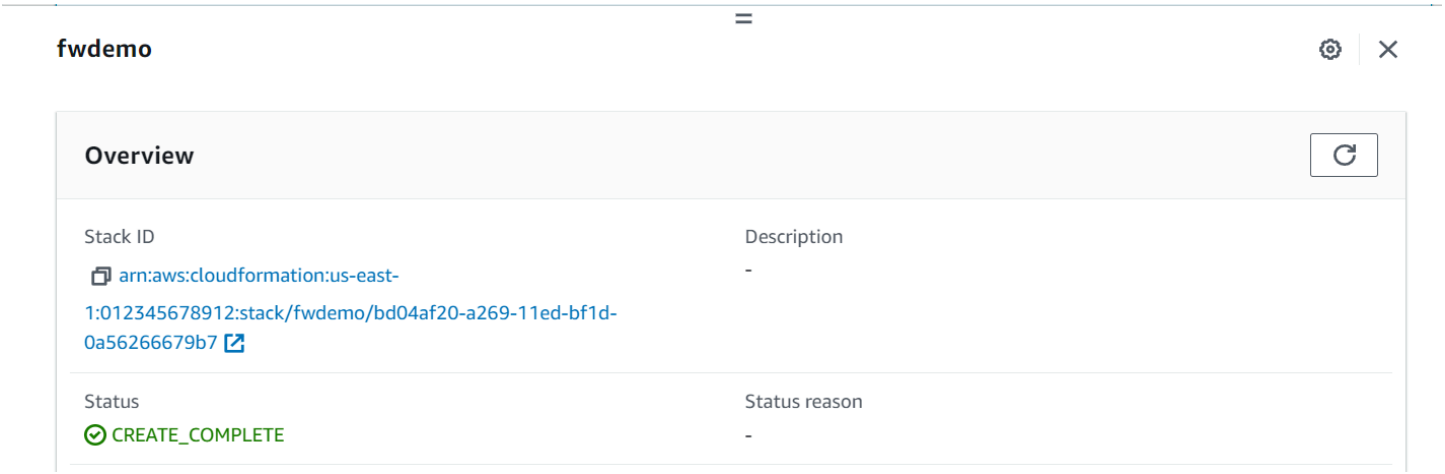
Etapa 1: Configurar o software Edge Agent para AWS IoT FleetWise

> **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 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. TCoreRegião de Io - Você pode especificar a região em AWS IoT que a coisa é criada atualizando o parâmetro TCore Região de Io. 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 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.



> **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 suporta 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

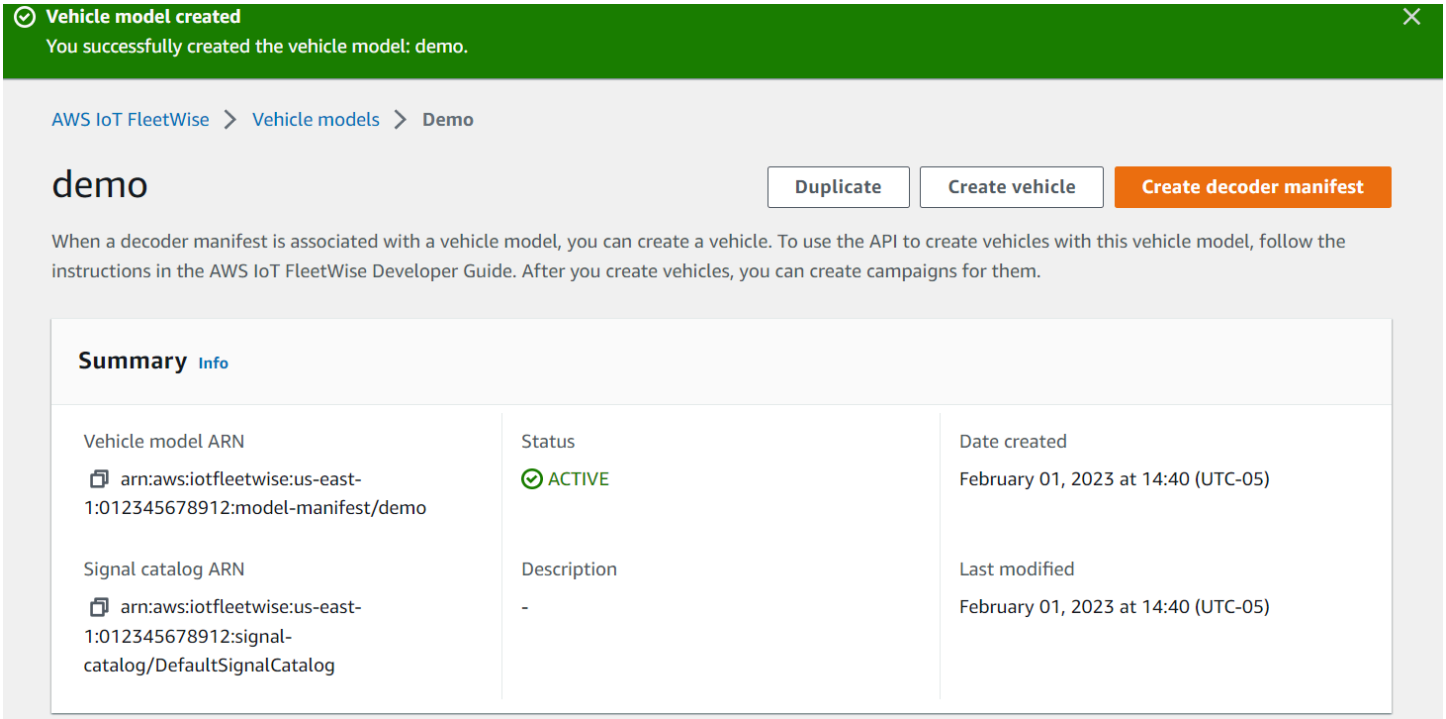
2. No painel de navegação, selecione Modelos de veículos.
3. Na página Modelos de veículos, selecione Criar modelo de veículo.
4. Na seção Informações gerais, insira o nome do modelo do seu veículo, como Veículo1, e uma descrição opcional. Escolha Próximo.
5. Escolha um ou mais sinais do catálogo de 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.



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 obter mais informações, consulte Criar um manifesto do decodificador (AWS CLI).

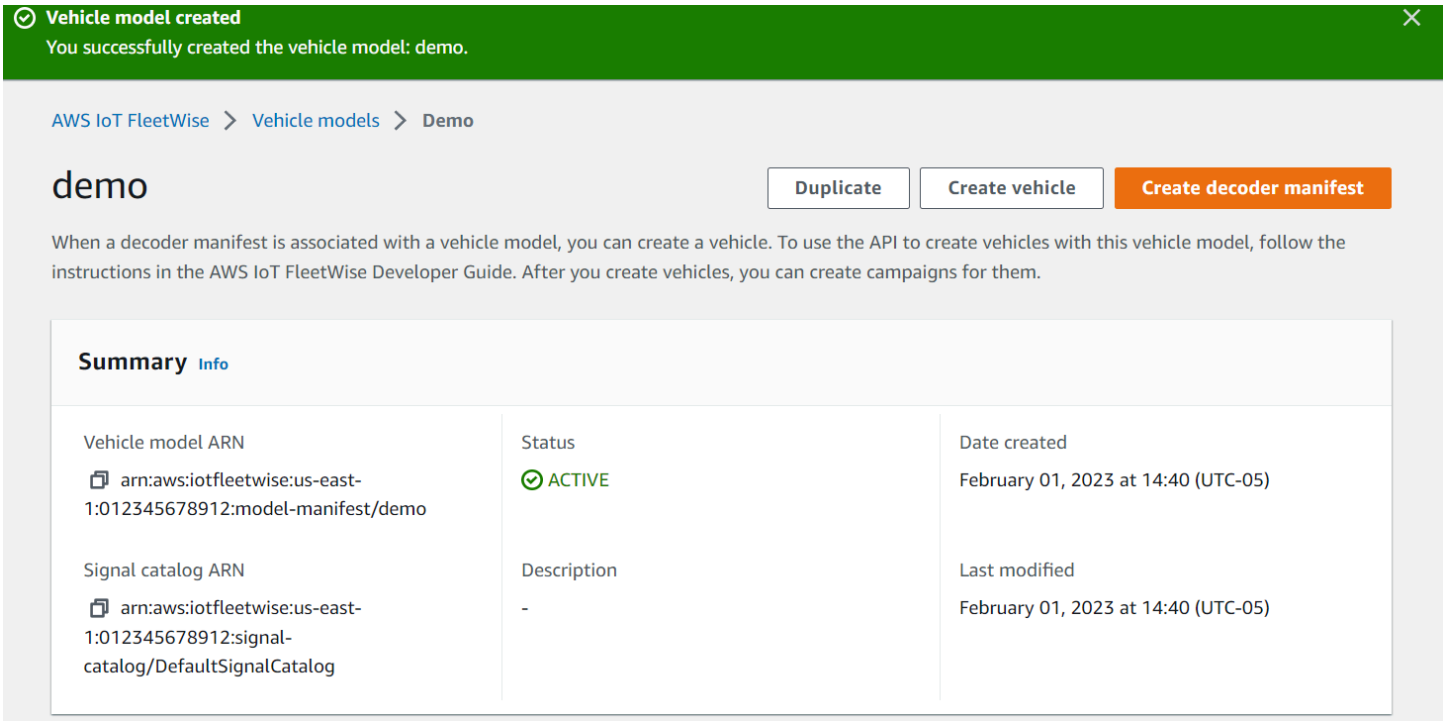
1. Para ajudar a identificar o manifesto do decodificador, insira um nome e uma descrição opcional para ele. Em seguida, escolha Próximo.
2. Para adicionar uma ou mais interfaces de rede, escolha o tipo CAN_INTERFACE ou OBD_INTERFACE.
   - Interface de diagnóstico a bordo (OBD) - Escolha esse tipo de interface se quiser um protocolo que defina como os dados de autodiagnóstico são comunicados entre as unidades de controle eletrônico (). ECUs Esse protocolo fornece vários códigos de problemas de diagnóstico padrão (DTCs) que podem ajudá-lo a solucionar problemas com seu veículo.
   - Interface de rede de área de controle (barramento CAN) - Escolha esse tipo de interface se quiser um protocolo que defina como os dados são comunicados entre ECUs eles. 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.



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, escolha 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 CloudFormation pilha e todos os recursos da pilha.

Para excluir a CloudFormation pilha

1. Abra o console do CloudFormation.
2. Na lista de pilhas, escolha a pilha que você criou na etapa 1.
3. Escolha Excluir.
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 obter mais informações, consulte Visualize dados de veículos AWS de FleetWise IoT.
2. Você pode solucionar e resolver problemas com a AWS IoT. FleetWise Para obter mais informações, consulte Solução de problemas de AWS IoT FleetWise.

Ingira dados AWS de FleetWise IoT na nuvem

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

> **Note**
>
> - AWS FleetWise A IoT não se destina ao uso ou em associação com a operação de 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 da AWS IoT FleetWise são apenas para fins informativos, e você não pode usar a AWS IoT FleetWise para controlar ou operar as funções do veículo.
> - Os dados do veículo coletados por meio do uso da AWS IoT FleetWise devem ser avaliados quanto à precisão, conforme apropriado para seu caso de uso, inclusive para fins de cumprimento de quaisquer obrigações de conformidade que você possa ter de acordo com os regulamentos de segurança veicular aplicáveis (como obrigações de monitoramento e emissão de relatórios de segurança). 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 seu FleetWise software Edge Agent para AWS IoT em seu veículo. Para obter mais informações sobre como trabalhar com o software Edge Agent, faça o seguinte para baixar o Guia do desenvolvedor do FleetWise software Edge Agent para AWS IoT.
    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.

2. Crie ou importe um catálogo de sinais contendo sinais que você usará para criar um modelo de veículo. Para obter mais informações, consulte Crie um catálogo de AWS sinais de IoT FleetWise 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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
> - AWS FleetWise Atualmente, a IoT suporta um catálogo de sinais para cada AWS conta por. Região da AWS

3. Use sinais no catálogo de sinais para criar um modelo de veículo. Para obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.

> **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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
> - 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 um modelo AWS de veículo de IoT FleetWise.
> - 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.

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 Gerencie manifestos AWS de decodificadores de IoT FleetWise.

> **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 AWS do decodificador de IoT FleetWise.
> - 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.

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 Gerencie AWS veículos de IoT FleetWise.
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 Gerencie frotas na IoT AWS FleetWise.
7. (Opcional) 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 Colete FleetWise dados de AWS IoT com campanhas. Você pode criar campanhas, modelos de estado (abaixo) ou ambos para coletar dados.

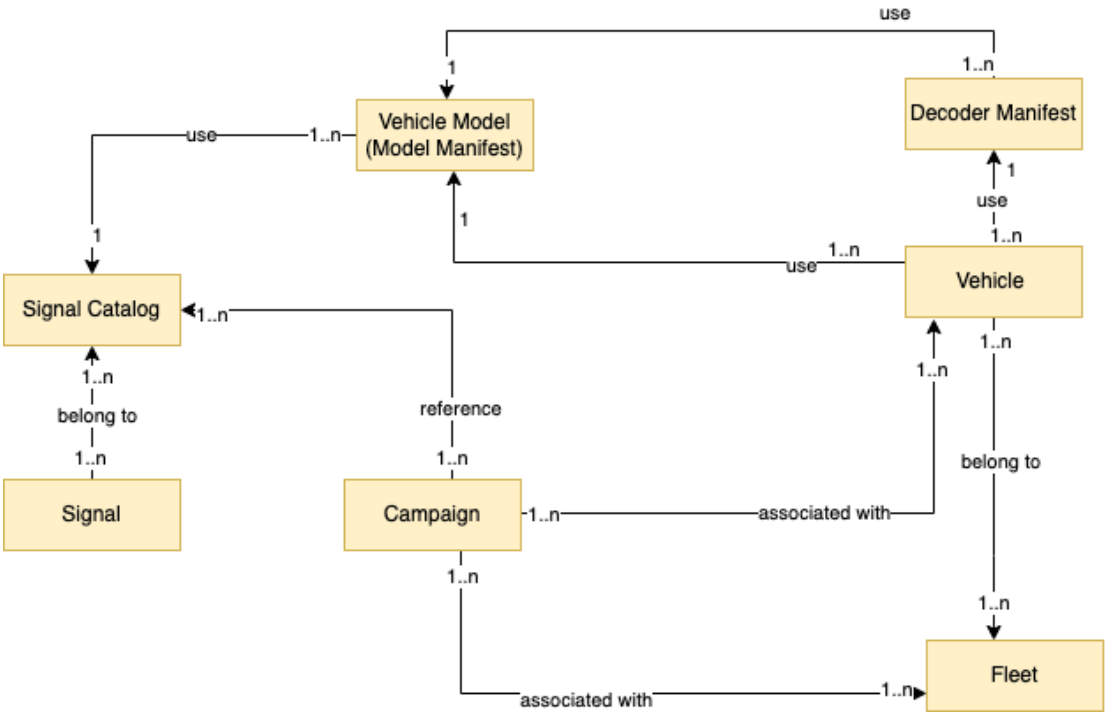
   > **Note**
   >
   > Você deve usar a operação de `UpdateCampaign` API para aprovar a campanha antes que a AWS FleetWise IoT possa implantá-la no veículo ou na frota. Para obter mais informações, consulte Atualizar uma campanha AWS de IoT FleetWise.

8. (Opcional) Crie modelos de estado. Os modelos de estado são implantados em um veículo. Os modelos de estado fornecem um mecanismo para que os proprietários de veículos rastreiem o estado de seus veículos. Para obter mais informações, consulte Monitorar o último estado conhecido de seus veículos.

O software Edge Agent transfere os dados do veículo para AWS IoT Core usar um tópico do MQTT que você escolher. Para enviar os dados para a AWS IoT FleetWise para campanhas, ele usa o tópico reservado. `$aws/iotfleetwise/vehicles/vehicleName/signals` Para Last Known State, o Edge Agent usa o tópico reservado`$aws/iotfleetwise/vehicles/vehicleName/last_known_states/data`. Para obter mais informações sobre como os dados ingeridos são processados, consulteVisualize dados de veículos AWS de FleetWise IoT.

Modelos de AWS veículos de IoT FleetWise

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 obter mais informações, consulte Gerencie AWS catálogos de sinais de IoT FleetWise.

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 Gerencie AWS catálogos de sinais de IoT FleetWise.

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 Gerencie modelos AWS de veículos de IoT FleetWise.

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 obter mais informações, consulte Gerencie manifestos AWS de decodificadores de IoT FleetWise.

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 obter mais informações, consulte Crie um catálogo de AWS sinais de IoT FleetWise 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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
   > - 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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.

> **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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
> - 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 um modelo AWS de veículo de IoT FleetWise.
> - 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 obter mais informações, consulte Gerencie manifestos AWS de decodificadores de IoT FleetWise.

> **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 AWS do decodificador de IoT FleetWise.
> - 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 decodificadores de sinal. 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

- Gerencie AWS catálogos de sinais de IoT FleetWise
- Gerencie modelos AWS de veículos de IoT FleetWise
- Gerencie manifestos AWS de decodificadores de IoT FleetWise

Gerencie AWS catálogos de sinais de IoT FleetWise

> **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.

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. 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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
> - 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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.
> - 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 AWS sinais de IoT FleetWise
- Crie um catálogo de AWS sinais de IoT FleetWise
- Importar um catálogo AWS de sinais de IoT FleetWise
- Atualizar um catálogo AWS de sinais de IoT FleetWise
- Excluir um catálogo AWS de sinais de IoT FleetWise
- Obtenha informações AWS do catálogo de FleetWise sinais de IoT

Configurar AWS sinais de IoT FleetWise

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,,,,,,, BOOLEAN, FLOAT, DOUBLE UINT8 INT16, STRING UINT16 INT32 UINT32, UNIX_TIMESTAMP INT64 UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 INT16 BOOLEAN_ARRAY, UINT16 FLOAT_ARRAY, INT32 DOUBLE_ARRAY, UINT32 STRING_ARRAY, INT64 UINT64 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,,,,,,,, BOOLEAN, FLOAT, DOUBLE UINT8 INT16, STRING UINT16 INT32, UNIX_TIMESTAMP UINT32 INT64 UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 _ARRAY, UINT8 INT16 BOOLEAN_ARRAY, UINT16 FLOAT_ARRAY, INT32 DOUBLE_ARRAY, UINT32 STRING_ARRAY, INT64 UINT64 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 estrutura, como um tipo de dados primitivo (como UINT8) ou outra estrutura (como timestamp). 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 complexo 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,,,,,,,, BOOLEAN, FLOAT, DOUBLE, STRING UINT8 INT16 UINT16, UNIX_TIMESTAMP INT32 UINT32 INT64, _ARRAY UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 BOOLEAN_ARRAY, UINT8 INT16 FLOAT_ARRAY, UINT16 DOUBLE_ARRAY, INT32 STRING_ARRAY, UINT32 INT64 UNIX_TIMESTAMP_ARRAY, STRUCT UINT64 T_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).

Crie um catálogo de AWS sinais de IoT FleetWise

Você pode usar a operação CreateSignalCatalogda 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) *`description`* Substitua por uma descrição para ajudá-lo a identificar o catálogo de sinais.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte Configurar AWS sinais de IoT FleetWise.

```
{
    "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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateSignalCatalog` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Importar um catálogo AWS de sinais de IoT FleetWise

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 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 ImportSignalCatalogAPI 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 a descrição por uma *description* para ajudá-lo 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 AWS sinais de IoT FleetWise.

```
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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ImportSignalCatalog` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualizar um catálogo AWS de sinais de IoT FleetWise

Você pode usar a operação UpdateSignalCatalogda 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 AWS sinais de IoT FleetWise.

> **Important**
>
> Estruturas personalizadas são imutáveis. Se você precisar reordenar ou inserir propriedades em uma estrutura personalizada existente (estrutura), exclua a estrutura e crie uma estrutura 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
      }
     }]
    }
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateSignalCatalog` API.

JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Verifique a atualização do catálogo de sinais

Você pode usar a operação ListSignalCatalogNodesda 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListSignalCatalogNodes` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
```

```
            ]
        }
    ]
}
```

Excluir um catálogo AWS de sinais de IoT FleetWise

Você pode usar a operação DeleteSignalCatalogda 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 instruções, consulte:
>
> - Excluir um modelo AWS de veículo de IoT FleetWise
> - Excluir um manifesto AWS do decodificador de IoT FleetWise
> - Excluir um veículo AWS de IoT FleetWise
> - Excluir uma frota AWS de IoT FleetWise
> - Excluir uma campanha AWS de IoT FleetWise

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
```

Verifique a exclusão do catálogo de sinais

Você pode usar a operação ListSignalCatalogsda 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListSignalCatalogs` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Obtenha informações AWS do catálogo de FleetWise sinais de IoT

Você pode usar a operação da GetSignalCatalogAPI 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `GetSignalCatalog` API.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

> **Note**
>
> Essa operação é eventualmente consistente. Em outras palavras, as alterações no catálogo de sinais podem não ser refletidas imediatamente.

Gerencie modelos AWS de veículos de IoT FleetWise

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 obter mais informações, consulte Gerencie AWS veículos de IoT FleetWise.

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**
>
> - Você deve ter um catálogo de sinais antes de criar um modelo de veículo usando a operação da `CreateModelManifest` API. Para obter mais informações, consulte Crie um catálogo de AWS sinais de IoT FleetWise.

- 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

- Crie um modelo de AWS veículo de IoT FleetWise
- Atualizar um modelo AWS de veículo de IoT FleetWise
- Excluir um modelo AWS de veículo de IoT FleetWise
- Obtenha informações sobre modelos de FleetWise veículos AWS de IoT

Crie um modelo de AWS veículo de IoT FleetWise

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

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 obter mais informações, consulte Gerencie manifestos AWS de decodificadores de IoT FleetWise.

Para criar um modelo de veículo usando um modelo

1. Abra 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 Adicionar.

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. 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 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).
   - Tipo de dados — O tipo de dados do sinal deve ser um dos seguintes: INT8,,,,,, BOOLEAN, FLOAT, DOUBLE, STRING UINT8 INT16, UNIX_TIMESTAMP UINT16 INT32 UINT32 INT64, _ARRAY UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 BOOLEAN_ARRAY, UINT8 INT16 FLOAT_ARRAY, UINT16 DOUBLE_ARRAY, INT32 STRING_ARRAY, UINT32 INT64 UNIX_TIMESTAMP_ARRAY ou UINT64 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. Da mesma forma JSONPath, use um ponto (.) para se referir ao sinal de uma criança. 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)
   - Tipo de dados — O tipo de dados do atributo deve ser um dos seguintes: INT8,,,,,,, BOOLEAN, FLOAT, DOUBLE UINT8, STRING INT16 UINT16 INT32, UNIX_TIMESTAMP UINT32 INT64 UINT64, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, _ARRAY, INT8 BOOLEAN_ARRAY, UINT8 FLOAT_ARRAY, INT16 DOUBLE_ARRAY, UINT16 INT32 STRING_ARRAY, UINT32 INT64 UNIX_TIMESTAMP_ARRAY ou UINT64 UNKNOWN
   - (Opcional) Unit — A unidade científica do atributo, como km ou Celsius.
   - (Opcional) Path — O caminho para o sinal. Da mesma forma JSONPath, use um ponto (.) para se referir ao sinal de uma criança. 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. Abra 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**
>
> Você deve ter um catálogo de sinais antes de criar um modelo de veículo usando a operação da `CreateModelManifest` API. Para obter mais informações sobre como criar um catálogo de sinais, consulte Crie um catálogo de AWS sinais de IoT FleetWise.

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.
- *`signal-catalog-ARN`*Substitua pelo Amazon Resource Name (ARN) do catálogo de sinais.
- (Opcional) *`description`* Substitua por uma descrição para ajudá-lo a identificar o modelo do veículo.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte Configurar AWS sinais de IoT FleetWise.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateModelManifest` API.

JSON

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

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualizar um modelo AWS de veículo de IoT FleetWise

Você pode usar a operação da UpdateModelManifestAPI 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 porACTIVE.

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

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

```
{
    "name": "vehicle-model-name",
```

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateModelManifest` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Verifique a atualização do modelo do veículo

Você pode usar a operação ListModelManifestNodesda API para verificar se um modelo de 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListModelManifestNodes` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Excluir um modelo AWS de veículo de IoT FleetWise

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 obter mais informações, consulte Excluir um veículo AWS de IoT FleetWise e Excluir um manifesto AWS do decodificador de IoT FleetWise.

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. Abra 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 Excluir.
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 DeleteModelManifestAPI 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
```

Verifique a exclusão do modelo do veículo

Você pode usar a operação ListModelManifestsda 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListModelManifests` API.

JSON

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

```
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Obtenha informações sobre modelos de FleetWise veículos AWS de IoT

Você pode usar a operação GetModelManifestda API para recuperar informações sobre o modelo de um veículo. O exemplo a seguir usa AWS CLI.

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

*vehicle-model*Substitua 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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `GetModelManifest` API.

JSON

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

```
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Gerencie manifestos AWS de decodificadores de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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 decodificadores de sinal 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 A 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). ECUs pode ser a unidade de controle do motor, os airbags ou o sistema de áudio.

On-board diagnostic (OBD) II

Um protocolo mais desenvolvido que define como os dados de autodiagnóstico são comunicados entre eles. ECUs Ele fornece vários códigos de problemas de diagnóstico padrão (DTCs) que ajudam a identificar o que há de errado com seu 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.

Interfaces personalizadas

Você também pode usar sua própria interface para decodificar sinais no Edge. Isso pode economizar seu tempo, pois você não precisa criar regras de decodificação na nuvem.

Decodificador de sinal

Fornece informações detalhadas de decodificação para um sinal específico. Cada sinal especificado no modelo do veículo deve ser emparelhado com um decodificador de sinal. 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 decodificadores de sinal OBD.

O manifesto do decodificador deve conter decodificadores de sinal de mensagem se também contiver interfaces de middleware do veículo. Ou, se o manifesto do decodificador contiver interfaces de decodificação personalizadas, ele também deverá conter sinais de decodificação personalizados.

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 o ListDecoderManifests e GetDecoderManifest APIs 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 FleetWise rede AWS de IoT e sinais de decodificador
- Crie um manifesto AWS do decodificador de IoT FleetWise
- Atualizar um manifesto AWS do decodificador de IoT FleetWise
- Excluir um manifesto AWS do decodificador de IoT FleetWise
- Obtenha informações sobre o manifesto do FleetWise decodificador AWS de IoT

Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

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

Tópicos

- Configurar interfaces de rede
- Configurar decodificadores de sinal

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 coleta os arquivos armazenados DTCs 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 World Wide Harmonization On-Board Diagnostics (WWH-OBD) -4. ISO15765
- (Opcional) `pidRequestIntervalSeconds` — Com que frequência solicitar o OBD II PIDs do veículo. Por exemplo, se o valor especificado for 120, o software Edge Agent coletará o OBD II PIDs uma vez a cada 2 minutos.
- (Opcional) `useExtendedIds` — Se deve usar estendido IDs 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: `ROS_2`

Para configurar uma interface de decodificação personalizada, especifique as informações a seguir.

- `name`— O nome do seu decodificador que você usa para decodificar sinais no Edge.

  O nome da interface do decodificador pode ter de 1 a 100 caracteres.

Configurar decodificadores de sinal

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

- `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 tamanho da mensagem em bits.
- `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.
- (Opcional) `signalValueType` — O tipo de valor do sinal. Integer é o tipo de valor padrão.

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

- `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.
- (Opcional) `isSigned` — Se a mensagem está assinada. Se estiver assinada, a mensagem poderá representar números positivos e negativos. A mensagem não é assinada por padrão (`false`).

- (Opcional) `signalValueType` — O tipo de valor do sinal. Integer é o tipo de valor padrão.

Para configurar um decodificador de sinal de mensagem, 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.

Para configurar um sinal de decodificação personalizado, especifique as informações a seguir.

- (Opcional) `id` — O ID do sinal que você mesmo decodifica usando a interface do decodificador. O ID do sinal pode ter de 1 a 150 caracteres. Se não for especificado, o `id` padrão é o `fullyQualifiedName` do sinal.

Crie um manifesto AWS do decodificador de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

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. Abra 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: Mapear a interface CAN
- 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.

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

1. Faça upload de um arquivo de interface de rede. Você pode carregar um arquivo.dbc para protocolos CAN ou um arquivo.json para ROS 2 ou interfaces personalizadas.
2. Insira um nome para sua interface de rede. Se você fez upload de uma interface personalizada, o nome já foi fornecido.

Mapeie os sinais ausentes

Se houver sinais no modelo do veículo sem decodificadores de sinal emparelhados nas interfaces de rede carregadas, você poderá criar um decodificador personalizado padrão que mapeará os sinais ausentes. Isso é opcional, pois você pode mapear manualmente os sinais na próxima etapa.

Para criar um decodificador personalizado padrão

1. Selecione Criar decodificador personalizado padrão para sinais perdidos.
2. Escolha Próximo.

Etapa 2: Mapear a interface CAN

Você pode mapear os sinais CAN com decodificadores de sinal CAN. Se você marcou a caixa de seleção Criar decodificador personalizado padrão para sinais ausentes, todos os sinais que não tiverem um sinal de decodificador serão automaticamente mapeados para decodificadores de sinal personalizados padrão.

Para mapear sinais CAN

1. No mapeamento de sinal CAN, selecione um decodificador de sinal.
2. Escolha Próximo.

> **Note**
>
> Se você adicionou um ROS 2 ou uma interface personalizada, você pode verificar os mapeamentos antes de criar o manifesto do decodificador.

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 CreateDecoderManifestAPI para criar manifestos do decodificador. O exemplo a seguir usa a AWS CLI.

> **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 obter mais informações, consulte Crie um modelo de AWS veículo de IoT FleetWise.

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.
- *vehicle-model-ARN*Substitua pelo Amazon Resource Name (ARN) do modelo do veículo.
- (Opcional) *description* Substitua por uma descrição para ajudá-lo a identificar o manifesto do decodificador.

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador.

```
{
    "name": "decoder-manifest-name",
    "modelManifestArn": "vehicle-model-arn",
    "description": "description",
    "networkInterfaces": [
        {
            "canInterface": {
                "name": "myNetworkInterface",
                "protocolName": "CAN",
                "protocolVersion": "2.0b"
            },
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_INTERFACE"
        }
    ],
    "signalDecoders": [
        {
            "canSignal": {
                "name": "Engine_Idle_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
                "offset": 0,
                "startBit": 16
            },
            "fullyQualifiedName": "Vehicle.EngineIdleTime",
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        },
        {
            "canSignal": {
                "name": "Engine_Run_Time",
                "factor": 1,
                "isBigEndian": true,
                "isSigned": false,
                "length": 24,
                "messageId": 271343712,
                "offset": 0,
                "startBit": 40
            },
            "fullyQualifiedName": "Vehicle.EngineRunTime",
```

```
            "interfaceId": "Qq1acaenByOB3sSM39SYm",
            "type": "CAN_SIGNAL"
        }
    ]
}
```

- *decoder-manifest-name*Substitua pelo nome do manifesto do decodificador que você está criando.
- *vehicle-model-ARN*Substitua pelo Amazon Resource Name (ARN) do modelo do veículo.
- (Opcional) *description* Substitua por uma descrição para ajudá-lo 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 FleetWise rede AWS de IoT e sinais de decodificador.

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

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

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

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

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

Para obter mais informações sobre como configurar ramificações, atributos, sensores e atuadores, consulte Configurar interfaces de FleetWise rede AWS de IoT e sinais de decodificador.

```
{
 "name": "decoder-manifest-name",
 "modelManifestArn": "vehicle-model-arn",
 "description": "description",
 "networkInterfaces": [
        {
      "interfaceId": "myCustomInterfaceId",
      "type": "CUSTOM_DECODING_INTERFACE",
           "customDecodingInterface": {
```

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

> **Note**
>
> 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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateDecoderManifest` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualizar um manifesto AWS do decodificador de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação da UpdateDecoderManifestAPI 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
```

Se os sinais não tiverem regras de decodificação especificadas, você poderá criar regras de decodificação padrão. Os sinais são adicionados a uma interface decodificada personalizada com o `CustomDecodingSignal$id` nome totalmente qualificado do sinal. Para atualizar um manifesto do decodificador com as regras de decodificação padrão, 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
                --default-for-unmapped-signals CUSTOM_DECODING
```

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateDecoderManifest` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
```

```
    ]
}
```

Verifique a atualização do manifesto do decodificador

Você pode usar a operação da ListDecoderManifestSignalsAPI 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListDecoderManifestSignals` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Você pode usar a operação da ListDecoderManifestNetworkInterfacesAPI 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListDecoderManifestNetworkInterfaces` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Excluir um manifesto AWS do decodificador de IoT FleetWise

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 obter mais informações, consulte Excluir um veículo AWS de IoT FleetWise.

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. Abra 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 DeleteDecoderManifestAPI 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 obter mais informações, consulte Excluir um veículo AWS de IoT FleetWise.

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
```

Verifique a exclusão do manifesto do decodificador

Você pode usar a operação da ListDecoderManifestsAPI 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListDecoderManifests` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Obtenha informações sobre o manifesto do FleetWise decodificador AWS de IoT

Você pode usar a operação da GetDecoderManifestAPI para verificar se as interfaces de rede e os decodificadores de sinal 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.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `GetDecoderManifest` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Gerencie AWS veículos de IoT FleetWise

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 Modelos de AWS veículos de IoT FleetWise.

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 AWS veículos de IoT FleetWise
- Tópicos reservados em AWS IoT FleetWise
- Crie um veículo AWS de IoT FleetWise
- Crie vários veículos AWS de IoT FleetWise
- Atualizar um veículo AWS de IoT FleetWise
- Atualize vários veículos AWS de IoT FleetWise
- Excluir um veículo AWS de IoT FleetWise
- Obtenha informações sobre AWS veículos de IoT FleetWise

Provisionar AWS veículos de IoT FleetWise

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 obter mais informações, consulte Crie um veículo AWS de IoT FleetWise.

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.
  - *policy-name*Substitua pelo nome da política que você deseja criar.
  - *file-name*Substitua pelo nome do arquivo JSON que contém a AWS IoT Core política.

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

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

- *us-east-1*Substitua pela AWS região em que você criou recursos de AWS IoT. FleetWise
- *111122223333*Substitua pelo ID AWS da sua conta.

Este exemplo inclui tópicos reservados pela AWS IoT FleetWise. Você deve adicionar os tópicos à política. Para obter mais informações, consulte Tópicos reservados em AWS IoT FleetWise.

JSON

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

```
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:topic/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/checkins",
                "arn:aws:iot:us-east-1:111122223333:topic/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/signals"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:topicfilter/$aws/
iotfleetwise/vehicles/${iot:Connection.Thing.ThingName}/collection_schemes",
                "arn:aws:iot:us-east-1:111122223333:topicfilter/$aws/
iotfleetwise/vehicles/${iot:Connection.Thing.ThingName}/decoder_manifests"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:topic/$aws/iotfleetwise/
vehicles/${iot:Connection.Thing.ThingName}/collection_schemes",
                "arn:aws:iot:us-east-1:111122223333:topic/$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.

   - *cert*Substitua pelo nome do arquivo que salva o conteúdo de saída do comando do CertificatePem.
   - *public-key*Substitua pelo nome do arquivo que salva o conteúdo de saída do comando do KeyPair. PublicKey.
   - *private-key*Substitua pelo nome do arquivo que salva o conteúdo de 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.

   - *policy-name*Substitua pelo nome da AWS IoT Core política que você criou.
   - *certificate-arn*Substitua 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.

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

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

Tópicos reservados em AWS IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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	Description	
`$aws/iotfleetwise/vehicles/` *vehicleName* `/checkins`	Publicar	O software Edge Agent publica informações de status do veículo neste tópico.	

Tópico	Operação do cliente permitida	Description	
		As informações de status do veículo são trocadas no formato de buffers de protocolo (Protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise software do Edge Agent para AWS IoT.	
`$aws/iotfleetwise/vehicles/ vehicleName /signals`	Publicar	O software Edge Agent publica sinais para este tópico. As informações do sinal são trocadas no formato de buffers de protocolo (Protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise software do Edge Agent para AWS IoT.	
`$aws/iotfleetwise/vehicles/ vehicleName /collection_schemes`	Assinar	AWS A IoT FleetWise publica esquemas de coleta de dados sobre esse tópico. Os veículos consomem esses esquemas de coleta de dados.	

Tópico	Operação do cliente permitida	Description	
`$aws/iotfleetwise/vehicles/vehicleName/decoder_manifests`	Assinar	AWS A IoT FleetWise publica manifestos de decodificadores para esse tópico. Os veículos consomem esses manifestos do decodificador.	
`$aws/iotfleetwise/vehicles/vehicleName/command/request`	Assinar	AWS A IoT FleetWise publica solicitações para executar comandos neste tópico. Os veículos então consomem essas solicitações de comando.	
`$aws/iotfleetwise/vehicles/vehicleName/command/response`	Publicar	O software Edge Agent publica respostas de comando do veículo a este tópico. As respostas dos comandos são trocadas no formato de buffers de protocolo (Protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise software do Edge Agent para AWS IoT.	

Tópico	Operação do cliente permitida	Description	
`$aws/iotfleetwise/vehicles/vehicleName/command/notification`	Assinar	AWS A IoT FleetWise publica atualizações de status de comando para este tópico. As notificações são enviadas em formato JSON.	
`$aws/iotfleetwise/vehicles/$vehicle_name/last_known_states/config`	Assinar	AWS A IoT FleetWise publica configurações de modelo de estado para este tópico. Os veículos consomem essas configurações de modelo de estado.	
`$aws/iotfleetwise/vehicles/$vehicle_name/last_known_states/data`	Publicar	O software Edge Agent publica dados coletados dos sinais para este tópico.	

Tópico	Operação do cliente permitida	Description	
`$aws/iotfleetwise/vehicles/ $vehicle_name /last_known_state/ $state_template_name /data`	Assinar	AWS A IoT FleetWise publica dados coletados dos sinais configurados no tópico especificado `$state_template_name` . As atualizações podem ser parciais. Por exemplo, se uma associação de modelo de estado contiver vários sinais com a estratégia de atualização em andamento, somente os sinais que foram alterados estarão contidos em uma determinada mensagem. As informações do sinal são trocadas no formato de buffers de protocolo (Protobuf). Para obter mais informações, consulte o Guia do desenvolvedor de FleetWise software do Edge Agent para AWS IoT.	

Crie um veículo AWS de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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 obter mais informações, consulte Gerencie modelos AWS de veículos de IoT FleetWise.
> - O modelo do veículo deve estar associado a um manifesto do decodificador, e o status do manifesto do decodificador deve ser `ACTIVE`. Para obter mais informações, consulte Gerencie manifestos AWS de decodificadores de IoT FleetWise.

Tópicos

- Criar um veículo (console)
- Criar um veículo (AWS CLI)

Criar um veículo (console)

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

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 CreateVehicleda API para criar um veículo. O exemplo a seguir usa a AWS CLI.

Para criar um veículo, execute o seguinte comando.

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

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

Example— 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. *`thing-name`*Substitua pelo nome da coisa que você deseja criar.

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

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

  > **Important**
  >
  > Certifique-se de que a AWS IoT coisa seja provisionada após a criação do veículo. Para obter mais informações, consulte Provisionar AWS veículos de IoT FleetWise.

- *`vehicle-name`*Substitua 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 (:).
- *`model-manifest-ARN`*Substitua pelo ARN do modelo do seu veículo (manifesto do modelo).
- *`decoder-manifest-ARN`*Substitua 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"
    }
}
```

Example— associar um modelo de estado ao veículo

Você pode associar modelos de estado ao veículo para permitir a coleta de atualizações de estado do veículo na nuvem usando o `stateTemplates` campo.

Neste exemplo, *`stateTemplateUpdateStrategy`* pode ser um dos seguintes:

- `periodic`: permite especificar uma taxa fixa na qual o software Edge Agent enviará atualizações de sinal para a nuvem (o software Edge Agent enviará atualizações mesmo que o valor do sinal não tenha sido alterado entre as atualizações).
- `onChange`: O software Edge Agent enviará atualizações de sinal sempre que o sinal mudar.

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

Onde o *`create-vehicle.json`* arquivo contém (por exemplo):

```
{
    "associationBehavior": "associationBehavior",
    "vehicleName": "vehicle-name",
    "modelManifestArn": "model-manifest-ARN",
    "decoderManifestArn": "decoder-manifest-ARN",
    "attributes": {
        "key": "value"
    },
    "stateTemplates": [
        {
            "identifier": "state-template-name",
            "stateTemplateUpdateStrategy": {
                "periodic": {
                    "stateTemplateUpdateRate": {
```

```
                        "unit": "SECOND",
                        "value": 10
                    }
                }
            }
        }
    ]
}
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateVehicle` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Crie vários veículos AWS de IoT FleetWise

Você pode usar a operação BatchCreateVehicleda API para criar vários veículos ao mesmo tempo. O exemplo a seguir usa a AWS CLI.

Para criar vários veículos, execute o seguinte comando.

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

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

Example— 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 Crie um veículo AWS de IoT FleetWise.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `BatchCreateVehicle` API.

JSON

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

```
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualizar um veículo AWS de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação UpdateVehicleda API para atualizar um veículo existente. O exemplo a seguir usa a AWS CLI.

Para atualizar um veículo, execute o seguinte comando.

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

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

Example— configuração do veículo

- *vehicle-name*Substitua pela ID do veículo que você deseja atualizar.
- (Opcional) *model-manifest-ARN* Substitua pelo ARN do modelo do veículo (manifesto do modelo) que você usa para substituir o modelo do veículo em uso.
- (Opcional) *decoder-manifest-ARN* Substitua 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"
        }
}
```

Example— adicionar ou remover modelos de estado associados ao veículo

Você pode associar modelos de estado adicionais ou remover associações existentes do veículo usando os seguintes campos:

- `stateTemplatesToAdd`
- `stateTemplatesToRemove`

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

Onde o *update-vehicle.json* arquivo contém (por exemplo):

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateVehicle` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualize vários veículos AWS de IoT FleetWise

Você pode usar a operação BatchUpdateVehicleda 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.

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

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

Example— 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 de IoT FleetWise.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `BatchUpdateVehicle` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Excluir um veículo AWS de IoT FleetWise

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 Gerencie frotas na IoT AWS FleetWise e Colete FleetWise dados de AWS IoT 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. Abra 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 DeleteVehicleda API para excluir um veículo. O exemplo a seguir usa AWS CLI.

Para excluir um veículo, execute o seguinte comando.

*`vehicle-name`*Substitua pela ID do veículo que você deseja excluir.

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

Verifique a exclusão do veículo

Você pode usar a operação ListVehiclesda 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListVehicles` API.

JSON

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

```
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Obtenha informações sobre AWS veículos de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação GetVehicleda API para recuperar informações do veículo. O exemplo a seguir usa a AWS CLI.

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

*vehicle-name*Substitua pela ID do veículo que você deseja 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.

Você pode usar a operação da GetVehicleStatusAPI para recuperar o status dos recursos associados a um veículo. O exemplo a seguir usa a AWS CLI.

Para recuperar o status dos recursos associados a um veículo, execute o comando a seguir.

- *vehicle-name*Substitua pela ID do veículo ao qual os recursos estão associados.
- *type*Substitua pelo tipo do recurso cujo status você deseja recuperar. Os valores válidos para `type` são `CAMPAIGN`, `STATE_TEMPLATE` e `DECODER`.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar as operações `GetVehicle` ou a `GetVehicleStatus` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Gerencie frotas na IoT AWS FleetWise

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 Colete FleetWise dados de AWS IoT com campanhas.

Uma frota contém as seguintes informações.

`fleetId`

O ID da frota.

(Optional) `description`

Uma descrição que ajuda você a encontrar a frota.

`signalCatalogArn`

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

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

- CreateFleet— Cria um grupo de veículos que contêm 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 IDs de todas as frotas às quais o veículo pertence.
- ListVehiclesInFleet— Recupera uma lista paginada de resumos de todos os veículos em uma frota.
- GetFleet— Recupera informações sobre uma frota.

Tópicos

- Crie uma frota AWS de IoT FleetWise
- Associe um FleetWise veículo de AWS IoT a uma frota
- Desassociar um veículo AWS de FleetWise IoT de uma frota
- Atualize uma frota AWS de IoT FleetWise
- Excluir uma frota AWS de IoT FleetWise
- Obtenha informações sobre AWS a frota de IoT FleetWise

Crie uma frota AWS de IoT FleetWise

Você pode usar a operação CreateFleetda API 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 Crie um catálogo de AWS sinais de IoT FleetWise.

Para criar uma frota, execute o seguinte comando.

- *`fleet-id`*Substitua pela 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 (:), hífens (-) e sublinhados (_).
- (Opcional) *`description`* Substitua por uma descrição.

  A descrição pode conter de 1 a 2.048 caracteres.
- *`signal-catalog-arn`*Substitua pelo ARN do catálogo de sinais.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateFleet` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Associe um FleetWise veículo de AWS IoT a uma frota

Você pode usar a operação AssociateVehicleFleetda API 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 Gerencie AWS veículos de IoT FleetWise.
> - Se você associar um veículo a uma frota que é alvo de uma campanha, a AWS IoT implanta FleetWise automaticamente a campanha no veículo.

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

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

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `AssociateVehicleFleet` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Desassociar um veículo AWS de FleetWise IoT de uma frota

Você pode usar a operação DisassociateVehicleFleetda API 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.

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

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `DisassociateVehicleFleet` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Atualize uma frota AWS de IoT FleetWise

Você pode usar a operação UpdateFleetda API para atualizar a descrição de uma frota. O exemplo a seguir usa AWS CLI.

Para atualizar uma frota, execute o comando a seguir.

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

  A descrição pode conter de 1 a 2.048 caracteres.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateFleet` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Excluir uma frota AWS de IoT FleetWise

Você pode usar a operação DeleteFleetda API 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 AWS de FleetWise IoT de uma frota.

Para excluir uma frota, execute o comando a seguir.

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

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

Verificar a exclusão da frota

Você pode usar a operação ListFleetsda API 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
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListFleets` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Obtenha informações sobre AWS a frota de IoT FleetWise

Você pode usar a operação da ListFleetsForVehicleAPI para recuperar uma lista paginada IDs de todas as frotas às quais o veículo pertence. O exemplo a seguir usa a AWS CLI.

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

*vehicle-name*Substitua pela ID do veículo.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListFleetsForVehicle` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Você pode usar a operação da ListVehiclesInFleetAPI 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.

*fleet-id*Substitua pelo ID da frota.

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

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `ListVehiclesInFleet` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Você pode usar a operação GetFleetda API 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.

*fleet-id*Substitua 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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `GetFleet` API.

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Colete FleetWise dados de AWS IoT com campanhas

Important

O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

Você cria as campanhas na nuvem. Depois que você ou sua equipe aprovarem uma campanha, a AWS IoT FleetWise define a campanha como pronta para ser implantada e ela será implantada no próximo check-in do veículo. 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.

Important

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. Abra o console AWS de IoT. FleetWise
  2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.
- Você se configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte Provisionar AWS veículos de IoT FleetWise.

Note

Você também pode Monitorar o último estado conhecido de seus veículos (não frotas) quase em tempo real usando modelos de estado que permitem transmitir dados de telemetria

com uma estratégia de atualização “On Change” ou “Periódica”. O recurso também fornece recursos “On Demand” para ativar ou desativar modelos implantados anteriormente ou solicitar o estado atual do veículo uma única vez (busca).
O acesso ao último estado conhecido está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Cada item da lista contém as seguintes informações.

`signalCatalogArn`

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

(Optional) `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. AWS FleetWise Atualmente, a IoT suporta o esquema de coleta baseado em condições e o esquema de coleta baseado em tempo.

- `conditionBasedCollectionScheme`— o esquema de coleta baseado em condições usa 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 coletar. 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 AWS de IoT FleetWise.
  - (Opcional) `conditionLanguageVersion` — a versão da linguagem de expressão condicional.

- (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) `triggerMode` — pode ser um dos seguintes valores:
  - `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.
- `timeBasedCollectionScheme`— ao definir um esquema de coleta baseado em tempo, especifique um período 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.
  - `periodMs`— o período de tempo (em milissegundos) para decidir com que frequência coletar dados.

(Optional) `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`

Escolha o único destino para o qual a campanha transferirá os dados do veículo. Você pode enviar os dados para um tópico do MQTT ou armazená-los no Amazon S3 ou no Amazon Timestream.

O MQTT (Message Queuing Telemetry Transport) é um protocolo de mensagens leve e amplamente adotado. Você pode publicar dados em um tópico do MQTT para criar suas próprias arquiteturas orientadas a eventos usando regras. AWS IoT AWS IoT o suporte para MQTT é baseado na especificação MQTT v3.1.1 e na especificação MQTT v5.0, com algumas diferenças. Para obter mais informações, consulte Diferenças do MQTT.

O S3 pode ser 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.

> **Note**
>
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

(Optional) `dataExtraDimensions`

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

(Optional) `dataPartitions`

Crie uma partição de dados para armazenar temporariamente os dados do sinal em um veículo. Você configura quando e como encaminhar os dados para a nuvem.

- Especifique como a AWS IoT FleetWise armazena os dados em um veículo ou frota definindo o tamanho máximo de armazenamento, o tempo mínimo de vida e o local de armazenamento.
- A campanha `spoolingMode` deve ser`TO_DISK`.
- As configurações de upload incluem a definição da versão da linguagem da condição e da expressão lógica.

(Optional) `description`

Adicione uma descrição para ajudar a identificar o objetivo da campanha.

(Optional) `diagnosticsMode`

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

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

(Optional) `expiryTime`

Defina 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)

(Optional) `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`

(Optional) `priority`

Especifique um número 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`

(Optional) `signalsToCollect`

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

- `name`— o nome do sinal a partir do qual os dados são coletados quando o esquema de coleta de dados é invocado.
- `dataPartitionId`— o ID da partição de dados a ser usada no sinal. O ID deve corresponder a um dos IDs fornecidos em`dataPartitions`. Se você carregar um sinal como condição em sua partição de dados, esses mesmos sinais deverão ser incluídos`signalsToCollect`.
- (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 invocado.
- (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

(Optional) `spoolingMode`

Se `spoolingMode` estiver configurado para `TO_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`

(Optional) `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 A 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`. Depois que a campanha é aprovada, a AWS IoT implanta FleetWise automaticamente a campanha no veículo ou frota alvo. Para obter mais informações, consulte Atualizar uma campanha AWS de IoT FleetWise.
- `RUNNING` — A campanha está ativa.
- `SUSPENDED` — A campanha está suspensa. Para retomar a campanha, use a operação da API `UpdateCampaign`.

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

- CreateCampaign— Cria uma nova 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

- Crie uma campanha AWS de IoT FleetWise
- Atualizar uma campanha AWS de IoT FleetWise
- Excluir uma campanha AWS de IoT FleetWise
- Obtenha informações AWS da campanha de IoT FleetWise
- Armazene e encaminhe dados da campanha
- Colete dados de códigos de diagnóstico de problemas usando a AWS IoT FleetWise
- Visualize dados de veículos AWS de FleetWise IoT

Crie uma campanha AWS de IoT FleetWise

Important

O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

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. Abra o console AWS de IoT. FleetWise
  2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore o Edge Agent.
- Você se configurou AWS IoT Core para provisionar seu veículo. Para obter mais informações, consulte Provisionar AWS veículos de IoT FleetWise.

Tópicos

- Criar uma campanha (console)
- Criar uma campanha (AWS CLI)
- Expressões lógicas para campanhas AWS de IoT FleetWise

Criar uma campanha (console)

Use o FleetWise console de AWS IoT para criar uma campanha para selecionar, coletar e transferir dados do veículo para a nuvem.

Como criar uma campanha

1. Abra o console AWS de 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: especificar as condições de armazenamento e upload
- Etapa 3: configurar o destino dos dados
- Etapa 4: adicionar veículos
- Etapa 5: analisar e criar
- Etapa 6: implantar uma campanha

> **Important**
>
> - Você deve ter um catálogo de sinalização e um veículo antes de criar uma campanha. Para obter mais informações, consulte Gerencie AWS catálogos de sinais de IoT FleetWise e Gerencie AWS veículos de IoT FleetWise.
> - Após a criação de uma campanha, você deverá aprová-la. Para obter mais informações, consulte Atualizar uma campanha AWS de IoT FleetWise.

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 FleetWise console de AWS IoT, 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 especificou um esquema de coleta de dados baseado em condições, deverá definir uma expressão lógica para reconhecer quais dados coletar. AWS A 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 `$variable.`myVehicle.InVehicleTemperature` > 50.0` expressão, a AWS IoT FleetWise coletará valores de temperatura maiores que 50,0. Consulte instruções de como escrever expressões em Expressões lógicas para campanhas AWS de IoT FleetWise.

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

4. (Opcional) Especifique a versão do idioma da expressão condicional. O valor padrão é 1.
5. (Opcional) Especifique 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 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.
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) Edite 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, especifique 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 A IoT define FleetWise 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 CreateCampaignda API.

     > **Note**
     >
     > AWS FleetWise Atualmente, a IoT suporta a extensão de formato de arquivo.json.

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

Etapa 2: especificar as condições de armazenamento e upload

Para escolher se o software Edge Agent armazenará temporariamente os dados localmente quando um veículo não estiver conectado à nuvem, especifique o modo de spooling.

- No modo de spooling de dados, escolha uma das seguintes opções:
  - Não armazenado — O software Edge Agent coleta, mas não armazena temporariamente dados localmente quando um veículo está off-line. O software Edge Agent transfere dados para a nuvem quando o veículo se reconecta.
  - Armazenado em disco — O software Edge Agent coleta e armazena temporariamente dados localmente quando um veículo está off-line. Os dados coletados são armazenados temporariamente em um local definido pela seção “persistência” do arquivo de configuração do Edge Agent. O Edge Agent transfere dados para a nuvem quando o veículo se reconecta.
  - Armazenado em disco com partições — O veículo sempre armazena temporariamente os dados no Edge na partição de dados especificada. Você pode escolher quando deseja encaminhar seus dados armazenados para a nuvem.

1. (Opcional) Insira uma ID de partição para designar um determinado conjunto de dados.
2. Insira um nome de pasta como o local onde os dados serão armazenados. O caminho absoluto do local de armazenamento é`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.
3. Insira o tamanho máximo de armazenamento dos dados armazenados na partição. Os dados mais recentes substituem os dados mais antigos quando a partição atinge o tamanho máximo.
4. Insira o tempo mínimo em que os dados dessa partição serão mantidos no disco.
5. (Opcional) Insira as condições de upload para a partição.

Especifique os sinais

Você pode especificar os sinais dos quais coletar dados durante a campanha.

Para especificar os sinais dos quais coletar dados

1. Selecione o nome do sinal.
2. (Opcional) Para Contagem máxima de amostras, insira o número máximo de amostras de dados que o software Edge Agent coleta e transfere para a nuvem durante a campanha.
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 Próximo.

Etapa 3: configurar o destino dos dados

> **Note**
>
> Se a campanha contiver sinais de dados do sistema de visão, você só poderá armazenar os dados do veículo no Amazon S3. Você não pode armazená-lo no Timestream nem enviá-lo para um tópico do MQTT.
> O recurso de dados de sistemas de visão está na versão de teste e está sujeito a alterações.
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

Escolha o destino para onde você deseja enviar ou armazenar os dados coletados pela campanha. Você pode enviar dados do veículo para um tópico do MQTT ou armazená-los no Amazon S3 ou no Amazon Timestream.

Em Configurações de destino, faça o seguinte:

- Escolha o tópico Amazon S3, Amazon Timestream ou MQTT na lista suspensa.

Amazon S3

> **Important**
>
> Você só pode transferir dados para o S3 se a AWS FleetWise IoT tiver permissões para gravar no bucket do S3. Para obter mais informações sobre como conceder acesso, consulte Como controlar o acesso com a AWS IoT FleetWise.

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 Como 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.

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.

Amazon Timestream

> **Important**
>
> Você só pode transferir dados para uma tabela se a AWS IoT FleetWise tiver permissões para gravar dados no Timestream. Para obter mais informações sobre como conceder acesso, consulte Como controlar o acesso com a AWS IoT FleetWise.
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

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.

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.

Tópico do MQTT

> **Important**
>
> Você só pode rotear dados para um tópico do MQTT se a AWS FleetWise IoT tiver permissões AWS IoT para os tópicos. Para obter mais informações sobre como conceder acesso, consulte Como controlar o acesso com a AWS IoT FleetWise.

Para enviar dados do veículo para um tópico do MQTT, escolha o tópico do MQTT.

Os dados do veículo enviados pelas mensagens MQTT são entregues quase em tempo real e permitem que você use regras para agir ou rotear dados para outros destinos. Para obter mais informações sobre o uso do MQTT, consulte Protocolos e regras de comunicação do dispositivo AWS IoT no Guia do AWS IoT Core desenvolvedor.

1. Em Tópico do MQTT, insira o nome do tópico.
2. No tópico Acesso ao serviço para MQTT, escolha se você deseja permitir que a AWS FleetWise IoT crie e use uma nova função de serviço para você. Se você quiser usar uma função de serviço existente, selecione a função na lista suspensa em Selecionar função.

- Escolha Próximo.

Etapa 4: 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 Selecionar tudo para adicionar todos os veículos à campanha.

5. Escolha Próximo.

Etapa 5: analisar 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 6: 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 Suspender. Para retomar a coleta de dados dos veículos conectados à campanha, selecione Continuar.

Criar uma campanha (AWS CLI)

Você pode usar a operação CreateCampaignda API para criar uma campanha. O exemplo a seguir usa a AWS CLI.

Quando você cria uma campanha, os dados coletados dos veículos podem ser enviados para um tópico do MQTT ou 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. Escolha o S3 para armazenamento de objetos com escalabilidade, disponibilidade de dados, segurança e desempenho líderes do setor. Escolha o MQTT para fornecer dados quase em tempo real e usar regras AWS IoT para realizar ações definidas por você ou rotear os dados para outros destinos.

> **Important**
>
> Você só pode transferir dados do veículo para um tópico do MQTT, Amazon S3 ou Amazon Timestream se a AWS FleetWise IoT tiver permissões para enviar mensagens MQTT em seu nome ou para gravar dados no S3 ou no Timestream. Para obter mais informações sobre como conceder acesso, consulte Como controlar o acesso com a AWS IoT FleetWise.
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

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 Gerencie AWS catálogos de sinais de IoT FleetWise, Gerencie AWS veículos de IoT FleetWise e Gerencie frotas na IoT AWS FleetWise.
> - 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 de IoT FleetWise.

Para criar uma campanha, execute o seguinte comando.

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

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

- *`campaign-name`*Substitua pelo nome da campanha que você está criando.

- *signal-catalog-arn*Substitua pelo Amazon Resource Name (ARN) do catálogo de sinais.
- *target-arn*Substitua pelo ARN de uma frota ou veículo que você criou.
- *bucket-arn*Substitua pelo ARN do bucket S3.

```
{
    "name": "campaign-name",
    "targetArn": "target-arn",
    "signalCatalogArn": "signal-catalog-arn",
    "collectionScheme": {
        "conditionBasedCollectionScheme": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
            "minimumTriggerIntervalMs": 1000,
            "triggerMode": "ALWAYS"
        }
    },
    "compression": "SNAPPY",
    "diagnosticsMode": "OFF",
    "postTriggerCollectionDuration": 1000,
    "priority": 0,
    "signalsToCollect": [
        {
         "maxSampleCount": 100,
         "minimumSamplingIntervalMs": 0,
         "name": "Vehicle.DemoEngineTorque"
        },
        {
         "maxSampleCount": 100,
         "minimumSamplingIntervalMs": 0,
         "name": "Vehicle.DemoBrakePedalPressure"
        }
    ],
    "spoolingMode": "TO_DISK",
    "dataDestinationConfigs": [
        {
         "s3Config": {
             "bucketArn": "bucket-arn",
             "dataFormat": "PARQUET",
             "prefix": "campaign-name",
              "storageCompressionFormat": "GZIP"
      }
    }
```

```
  ],
     "dataPartitions": [
      { ...  }
  ]
}
```

> **Note**
>
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

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

- *campaign-name*Substitua pelo nome da campanha que você está criando.
- *signal-catalog-arn*Substitua pelo Amazon Resource Name (ARN) do catálogo de sinais.
- *target-arn*Substitua pelo ARN de uma frota ou veículo que você criou.
- *topic-arn*Substitua pelo ARN do tópico MQTT que você especificou como destino para mensagens contendo dados do veículo.
- *role-arn*Substitua pelo ARN da função de execução de tarefas que concede FleetWise permissão à AWS IoT para enviar, receber e agir em mensagens do tópico MQTT que você especificou.

```
{
  "name": "campaign-name",
  "targetArn": "target-arn",
  "signalCatalogArn": "signal-catalog-arn",
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      "expression": "$variable.`Vehicle.DemoBrakePedalPressure` > 7000",
      "minimumTriggerIntervalMs": 1000,
```

```
      "triggerMode": "ALWAYS"
    }
  },
  "compression": "SNAPPY",
  "diagnosticsMode": "OFF",
  "postTriggerCollectionDuration": 1000,
  "priority": 0,
  "signalsToCollect": [
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoEngineTorque"
    },
    {
      "maxSampleCount": 100,
      "minimumSamplingIntervalMs": 0,
      "name": "Vehicle.DemoBrakePedalPressure"
    }
  ],
  "spoolingMode": "TO_DISK",
  "dataDestinationConfigs": [
      {
          "mqttTopicConfig": {
              "mqttTopicArn": "topic-arn",
              "executionRoleArn": "role-arn"
          }
      }
  ]
}
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `CreateCampaign` API.

JSON

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

```
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Expressões lógicas para campanhas AWS de IoT FleetWise

AWS A IoT FleetWise usa uma expressão lógica para reconhecer quais dados coletar 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 de IoT FleetWise

Você pode usar a operação UpdateCampaignda API para atualizar uma campanha existente. O comando a seguir usa AWS CLI.

- *campaign-name*Substitua pelo nome da campanha que você está atualizando.
- Substitua *action* por um dos seguintes:
  - APPROVE— Aprova a campanha para permitir que a AWS FleetWise IoT a 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 está programada para ser redistribuída para todos os veículos no próximo check-in e os veículos retomarão o envio de dados.

  - UPDATE— Atualiza a campanha definindo atributos e associando-os à campanha.
- *description*Substitua por uma nova descrição.

  A descrição pode ter até 2.048 caracteres.
- *data-extra-dimensions*Substitua por atributos específicos do veículo para enriquecer os dados coletados durante a campanha. Por exemplo, você pode adicionar a marca e o modelo do veículo à campanha, e a AWS IoT FleetWise associará os dados a esses atributos como dimensões no Amazon Timestream. Em seguida, você pode consultar os dados em relação à marca e ao modelo do veículo.

```
aws iotfleetwise update-campaign \
            --name campaign-name \
            --action action \
            --description description \
            --data-extra-dimensions data-extra-dimensions
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `UpdateCampaign` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey*",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Excluir uma campanha AWS de IoT FleetWise

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

Excluir uma campanha (console)

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

Excluir uma campanha

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Campanhas.
3. Na página Campanhas, selecione a campanha alvo.
4. Escolha 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 DeleteCampaignda API para excluir uma campanha. O exemplo a seguir usa AWS CLI.

Para excluir uma campanha, execute o seguinte comando.

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

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

> As partições de dados excluídas não são recuperáveis
>
> A exclusão de uma campanha remove todos os dados dos dispositivos e os dados em uma partição não serão enviados para a nuvem.

Verificar a exclusão da campanha

Você pode usar a operação ListCampaignsda API 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
```

Obtenha informações AWS da campanha de IoT FleetWise

Você pode usar a operação da GetCampaignAPI 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.

*campaign-name*Substitua 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.

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar a operação da `GetCampaign` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
```

```
    ]
}
```

Armazene e encaminhe dados da campanha

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Use partições de dados em campanhas para armazenar temporariamente dados de sinal no Edge para veículos e frotas. Ao configurar as opções de upload e armazenamento para partições de dados, você pode otimizar suas condições ideais para o encaminhamento de dados para seus destinos de dados designados (como um bucket do Amazon S3). Por exemplo, você pode configurar a partição de dados para armazenar dados em um veículo até que ele se conecte ao Wi-Fi. Então, quando o veículo se conecta, a campanha aciona os dados dessa partição específica para serem enviados para a nuvem. Como alternativa, você pode coletar dados usando AWS IoT Jobs.

Tópicos

- Crie partições de dados
- Carregar dados da campanha
- Carregar dados usando AWS IoT Jobs

Crie partições de dados

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Uma partição de dados em uma campanha armazena temporariamente os dados do sinal. Você configura quando e como encaminhar os dados para a nuvem.

Uma partição de dados funciona designando primeiro um conjunto específico de dados usando o `dataPartitionId` para uma campanha. Em seguida, você pode definir ainda mais as opções de armazenamento de partições, como tamanho máximo, tempo mínimo para manter a partição de dados ativa (no disco) e onde armazenar os dados no Edge. Você pode determinar o local de armazenamento no veículo usando`storageLocation`. O local de armazenamento determina o nome da pasta para a partição de dados na pasta de armazenamento da campanha. A pasta de armazenamento da campanha está em uma pasta com o nome do veículo em um caminho de persistência definido no arquivo de configuração do Edge. Este é o caminho absoluto do local de armazenamento:`{persistency_path} / {vehicle_name} / {campaign_name} / {storage_location}`.

O modo de spooling definido para `TO_DISK` especifica que os dados particionados devem ser salvos em um disco no veículo. O armazenamento de dados para partições de dados opera com base no FIFO (primeiro a entrar, primeiro a sair). Se você excluir uma campanha, também excluirá os dados na partição de dados associada. Se você não especificar uma partição de dados para casos de on/off uso de conectividade, a AWS IoT FleetWise ainda armazenará dados em um buffer circular no veículo quando não houver conectividade. Quando a conectividade é retomada, AWS a FleetWise IoT carrega os dados para a nuvem. Esse comportamento é configurável no software Edge Agent for AWS FleetWise IoT.

> **Important**
>
> Se sua partição de dados exceder o limite máximo de armazenamento definido, os dados mais recentes substituirão os dados mais antigos quando a partição atingir o tamanho máximo. Os dados perdidos no Edge não são recuperáveis. O tamanho do armazenamento é determinado pelo limite de armazenamento do Edge.
> Quando os dados são enviados para a nuvem, eles podem ser removidos após o tempo mínimo de vida. Defina o tempo mínimo de vida adequado para evitar a exclusão não intencional.

As opções de upload determinam as expressões variáveis e o idioma da condição. Se as opções de upload forem especificadas, você também deverá especificar as opções de armazenamento. Você também pode solicitar que os sinais nas partições de dados sejam enviados para a nuvem. Para obter mais informações, consulte Carregar dados da campanha.

Depois que as condições da partição de dados são definidas, `signalsToCollect` ajuda a especificar quais sinais devem ser considerados na partição de dados. Você pode especificar IDs

para partições de dados ou definir o `dataPartitionId` `default` para usar uma partição de dados padrão estabelecida. Um sinal sem um especificado `dataPartitionId` será associado ao padrão`dataPartition`.

Para criar uma partição de dados

Usando o exemplo a seguir, crie uma campanha com uma condição de armazenamento de partição de dados. Este exemplo de campanha está configurado para armazenar dados de veículos no Amazon Timestream.

1. *`campaign-name`*Substitua pelo nome da campanha que você está criando.
2. (Opcional) Forneça uma descrição.
3. *`role-arn`*Substitua pelo Amazon Resource Name (ARN) da função de execução de tarefas que concede FleetWise permissão à AWS IoT para entregar dados à tabela Timestream.
4. *`table-arn`*Substitua pelo ARN da tabela Timestream.
5. *`signal-catalog-arn`*Substitua pelo ARN do catálogo de sinais.
6. Substitua *`data-partition-id`* tanto pela `dataPartitions` ID quanto pela ID à qual se associar`signalsToCollect`. Primeiro, substitua o ID da partição de dados a ser usada no sinal. Pois`signalsToCollect`, o ID deve corresponder a um dos IDs fornecidos em`dataPartitions`.

   > **Note**
   >
   > Estabeleça uma partição de dados padrão para uma campanha usando `default` como ID.

7. *`target-arn`*Substitua pelo ARN de uma frota ou veículo que você criou.

```
{
    "name": "campaign-name",
    "description": "Measurement of SOC, SOH, thermal, and power optimization for Fleet
 2704",
    "targetArn": "target-arn",
    "collectionScheme": {
        "conditionBasedCollectionScheme": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS` > 50",
            "minimumTriggerIntervalMs": 1000,
```

```
            "triggerMode": "ALWAYS"
        }
    },
    "compression": "SNAPPY",
    "dataDestinationConfigs": [{
        "timestreamConfig": {
            "executionRoleArn": "role-arn",
            "timestreamTableArn": "table-arn"
        }
    }],
    "dataPartitions": [{
        "id": "data-partition-id",
        "storageOptions": {
            "maximumSize": {
                "unit": "GB",
                "value": 1024
            },
            "minimumTimeToLive": {
                "unit": "WEEKS",
                "value": 6
            },
            "storageLocation": "string"
        },
        "uploadOptions": {
            "conditionLanguageVersion": 1,
            "expression": "$variable.`Vehicle.BMS.PowerOptimization` > 90"
        }
    }],
    "signalCatalogArn": "signal-catalog-arn",
    "signalsToCollect": [{
        "dataPartitionId": "data-partition-id",
        "maxSampleCount": 50000,
        "minimumSamplingIntervalMs": 100,
        "name": "Below-90-percent"
    }],
    "spoolingMode": "TO_DISK",
    "tags": [{
        "Key": "BMS",
        "Value": "Under-90"
    }]
}
```

Depois de atender a todas as condições especificadas, os dados particionados são encaminhados para a nuvem, permitindo a coleta e o armazenamento de novos sinais particionados.

Em seguida, você chamará a `UpdateCampaign` API para implantá-la no software Edge Agent for AWS IoT FleetWise . Para obter mais informações, consulte Carregar dados da campanha.

Carregar dados da campanha

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Há duas maneiras de fazer upload de dados da campanha no Edge:

- As campanhas que atenderem às suas condições de upload enviarão automaticamente os dados para a nuvem após serem aprovadas. Para aprovar uma campanha, use a operação da `updateCampaign` API.
- Por meio de AWS IoT Jobs, você pode forçar o upload de dados mesmo quando as condições especificadas não forem atendidas. Para obter mais informações, consulte Carregar dados usando AWS IoT Jobs.

Para fazer upload dos dados da campanha usando a operação **`UpdateCampaign`** da API

Depois de criar a campanha, o status da campanha é exibido `WAITING_FOR_APPROVAL` até você alterar o `action` para`APPROVED`.

- Use o exemplo a seguir para atualizar a campanha `action` chamando a operação UpdateCampaignda API.

```
{
   "action": "APPROVED",
   "dataExtraDimensions": [ "string" ],
   "description": "string",
   "name": "string"
}
```

Carregar dados usando AWS IoT Jobs

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Com o AWS IoT Jobs, você pode configurar campanhas para carregar dados de veículos armazenados na nuvem sempre que precisar.

Para criar um documento de trabalho para sua campanha

- Use o exemplo a seguir para criar um documento de trabalho para a campanha. Um documento de trabalho é um arquivo.json que contém informações sobre veículos ou frotas necessários para realizar um trabalho. Para obter mais informações sobre a criação de documentos de trabalho, consulte Criar e gerenciar trabalhos usando o AWS CLI Guia do AWS IoT desenvolvedor.

  Para solicitar que apenas um veículo carregue dados, defina a meta do trabalho para o AWS IoT que está associado ao veículo. Para solicitar que vários veículos (na mesma campanha) enviem dados, crie um grupo de coisas com todas as coisas correspondentes aos veículos e, em seguida, defina a meta de trabalho para o grupo de coisas.

  ```
  {
    "version": "1.0",
    "parameters": {
       "campaignArn": ${aws:iot:parameter:campaignArn},
       "endTime": ${aws:iot:parameter:endTime}
    }
  }
  ```

  a. `CampaignArn`Substitua pelo nome de recurso da Amazon (ARN) de uma campanha na mesma região e conta. O ARN da campanha é obrigatório.

  b. (Opcional) `endTime` Substitua pelo timestamp dos dados coletados no veículo no formato ISO 8601 UTC (sem milissegundos). Por exemplo, `.2024-03-05T23:00:00Z` O carimbo de data/hora é exclusivo e determina o último ponto de dados a ser carregado. Se você

omitir`endTime`, o software Edge Agent continuará sendo carregado até que todos os dados armazenados de uma campanha sejam carregados. Depois que todos os dados são carregados, ele atualiza o status de execução do trabalho para`SUCCEEDED`. O estado do trabalho é atualizado para`COMPLETED`.

Para criar um trabalho usando um modelo de trabalho gerenciado

1. Escolha IoT-io TFleet Wise- na lista CollectCampaignData de modelos gerenciados. Para obter mais informações, consulte Criar um trabalho a partir de modelos AWS gerenciados no Guia do AWS IoT desenvolvedor.
2. O modelo gerenciado tem `CampaignArn` os `endTime` parâmetros e.

   a. `CampaignArn`Substitua pelo nome de recurso da Amazon (ARN) de uma campanha na mesma região e conta. O ARN da campanha é obrigatório.

   b. (Opcional) `endTime` Substitua pelo timestamp dos dados coletados no veículo no formato ISO 8601 UTC (sem milissegundos). Por exemplo, `.2024-03-05T23:00:00Z` O carimbo de data/hora é exclusivo e determina o último ponto de dados a ser carregado. Se você omitir`endTime`, o software Edge Agent continuará sendo carregado até que todos os dados armazenados de uma campanha sejam carregados. Depois que todos os dados são carregados, ele atualiza o status de execução do trabalho para`SUCCEEDED`. O estado do trabalho é atualizado para`COMPLETED`.

Para tópicos relacionados à solução de problemas, consulteArmazenar e encaminhar problemas.

Para obter mais informações sobre AWS IoT trabalhos, consulte Trabalhos no Guia do AWS IoT desenvolvedor.

Colete dados de códigos de diagnóstico de problemas usando a AWS IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Quando um veículo detecta um erro, ele gera um código de problema de diagnóstico (DTC) e registra um instantâneo dos sensores ou atuadores afetados. DTCs ajudam você a aprender sobre erros quase em tempo real, entender o que os está causando e tomar ações corretivas. AWS A IoT FleetWise suporta a coleta de DTCs, incluindo instantâneos de DTC correspondentes e dados estendidos por meio de uma campanha de coleta de dados. Este tópico apresenta os conceitos, fluxos de trabalho e palavras-chave que facilitam a coleta de dados do DTC, ilustrados com exemplos.

A seguir, mostramos os principais conceitos para usar o DTC.

Funções personalizadas definidas

Uma função definida de forma personalizada é a capacidade de invocar e executar suas próprias funções predefinidas no Edge Agent, ampliando o conceito de decodificação personalizada. Essas funções são usadas em coordenação com o AWS IoT FleetWise Agent. O FleetWise software Edge Agent para AWS IoT fornece funções integradas para calcular estatísticas de sinal, como valores mínimos, máximos e médios. Uma função personalizada amplia esse recurso, permitindo que você crie uma lógica personalizada para casos de uso específicos. Para a coleta de dados de código de problema de diagnóstico (DTC), os desenvolvedores podem aproveitar funções personalizadas para implementar mecanismos avançados de recuperação de dados, como obter códigos DTC, instantâneos e dados estendidos diretamente do Edge do veículo por meio de Serviços de Diagnóstico Unificados (UDS) ou interfaces alternativas de diagnóstico.

Para obter mais informações, consulte o guia de funções personalizadas e a implementação da referência de coleta de dados do DTC no Guia do Desenvolvedor do Edge Agent.

Busca de sinal

Em campanhas de coleta de dados, os sinais geralmente são coletados continuamente de um dispositivo e armazenados em buffer no software Edge Agent. Os sinais são então enviados ou armazenados periodicamente em campanhas baseadas em tempo ou acionados por condições específicas em campanhas baseadas em condições. No entanto, devido a preocupações com o congestionamento do tráfego do dispositivo, os sinais de DTC não podem ser coletados dos dispositivos e armazenados em buffer continuamente. Para resolver isso, a AWS IoT FleetWise fornece busca de sinal, o que garante que o sinal alvo seja obtido de forma descontínua de um dispositivo.

A busca de sinais suporta ações periódicas e orientadas por condições. Você pode definir o método, as condições e as ações exatas orientadas pela busca usando funções personalizadas

para cada sinal que não deve ser coletado continuamente de um dispositivo. Para sinais gerenciados pelo mecanismo de busca de sinal, o tipo de gatilho e as condições para armazenamento local ou upload na nuvem ainda são regidos pelo`CollectionScheme`, ambos `timeBasedCollectionScheme` e `conditionBasedCollectionScheme` são suportados, o que é igual aos sinais regulares.

Os tópicos a seguir mostram como você pode criar e usar DTCs.

Tópicos

- Palavras-chave do código de problema de diagnóstico
- Crie uma campanha de coleta de dados para códigos de problemas de diagnóstico
- Casos de uso do código de problema de diagnóstico

Palavras-chave do código de problema de diagnóstico

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

`signalsToFetch`parâmetro para criar campanha

Use a signalsToFetchsintaxe para configurar como as informações do sinal podem ser obtidas no Edge. A busca de sinal padrão é controlada pela modelagem conforme regras definidas explicitamente em um manifesto do decodificador ou definidas de forma personalizada por meio do Edge First Modeling. Com sinais para buscar, você pode definir quando e como os dados são buscados durante as campanhas.

Os sinais a serem buscados permitem a coleta de informações do DTC. Por exemplo, você pode criar um sinal do tipo string chamado `DTC_Info` que pode conter informações de DTC para cada unidade de controle do motor (ECU). Ou você pode filtrar por uma ECU específica.

- `SignalFetchInformation`definições de estrutura e parâmetros.

```
structure SignalFetchInformation {
```

```
    @required
    fullyQualifiedName: NodePath,
    @required
    signalFetchConfig: SignalFetchConfig,
    // Conditional language version for this config
    conditionLanguageVersion: languageVersion,
    @required
    actions: EventExpressionList,
}
```

- `fullyQualifiedName`: o nome totalmente qualificado (FQDN) do sinal para o qual você deseja usar a busca personalizada.
- `signalFetchConfig`: define regras sobre como os sinais definidos acima devem ser buscados. Ele suporta a busca baseada em tempo e em condições.
- `conditionLanguageVersion`: a versão de linguagem condicional usada para analisar a expressão na configuração.
- `actions`: uma lista de todas as expressões de ação avaliadas no Edge. O Edge obterá o valor do sinal definido.

> **Important**
>
> As ações só podem ser usadas`custom_function`.

Palavras-chave de expressão de campanha

A expressão a seguir usa o nome totalmente qualificado de um sinal suportado pelo veículo e retorna verdadeiro se o sinal não tiver dados nos buffers de sinal no Edge. Por outro lado, ele retorna falso.

```
isNull(signalFqdn:String): Boolean
```

Example uso

```
isNull($variable.`Vehicle.ECU1.DTC_INFO`) == false

We want to make sure DTC_Info signal is being generated
on edge.
```

Essa expressão usa a seguinte entrada:

Nome da função: string

O nome da função personalizada que é suportada pelo Edge

parâmetros: vararga ***Expression***

Parâmetros para`functionName`. Isso pode ser qualquer lista de expressões.

Os parâmetros suportam o tipo literal: String, Int, Boolean ou Double.

```
custom_function(functionName:String, params: varargsExpression): Void
```

Example uso

```
{
       "fullyQualifiedName":"Vehicle.ECU1.DTC_INFO",
       "signalFetchConfig":{
          "timeBased":{
             "executionFrequencyMs":2000
          }
       },
       "actions":"custom_function(“DTC_QUERY”, -1, 2, -1)"
    }
```

Crie uma campanha de coleta de dados para códigos de problemas de diagnóstico

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Este tópico descreve como criar uma campanha de coleta de dados para códigos de problemas de diagnóstico (DTC).

1. Defina um sinal personalizado no Edge. Você precisa definir as regras de decodificação para o sinal DTC no Edge como um sinal decodificado personalizado. Para obter mais informações,

consulte Tutorial: Configurar a coleta de dados independente da rede usando uma interface de decodificação personalizada.

2. Defina a função personalizada no Edge. Você precisa definir uma função personalizada para coletar sinais DTC no Edge em um tempo compilado.

   Para obter mais informações, consulte o guia de funções personalizadas e a implementação da referência de coleta de dados do DTC no Guia do Desenvolvedor do Edge Agent.

   > **Note**
   >
   > Um exemplo de função personalizada `DTC_QUERY` definida é mostrado no script de demonstração.

3. Crie um catálogo de sinais que modele um sinal DTC como um tipo de string.

```
[
 {
    "branch": {
        "fullyQualifiedName": "Vehicle",
        "description": "Vehicle"
        }
      },
      {
    "branch": {
        "fullyQualifiedName": "Vehicle.ECU1",
        "description": "Vehicle.ECU1"
        }
      },
      {
    "sensor": {
        "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
        "description": "Vehicle.ECU1.DTC_INFO",
        "dataType": "STRING"
      }
   }
 ]
```

4. Crie e ative um modelo de veículo com o sinal DTC adicionado.
5. Crie e ative um manifesto do decodificador com o sinal DTC adicionado. O sinal DTC deve ser do tipo decodificador de `CUSTOM_DECODING_SIGNAL` sinal com um tipo de interface de `CUSTOM_DECODING_INTERFACE` rede.

Example decodificador de sinal

```
[
  {
    "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
    "interfaceId": "UDS_DTC",
    "type": "CUSTOM_DECODING_SIGNAL",
    "customDecodingSignal": {
      "id": "Vehicle.ECU1.DTC_INFO"
    }
  }
 ]
```

Example interface da rede

```
[
  {
    "interfaceId": "UDS_DTC",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
```

> **Note**
>
> Os sinais da Controller Area Network (CAN) não suportam o tipo de dados de string.

6. Provisionar e criar veículos. Os veículos devem utilizar um modelo de veículo (manifesto do modelo) e um manifesto do decodificador que foram ativados nas etapas anteriores.
7. Crie e aprove a campanha. Você precisa criar uma campanha definindo sinais de DTC (opcionalmente com sinais de telemetria) e implantá-la em veículos.
8. Acesse os dados no destino definido. Os dados do DTC incluem`DTCCode`,`DTCSnapshot`, e `DTCExtendedDatastrings` como uma string bruta no destino de dados definido na campanha.

Casos de uso do código de problema de diagnóstico

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Os casos de uso a seguir presumem que a `DTC_QUERY` função foi definida no script de demonstração.

Busca periódica

Obtenha uma coleção de DTC em intervalos configurados.

O exemplo a seguir é uma campanha com busca periódica de sinais `Vehicle.DTC_INFO` para todos DTCs com uma máscara de status para todos. ECUs Há uma condição para que os dados sejam coletados`Vehicle.DTC_INFO`.

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "timeBased": {
        // The FleetWise Edge Agent will query the UDS module for all DTCs every five
 seconds.
          "executionFrequencyMs": 5000
        }
      },
      "actions": [
      // Every five seconds, this action is called and its output is stored in the
      // signal history buffer of Vehicle.DTC_INFO
        "custom_function(\"DTC_QUERY\", -1, 2, -1)"
      ]
    }
  ],
  "signalsToCollect": [
    {
```

```
      "name": "Vehicle.ECU1.DTC_INFO"
    }
  ],
  "collectionScheme": {
    "conditionBasedCollectionScheme": {
      "conditionLanguageVersion": 1,
      // Whenever a new DTC is filled into the signal, the data is ingested.
      "expression": "!isNull($variable.`Vehicle.ECU1.DTC_INFO`)",
      "minimumTriggerIntervalMs": 1000,
      // Make sure that data is ingested only when there are new DTCs.
      "triggerMode": "RISING_EDGE"
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Busca baseada em condições

Busque uma coleção de DTC quando uma condição for atendida. Por exemplo, quando o sinal CAN estiver`Vehicle.Ignition == 1`, busque e carregue os dados do DTC.

O exemplo de campanha a seguir tem a busca de sinal orientada por condições `Vehicle.ECU1.DTC_INFO` para verificar se o DTC (” AAA123 “) está pendente com o recorde número 1 do ECU-1. Esta campanha tem coleta e upload de dados com base no tempo.

```
{
  "compression": "SNAPPY",
  "spoolingMode": "TO_DISK",
  "signalsToFetch": [
    {
      "fullyQualifiedName": "Vehicle.ECU1.DTC_INFO",
      "signalFetchConfig": {
        "conditionBased": {
```

```
        // The action will only run when the ignition is on.
          "conditionExpression": "$variable.`Vehicle.Ignition` == 1",
          "triggerMode": "ALWAYS"
        }
      },
      // The UDS module is only requested for the specific ECU address and the specific
 DTC Number/Status.
      "actions": ["custom_function(\"DTC_QUERY\", 1, 2, 8, \"0xAAA123\")"]
    }
  ],
  "signalsToCollect": [
    {
      "name": "Vehicle.ECU1.DTC_INFO"
    },
    {
      "name": "Vehicle.Ignition"
    }
  ],
  "collectionScheme": {
    "timeBasedCollectionScheme": {
      "periodMs": 10000
    }
  },
  "dataDestinationConfigs": [
    {
      "s3Config":
        {
          "bucketArn": "bucket-arn",
          "dataFormat": "PARQUET",
          "prefix": "campaign-name",
          "storageCompressionFormat": "GZIP"
        }
    }
  ]
}
```

Busca sob demanda

Obtenha um DTC específico para uma frota.

Para um caso de uso sob demanda, você pode usar a mesma campanha definida na busca periódica. O efeito sob demanda é obtido suspendendo a campanha logo após a implantação da campanha usando o FleetWise console de AWS IoT ou executando o seguinte comando da CLI.

- *command-name*Substitua pelo nome do comando.

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

Em seguida, suspenda a campanha após a chegada dos dados do DTC.

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

Você pode retomar a campanha novamente para a busca de dados do DTC.

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

Visualize dados de veículos AWS de FleetWise IoT

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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

> **Note**
>
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

Processando dados do veículo enviados para um tópico do MQTT

Os dados do veículo enviados pelas mensagens MQTT são entregues quase em tempo real e permitem que você use as Regras para agir ou rotear dados para outros destinos. Para obter mais informações sobre o uso do MQTT, consulte Protocolos e regras de comunicação do dispositivo AWS IoT no Guia do AWS IoT Core desenvolvedor.

O esquema padrão de dados que é enviado em uma mensagem MQTT contém os seguintes campos.

Nome do campo	Tipo de dados	Description
`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.
`measure_value::double`	double	Valores de sinal do tipo Duplo.
`measure_value::boolean`	booleano	Valores de sinal do tipo Booleano.

Nome do campo	Tipo de dados	Description
`measure_value::varchar`	varchar	Valores de sinal do tipo varchar.

Processar 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	Description
`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	Description	
`measure_value::double`	double	Valores de sinal do tipo Duplo.	
`measure_value::boolean`	booleano	Valores de sinal do tipo Booleano.	
`measure_value::varchar`	varchar	Valores de sinal do tipo varchar.	

Visualize os 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 várias AWS fontes (como Amazon CloudWatch e Timestream) e outras fontes de dados com um único painel da Grafana.
- Analise e visualize dados em painéis usando o Quick.

Processar dados do veículo no Amazon S3

Amazon S3 é um serviço de armazenamento de objetos que armazena e protege qualquer quantidade de dados. Você pode usar o S3 para diversos casos de uso, como lagos de dados, backup e restauração, arquivamento, aplicativos corporativos, AWS IoT dispositivos 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	Description	
`eventId`	varchar	O ID do evento de coleta de dados.	

Nome do campo	Tipo de dados	Description
`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.
`measure_value_DOUBLE`	double	Valores de sinal do tipo Duplo.
`measure_value_BOOLEAN`	booleano	Valores de sinal do tipo Booleano.
`measure_value_STRUCT`	struct	Valores de sinal do tipo Struct.
`measure_value_VARCHAR`	varchar	Valores de sinal do tipo varchar.

Formato de objeto Amazon S3

AWS A 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. Ele pode estar em vários formatos, como imagens ou vídeos.

As mensagens do veículo passadas para a AWS IoT FleetWise com dados de sinal dos arquivos 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, a 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
```

Analise os dados do veículo armazenados no Amazon 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 AI para fluxos de trabalho downstream de rotulagem e aprendizado de máquina (ML).

Para obter mais informações, consulte os seguintes tópicos no Amazon SageMaker AI Developer Guide:

- 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
- Usando 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 Quick lendo diretamente a tabela do Athena ou o bucket do S3.

> **Tip**
>
> Se você estiver lendo diretamente do S3, confirme se os dados do seu veículo estão no formato JSON porque o Quick não é compatível com o formato Apache Parquet.

Para obter mais informações, consulte os seguintes tópicos no Amazon Quick User Guide:

- Fonte de dados compatíveis

- Criação de fonte de dados

Comandos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.
> Esta documentação descreve como usar o recurso de comandos para AWS IoT FleetWise. Para obter informações sobre como usar o recurso de comandos em AWS IoT Device Management, consulte comandos.
> Você é o único responsável por implantar comandos de forma segura e compatível com as leis aplicáveis. Para ter mais informações sobre suas responsabilidades, consulte os Termos de Serviço da AWS para Serviços da AWS IoT.

Use o recurso de comandos para executar comandos em um veículo a partir da nuvem. Os comandos têm como alvo um dispositivo por vez e podem ser usados para aplicações de baixa latência e alto throughput, como recuperar os logs do lado do dispositivo ou iniciar uma alteração no estado do dispositivo.

O comando é um recurso gerenciado pelo AWS IoT Device Management. Ele contém configurações reutilizáveis que são aplicadas ao enviar uma execução de comando para o veículo. Você pode predefinir um conjunto de comandos para casos de uso específicos ou usá-los para criar configurações reutilizáveis para casos de uso recorrentes. Por exemplo, você pode configurar comandos que podem ser usados por um aplicativo para trancar a porta de um veículo ou alterar a temperatura remotamente.

Usando o recurso de AWS IoT comandos, você pode:

- Crie um recurso de comando e reutilize a configuração para enviar vários comandos ao dispositivo de destino e, em seguida, executá-los no dispositivo.
- Controle a granularidade com a qual você deseja que cada comando seja executado no dispositivo. Por exemplo, você pode provisionar um veículo como uma AWS IoT coisa e, em seguida, enviar um comando para trancar ou destrancar as portas do veículo.
- Execute vários comandos simultaneamente no dispositivo de destino sem esperar que o anterior seja concluído.

- Escolha ativar notificações para eventos de comandos e recuperar as informações de status e resultado do dispositivo à medida que ele executa o comando e depois de concluído.

Os tópicos a seguir mostram como criar, enviar, receber e gerenciar comandos.

Tópicos

- Conceitos de comandos
- Veículos e comandos
- Criar e gerenciar comandos
- Iniciar e monitorar execuções de comando
- Exemplo: Usando comandos para controlar o modo de direção de um veículo (AWS CLI)
- Cenários de uso de comandos

Conceitos de comandos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Comandos são instruções enviadas da nuvem para o dispositivo de destino. O dispositivo de destino pode ser um veículo e deve ser registrado como uma AWS IoT coisa no registro de coisas. O comando pode conter parâmetros que definem uma ação que os atuadores do veículo precisam realizar. O veículo então analisa o comando e seus parâmetros e os processa para realizar a ação correspondente. Em seguida, ele responde ao aplicativo em nuvem com o status da execução do comando.

Para obter o fluxo de trabalho detalhado, consulteVeículos e comandos.

Tópicos

- Conceitos de chaves de comandos
- Status de execução do comando

Conceitos de chaves de comandos

Veja a seguir alguns conceitos-chave para usar o recurso de comandos e como ele funciona com os modelos de estado do último estado conhecido (LKS).

Comando

Um Comando é uma entidade que você pode usar para enviar instruções a um veículo físico para que ele execute ações como ligar o motor ou alterar a posição das janelas. Você pode predefinir um conjunto de comandos para casos de uso específicos ou usá-los para criar configurações reutilizáveis para casos de uso recorrentes. Por exemplo, você pode configurar comandos que podem ser usados por um aplicativo para trancar a porta de um veículo ou alterar a temperatura remotamente.

Namespace

Ao usar o recurso de comandos, você deve especificar o namespace para o comando. Ao criar um comando na AWS IoT FleetWise, você deve escolher `AWS-IoT-FleetWise` como seu namespace. Ao usar esse namespace, você deve fornecer os parâmetros que serão usados para executar o comando no veículo. Se você quiser criar um comando em AWS IoT Device Management vez disso, você deve usar o `AWS-IoT` namespace em vez disso. Para obter mais informações, consulte os comandos no guia do AWS IoT Device Management desenvolvedor.

Estados de comando

Os comandos que você criar estarão em um estado disponível, o que significa que podem ser usados para iniciar a execução de um comando no veículo. Se um comando ficar desatualizado, você poderá descontinuar o comando. Para um comando no estado obsoleto, as execuções de comandos existentes serão executadas até a conclusão. Você não pode atualizar o comando nem executar novas execuções. Para enviar novas execuções, você deve restaurar o comando para que ele fique disponível.

Você também pode excluir um comando se ele não for mais necessário. Quando você marca um comando para exclusão, se o comando tiver sido descontinuado por um período maior que o tempo limite máximo de 24 horas, o comando será excluído imediatamente. Se o comando não for suspenso ou permanecer suspenso por um período menor que o valor do tempo limite máximo, ele entrará no estado de exclusão pendente. O comando será removido automaticamente da sua conta após 24 horas.

Parâmetros

Ao criar um comando, você pode especificar opcionalmente os parâmetros que deseja que o veículo de destino execute ao executar o comando. O comando que você cria é uma configuração reutilizável e pode ser usado para enviar várias execuções de comando ao seu veículo e executá-las simultaneamente. Como alternativa, você também pode especificar os parâmetros somente em tempo de execução e optar por realizar uma operação única de criar um comando e enviá-lo ao seu veículo.

Veículo alvo

Quando quiser executar o comando, você deve especificar um veículo alvo que receberá o comando e executará ações específicas. O veículo-alvo já deve ter sido registrado como algo com AWS IoT. Depois de enviar o comando para o veículo, ele começará a executar uma instância do comando com base nos parâmetros e nos valores que você especificou.

Atuadores

Quando quiser executar o comando, você deve especificar os atuadores no veículo que receberão o comando e seus valores que determinam as ações a serem executadas. Opcionalmente, você pode configurar valores padrão para os atuadores para evitar o envio de comandos imprecisos. Por exemplo, você pode usar um valor padrão de `LockDoor` para um atuador de fechadura de porta para que o comando não destranque acidentalmente as portas. Para obter informações gerais sobre atuadores, consultePrincipais conceitos.

Suporte ao tipo de dados

Os seguintes tipos de dados são compatíveis com os atuadores usados para o recurso de comandos.

> **Note**
>
> As matrizes não são suportadas para dados telemáticos, comandos ou último estado conhecido (LKS). Você só pode usar o tipo de dados de matriz para dados de sistemas de visão.

- Tipos de ponto flutuante. Os tipos a seguir são aceitos.
  - Flutuante (32 bits)
  - Duplo (64 bits)

- Inteiro (assinado e não assinado). Os seguintes tipos de números inteiros são suportados.
  - int8 e uint8
  - int16 e uint16
  - int32 e uint32
- Longo. Os seguintes tipos longos são compatíveis.
  - Longo (int64)
  - Não assinado por muito tempo (uint64)
- String
- Booleano

Execução de comandos

A execução de um comando é uma instância de um comando em execução em um dispositivo de destino. O veículo executa o comando usando os parâmetros que você especificou quando criou o comando ou quando iniciou a execução do comando. O veículo então executa as operações especificadas e retorna o status da execução.

> **Note**
>
> Para um determinado veículo, você pode executar vários comandos simultaneamente. Para obter informações sobre o número máximo de execuções simultâneas que você pode executar para cada veículo, consulte cotas de AWS IoT Device Management comandos.

Modelos de estado do último estado conhecido (LKS)

Os modelos de estado fornecem um mecanismo para que os proprietários de veículos rastreiem o estado de seus veículos. Para monitorar o último estado conhecido (LKS) de seus veículos quase em tempo real, você pode criar modelos de estado e associá-los aos seus veículos.

Usando o recurso de comandos, você pode realizar operações “On Demand” que podem ser usadas para coleta e processamento de dados estaduais. Por exemplo, você pode solicitar o estado atual do veículo uma única vez (buscar) ou ativar ou desativar modelos de estado do LKS implantados anteriormente para iniciar ou parar de relatar dados do veículo. Para exemplos que mostram como usar comandos com modelos de estado, consulteCenários de uso de comandos.

Status de execução do comando

Depois de iniciar a execução do comando, seu veículo pode publicar o status da execução e fornecer os motivos do status como informações adicionais sobre a execução. As seções a seguir descrevem os vários status de execução do comando e os códigos de status.

Tópicos

- Status da execução do comando, motivo, código e descrição
- Status de execução do comando e códigos de status
- Status do tempo limite de execução do comando

Status da execução do comando, motivo, código e descrição

Para relatar atualizações no status de execução do comando, seus veículos podem usar a `UpdateCommandExecution` API para publicar as informações de status atualizadas na nuvem, usando os tópicos reservados de Comandos descritos no guia do AWS IoT Core desenvolvedor. Ao relatar as informações de status, seus dispositivos podem fornecer contexto adicional sobre o status de cada execução de comando usando o `StatusReason` objeto `reasonCode` e os campos `reasonDescription` que estão contidos no objeto.

Status de execução do comando e códigos de status

A tabela a seguir mostra os vários códigos de status de execução de comandos e os status permitidos para os quais uma execução de comando pode fazer a transição. Também mostra se a execução de um comando é “terminal” (ou seja, não há mais atualizações de status disponíveis), se a alteração foi iniciada pelo veículo ou pela nuvem, os diferentes códigos de status predefinidos e como eles são mapeados para os status que são relatados pela nuvem.

- Para obter informações sobre como AWS IoT FleetWise usa os códigos de status predefinidos e o `statusReason` objeto, consulte Status do comando na documentação do software Edge Agent para AWS FleetWise IoT.
- Para obter informações adicionais sobre execuções terminais e não terminais e as transições entre os status, consulte Status de execução de comandos no guia do desenvolvedor.AWS IoT Core

Status e origem da execução do comando

Status de execução do comando	Description	Iniciado pelo dispositivo/ nuvem?	Execução do terminal?	Transições de status permitidas	Códigos de status predefinidos
`CREATED`	Quando a solicitação da API para começar a executar o comando (`StartCommandExecution` API) é bem-sucedida, o status de execução do comando muda para`CREATED`.	Nuvem	Não	• `IN_PROGRESS` • `SUCCEEDED` • `FAILED` • `REJECTED` • `TIMED_OUT`	Nenhum
`IN_PROGRESS`	Quando o veículo começa a executar o comando, ele pode publicar uma mensagem no tópico de resposta para atualizar o status. `IN_PROGRESS`	Dispositivo	Não	• IN_PROGRESS • SUCCEEDED • FAILED • REJEITADA • TEMPORIZADO_LIMITE	`COMMAND_STATUS_COMMAND_IN_PROGRESS`

Status de execução do comando	Description	Iniciado pelo dispositivo/ nuvem?	Execução do terminal?	Transições de status permitidas	Códigos de status predefinidos
`SUCCEEDED`	Quando o veículo tiver processado o comando com sucesso e concluído a execução, ele poderá publicar uma mensagem no tópico de resposta para atualizar o status`SUCCEED`.	Dispositivo	Sim	Não aplicável	`COMMAND_STATUS_SUCCEEDED`
`FAILED`	Quando o veículo falhou ao executar o comando, ele pode publicar uma mensagem no tópico de resposta para atualizar o status`FAILED`.	Dispositivo	Sim	Não aplicável	`COMMAND_STATUS_EXECUTION_FAILED`

Status de execução do comando	Description	Iniciado pelo dispositivo/ nuvem?	Execução do terminal?	Transições de status permitidas	Códigos de status predefinidos
`REJECTED`	Se o veículo não aceitar o comando, ele poderá publicar uma mensagem no tópico de resposta para atualizar o statusREJECTE	Dispositivo	Sim	Não aplicável	Nenhum

Status de execução do comando	Description	Iniciado pelo dispositivo/ nuvem?	Execução do terminal?	Transições de status permitidas	Códigos de status predefinidos
`TIMED_OUT`	O status de execução do comando pode mudar para `TIMED_OUT` devido a qualquer um dos motivos a seguir. • O resultado da execução do comando não foi recebido e a nuvem relata automaticamente um `TIMED_OUT` status. • O veículo relata que ocorreu um tempo limite quando ele estava tentando executar o	Dispositivo e nuvem	Não	• SUCCEEDED • FAILED • REJEITADA • TEMPORIZADO_LIMITE	`COMMAND_STATUS_EXECUTION_TIMEOUT`

Status de execução do comando	Description	Iniciado pelo dispositivo/ nuvem?	Execução do terminal?	Transições de status permitidas	Códigos de status predefinidos
	comando. Nesse caso, a execução do comando se torna terminal. Para obter mais informações sobre esse status, consulte Status do tempo limite de execução do comando.				

Status do tempo limite de execução do comando

O tempo limite de execução do comando pode ser relatado tanto pela nuvem quanto pelo dispositivo. Depois que o comando é enviado ao dispositivo, um temporizador é iniciado. Se não houver resposta recebida do dispositivo dentro da duração especificada, a nuvem relatará um `TIMED_OUT` status. Nesse caso, a execução do comando no `TIMED_OUT` status não é terminal.

O dispositivo pode substituir esse status por um status de terminal, como `SUCCEEDED`, `FAILED`, ou `REJECTED`. Ele também pode relatar que ocorreu um tempo limite ao executar o comando. Nesse caso, o status de execução do comando permanece em `TIMED_OUT`, mas os campos do `StatusReason` objeto são atualizados com base nas informações relatadas pelo dispositivo. A execução do comando no `TIMED_OUT` status agora se torna terminal.

Para obter informações adicionais, consulte Considerações sobre o tempo limite de execução do comando no guia do AWS IoT Core desenvolvedor.

Veículos e comandos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.
> Você é o único responsável por implantar comandos de forma segura e compatível com as leis aplicáveis.

Para usar o recurso de comandos:

1. Primeiro, crie um recurso de comando. Opcionalmente, especifique os parâmetros que contêm as informações necessárias para executar o comando.
2. Especifique o veículo alvo que receberá o comando e executará as ações especificadas.
3. Agora, você pode executar o comando no dispositivo de destino e verificar os detalhes da execução do comando para recuperar o status e usar CloudWatch os registros para solucionar problemas adicionais.

As seções a seguir mostram o fluxo de trabalho entre veículos e comandos.

Tópicos

- Visão geral do fluxo de trabalho
- fluxo de trabalho do veículo
- Fluxo de trabalho de comandos
- (Opcional) Notificações de comandos

Visão geral do fluxo de trabalho

As etapas a seguir fornecem uma visão geral do fluxo de trabalho de comandos entre seus veículos e comandos. Quando você usa qualquer um dos comandos das operações da API HTTP, a solicitação é assinada usando credenciais do Sigv4.

> **Note**
>
> Com exceção da operação da `StartCommandExecution` API, todas as operações executadas pelo protocolo HTTP usam o endpoint do plano de controle.

1. Estabeleça uma conexão MQTT e assine os tópicos de comandos

   Para se preparar para o fluxo de trabalho de comandos, os dispositivos devem estabelecer uma conexão MQTT com o `iot:Data-ATS` endpoint e assinar o tópico de solicitação de comandos mencionado acima. Opcionalmente, seus dispositivos também podem assinar os tópicos de resposta aceitos e rejeitados dos comandos.

2. Crie um modelo de veículo e um recurso de comando

   Agora você pode criar um veículo e um recurso de comando usando as operações da API `CreateVehicle` e do plano de `CreateCommand` controle. O recurso de comando contém as configurações a serem aplicadas quando o comando é executado no veículo.

3. Iniciar a execução do comando no dispositivo de destino

   Inicie a execução do comando no veículo usando a API do plano `StartCommandExecution` de dados com o endpoint específico da sua conta`iot:Jobs`. A API publica uma mensagem de carga útil codificada por protobuf no tópico de solicitação de comandos.

4. Atualizar o resultado da execução do comando

   O veículo processa o comando e a carga recebida e, em seguida, publica o resultado da execução do comando no tópico de resposta usando a `UpdateCommandExecution` API. Se seu veículo se inscreveu nos tópicos de resposta aceitos e rejeitados dos comandos, ele receberá uma mensagem indicando se a resposta foi aceita ou rejeitada pelo serviço em nuvem.

5. (Opcional) Recuperar o resultado da execução do comando

   Para recuperar o resultado da execução do comando, você pode usar a operação da API do plano de `GetCommandExecution` controle. Depois que seu veículo publicar o resultado da execução do comando no tópico de resposta, essa API retornará as informações atualizadas.

6. (Opcional) Inscrever-se e gerenciar eventos de comandos

   Para receber notificações sobre atualizações de status de execução de comandos, você pode se inscrever no tópico de eventos de comandos. Em seguida, você pode usar a API do plano

de `CreateTopicRule` controle para rotear os dados de eventos de comandos para outros aplicativos, como AWS Lambda funções ou Amazon SQS, e criar aplicativos em cima deles.

fluxo de trabalho do veículo

As etapas a seguir descrevem detalhadamente o fluxo de trabalho do veículo ao usar o recurso de comandos.

> **Note**
>
> As operações descritas nesta seção usam o protocolo MQTT.

1. Estabelecer uma conexão do MQTT

   Para preparar seus veículos para usar o recurso de comandos, ele deve primeiro se conectar ao agente de AWS IoT Core mensagens. Seu veículo deve ter permissão para realizar a `iot:Connect` ação para se conectar AWS IoT Core e estabelecer uma conexão MQTT com o agente de mensagens. Para encontrar o endpoint do plano de dados para seu Conta da AWS, use a `DescribeEndpoint` API ou o comando `describe-endpoint` CLI, conforme mostrado abaixo.

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

   A execução desse comando exibe o endpoint do plano de dados específico da conta, conforme mostrado abaixo.

   ```
   account-specific-prefix.iot.region.amazonaws.com
   ```

2. Inscreva-se no tópico de solicitação de comandos

   Depois que uma conexão for estabelecida, seus dispositivos poderão assinar o tópico de solicitação de AWS IoT comandos do MQTT. Quando você cria um comando e inicia a execução do comando no dispositivo de destino, uma mensagem de carga útil codificada em protobuf será publicada no tópico da solicitação pelo agente de mensagens. Seu dispositivo pode então receber a mensagem de carga útil e processar o comando. Neste exemplo, *<DeviceID>* substitua pelo identificador exclusivo do seu veículo alvo. Esse ID pode ser o identificador exclusivo do seu veículo ou o nome de uma coisa.

> **Note**
>
> A mensagem de carga que é enviada ao dispositivo deve usar o formato protobuf.

```
$aws/commands/things/<DeviceID>/executions/+/request/protobuf
```

3. (Opcional) Inscreva-se nos tópicos de resposta de comandos

   Opcionalmente, você pode assinar esses tópicos de resposta de comandos para receber uma mensagem indicando se o serviço de nuvem aceitou ou rejeitou a resposta do dispositivo.

   > **Note**
   >
   > É opcional que seus veículos se inscrevam nos tópicos `/accepted` e de `/rejected` resposta. Seus veículos receberão automaticamente essas mensagens de resposta, mesmo que não tenham se inscrito explicitamente nesses tópicos.

   ```
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/accepted
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/rejected
   ```

4. Atualizar o resultado de uma execução de comando

   O veículo alvo então processa o comando. Em seguida, ele usa a `UpdateCommandExecution` API para publicar o resultado da execução no seguinte tópico de resposta do MQTT.

   > **Note**
   >
   > Para um determinado veículo e execução de comando, ele *<DeviceID>* deve corresponder ao campo correspondente no tópico de solicitação ao qual o dispositivo se inscreveu.

   ```
   $aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf
   ```

A `UpdateCommandExecution` API é uma operação de API de plano de dados em MQTT autenticada com TLS.

- Se o serviço de nuvem processou com sucesso o resultado da execução do comando, uma mensagem será publicada no tópico aceito pelo MQTT. O tópico aceito usa o seguinte formato.

```
$aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
accepted
```

- Se o serviço de nuvem falhar ao processar o resultado da execução do comando, uma resposta será publicada no tópico rejeitado do MQTT. O tópico rejeitado usa o seguinte formato.

```
$aws/commands/things/<DeviceID>/executions/<ExecutionId>/response/protobuf/
rejected
```

Para obter mais informações sobre essa API e um exemplo, consulteAtualizar o resultado da execução do comando.

Fluxo de trabalho de comandos

As etapas a seguir descrevem detalhadamente o fluxo de trabalho dos comandos.

> **Note**
>
> As operações descritas nesta seção usam o protocolo HTTP.

1. Registre seu veículo

   Agora que você preparou seu veículo para usar o recurso de comandos, você pode preparar seu aplicativo registrando seu veículo e criando um comando que será enviado ao veículo. Para registrar o veículo, crie uma instância de um modelo de veículo (manifesto do modelo) usando a operação da API do plano de `CreateVehicle`controle. Para obter mais informações e exemplos, consulte Criar um veículo.

2. Crie um comando

   Use a operação da API do plano de controle `CreateCommand`HTTP para modelar comandos aplicáveis ao veículo que você está segmentando. Especifique todos os parâmetros e valores padrão a serem usados ao executar o comando e certifique-se de que ele use o `AWS-IoT-FleetWise` namespace. Para obter mais informações e exemplos de uso dessa API, consulteCriar um recurso de comando.

3. Inicie a execução do comando

   Agora você pode executar o comando que criou no veículo usando a operação da API do plano de `StartCommandExecution`dados. AWS IoT Device Management busca o comando e os parâmetros do comando e valida a solicitação recebida. Em seguida, ele invoca a AWS IoT FleetWise API com os parâmetros necessários para gerar a carga útil específica do veículo. A carga é então enviada ao dispositivo por meio AWS IoT Device Management do MQTT para o tópico de solicitação de comando no qual seu dispositivo se inscreveu. Para obter mais informações e exemplos de uso dessa API, consulteEnviar um comando (AWS CLI).

   ```
   $aws/commands/things/<DeviceID>/executions/+/request/protobuf
   ```

   > **Note**
   >
   > Se o dispositivo estava off-line quando o comando foi enviado da nuvem e as sessões persistentes do MQTT estão em uso, o comando espera no agente de mensagens. Se o dispositivo voltar a ficar on-line antes do tempo limite e se tiver se inscrito no tópico de solicitação de comandos, o dispositivo poderá processar o comando e publicar o resultado no tópico de resposta. Se o dispositivo não voltar a ficar on-line antes do tempo limite, a execução do comando atingirá o tempo limite e a mensagem de carga útil expirará.

4. Recupere a execução do comando

   Depois de executar o comando no dispositivo, use a operação da API do plano de `GetCommandExecution`controle para recuperar e monitorar o resultado da execução do comando. Você também pode usar a API para obter informações adicionais sobre os dados de execução, como quando foram atualizados pela última vez, quando a execução foi concluída e os parâmetros especificados.

> **Note**
>
> Para recuperar as informações de status mais recentes, seu dispositivo deve ter publicado o resultado da execução do comando no tópico de resposta.

Para obter mais informações e exemplos de uso dessa API, consulteObtenha a execução do comando.

(Opcional) Notificações de comandos

É possível assinar eventos de comandos para receber notificações quando o status de uma execução é alterado. As etapas a seguir mostram como assinar eventos de comandos e processá-los.

1. Criar uma regra de tópico

   É possível assinar eventos de tópicos de comandos e receber notificações quando o status de uma execução de comando é alterado. Você também pode criar uma regra de tópico para rotear os dados processados pelo veículo para outros aplicativos, como AWS Lambda funções. Você pode criar uma regra de tópico usando o AWS IoT console ou a operação da API do plano de `CreateTopicRule` AWS IoT Core controle. Para obter mais informações, consulte Criação de uma AWS IoT regra.

   Neste exemplo, substitua *<CommandID>* pelo identificador do comando para o qual você deseja receber notificações e *<CommandExecutionStatus>* pelo status da execução do comando.

   ```
   $aws/events/commandExecution/<CommandID>/<CommandExecutionStatus>
   ```

   > **Note**
   >
   > Para receber notificações de todos os comandos e os status de execução deles, você pode usar caracteres curinga e assinar o tópico a seguir.

   ```
   $aws/events/commandExecution/+/#
   ```

2. Receber e processar eventos de comandos

   Se você criou uma regra de tópico na etapa anterior para assinar eventos de comandos, poderá gerenciar as notificações push de comandos recebidas. Opcionalmente, você também pode criar aplicativos em cima dele, como with, Amazon SQS AWS Lambda, Amazon SNS ou Step AWS Functions usando a regra de tópico que você criou.

O código a seguir mostra um exemplo de carga útil para as notificações de eventos de comandos que você receberá.

```
{
    "executionId": "2bd65c51-4cfd-49e4-9310-d5cbfdbc8554",
    "status":"FAILED",
    "statusReason": {
         "reasonCode": "4",
         "reasonDescription": ""
    },
    "eventType": "COMMAND_EXECUTION",
    "commandArn":"arn:aws:iot:us-east-1:123456789012:command/0b9d9ddf-
e873-43a9-8e2c-9fe004a90086",
    "targetArn":"arn:aws:iot:us-east-1:123456789012:thing/5006c3fc-
de96-4def-8427-7eee36c6f2bd",
    "timestamp":1717708862107
}
```

Criar e gerenciar comandos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode configurar ações remotas reutilizáveis ou enviar instruções únicas e imediatas para seus dispositivos. Ao usar esse recurso, você pode especificar as instruções que seus dispositivos podem executar quase em tempo real. Um comando permite que você configure ações remotas reutilizáveis para o veículo de destino. Depois de criar um comando, você pode iniciar a execução de um comando direcionado a um veículo específico.

Este tópico mostra como você pode criar e gerenciar um recurso de comando usando a AWS IoT Core API ou AWS CLI o. Ele mostra como realizar as seguintes ações em um recurso de comando.

Tópicos

- Criar um recurso de comando
- Recuperar informações sobre um comando
- Listar comandos na sua conta
- Atualizar ou descontinuar um recurso de comando
- Excluir um recurso de comando

Criar um recurso de comando

Você pode usar a operação da API do plano de `CreateCommand` AWS IoT Core controle ou o FleetWise console de AWS IoT para criar um comando.

Crie um comando (console)

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

Para criar um comando

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, escolha Comandos.
3. Escolha Criar comando.
4. Especifique um ID de comando exclusivo para ajudá-lo a identificar o comando que você deseja executar no veículo.
5. (Opcional) Especifique um nome de exibição e uma descrição opcionais.
6. (Opcional) Selecione o atuador e o valor padrão do parâmetro. Os parâmetros especificam as ações que o veículo-alvo pode realizar ao receber o comando. Se você não adicionar parâmetros, precisará fornecê-los ao executar o comando.
7. Escolha uma função do IAM que conceda permissões para gerar a carga útil dos comandos. Consulte Controle de acesso.
8. Escolha Criar comando.

Crie um comando (AWS CLI)

O exemplo a seguir mostra como criar um comando com um parâmetro.

Considerações ao criar um comando

Quando você cria um comando em AWS IoT FleetWise:

- Você deve especificar o `roleArn` que concede permissão para criar e executar comandos em seu veículo. Para obter mais informações e sobre exemplos de políticas, incluindo quando as chaves KMS estão habilitadas, consulteConceda AWS IoT Device Management permissão para gerar a carga útil para comandos com AWS IoT FleetWise.
- Você deve especificar `AWS-IoT-FleetWise` como namespace.
- Em vez disso, você pode pular o `mandatory-parameters` campo e especificá-los em tempo de execução. Como alternativa, você pode criar um comando com parâmetros e, opcionalmente, especificar valores padrão para eles. Se você especificou valores padrão, em tempo de execução, poderá usar esses valores ou substituí-los especificando seus próprios valores. Para ver esses exemplos adicionais, consulteCenários de uso de comandos.
- Você pode especificar até três pares de nome-valor para o `mandatory-parameters` campo. No entanto, ao executar o comando no veículo, somente um par nome-valor é aceito, e o `name` campo deve usar o nome totalmente qualificado com o prefixo. `$actuatorPath.`

- *`command-id`*Substitua por um identificador exclusivo para o comando. Você pode usar UUID, caracteres alfanuméricos, “-” e “_”.
- *`role-arn`*Substitua pela função do IAM que concede permissão para criar e executar comandos, por exemplo,`"arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*`"`.
- (Opcional) *`display-name`* Substitua por um nome fácil de usar para o comando e *`description`* por uma descrição significativa do comando.
- Substitua *`name`* e *`value`* do `mandatory-parameters` objeto pelas informações necessárias para o comando que está sendo criado. O `name` campo é o nome totalmente qualificado, conforme definido no catálogo de sinais com `$actuatorPath.` o prefixo. Por exemplo, `name` pode ser *`$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode`* e `value` pode ser um booleano que indica um status do modo de direção, como*`{"B": false}`*.

```
aws iot create-command --command-id command-id \
```

```
    --role-arn role-arn \
    --description description \
    --display-name display-name \
    --namespace "AWS-IoT-FleetWise" \
    --mandatory-parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

A operação da `CreateCommand` API retorna uma resposta que contém o ID e o ARN (Amazon Resource Name) do comando.

```
{
    "commandId": "HandsOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"
}
```

Recuperar informações sobre um comando

Você pode usar a operação da API do plano de `GetCommand` AWS IoT Core controle para recuperar informações sobre um recurso de comando.

Para obter informações sobre um recurso de comando, execute o comando a seguir. *command-id*Substitua pelo identificador que foi usado ao criar o comando.

```
aws iot get-command --command-id command-id
```

A operação `GetCommand` da API retorna uma resposta que contém as informações a seguir.

- O ID e o ARN (Amazon Resource Name) do comando.
- A data e a hora em que o comando foi criado e atualizado pela última vez.
- O estado do comando que indica se ele está disponível para ser executado no veículo.
- Qualquer parâmetro que você especificou ao criar o comando.

```
{
    "commandId": "HandsOffSteeringMode",
```

```
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/HandsOffSteeringMode"",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode",
            "value": {"B": false }
        }
    ],
    "createdAt": "2024-03-23T11:24:14.919000-07:00",
    "lastUpdatedAt": "2024-03-23T11:24:14.919000-07:00",
    "deprecated": false,
    "pendingDeletion": false
}
```

Listar comandos na sua conta

Você pode usar a operação da API do plano de `ListCommands` AWS IoT Core controle para listar todos os comandos que você criou na sua conta.

Para listar comandos em sua conta, execute o comando a seguir. Por padrão, a API retorna comandos que foram criados para os dois namespaces. Para filtrar a lista para exibir somente os comandos que foram criados para AWS IoT FleetWise, execute o comando a seguir.

> **Note**
>
> Você também pode classificar a lista em ordem crescente ou decrescente, ou filtrar a lista para exibir somente comandos que tenham um nome de parâmetro de comando específico.

```
aws iot list-commands --namespace "AWS-IoT-FleetWise"
```

A operação `ListCommands` da API retorna uma resposta que contém as informações a seguir.

- O ID e o ARN (Amazon Resource Name) dos comandos.
- A data e a hora em que o comando foi criado e atualizado pela última vez.
- O estado do comando que indica se os comandos estão disponíveis para execução no veículo.

Atualizar ou descontinuar um recurso de comando

Você pode usar a operação da API do plano de `UpdateCommand` AWS IoT Core controle ou o FleetWise console de AWS IoT para atualizar um recurso de comando. Você pode atualizar o nome de exibição e a descrição de um comando. Você também pode descontinuar um comando se ele não estiver sendo usado no momento.

> **Note**
>
> Você não pode modificar as informações do namespace ou os parâmetros a serem usados ao executar o comando.

Atualizar um comando (console)

Atualizar um comando

Para atualizar um comando do console, acesse a página Comandos do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o comando que você deseja atualizar e, em seguida, escolha Editar.
2. Edite os detalhes do comando e escolha Salvar alterações.

Descontinuar um comando

Para descontinuar um comando do console, acesse a página Comandos do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o comando que você deseja descontinuar e, em seguida, escolha Descontinuar.
2. Confirme a suspensão de uso e, em seguida, escolha Descontinuar.

Atualizar um comando (AWS CLI)

Atualizar um comando

Para atualizar um recurso de comando, execute o comando a seguir. *`command-id`*Substitua pelo identificador do comando que você deseja atualizar e forneça o *`description`* e. *`display-name`*

```
aws iot update-command \
    --command-id command-id \
```

```
    --display-name display-name \
    --description description
```

A operação `UpdateCommand` da API retorna a seguinte resposta.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": false,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Descontinuar um comando

Você desaprova um comando quando não pretende mais continuar a usá-lo em seu dispositivo ou quando ele está desatualizado. O exemplo a seguir mostra como descontinuar um comando.

```
aws iot update-command \
    --command-id command-id \
    --deprecated
```

A operação da `UpdateCommand` API retorna uma resposta que contém o ID e o ARN (Amazon Resource Name) do comando.

```
{
    "commandId": "HandsOffSteeringMode",
    "deprecated": true,
    "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
}
```

Depois que um comando for descontinuado, as execuções de comandos existentes continuarão sendo executadas no veículo até se tornarem terminais. Para executar qualquer nova execução de comando, você deve usar a `UpdateCommand` API para restaurar o comando para que ele fique disponível. Para obter informações adicionais sobre a descontinuação e a restauração de um comando e considerações sobre ele, consulte Descontinuar um recurso de comando no Guia do desenvolvedor.AWS IoT Core

Excluir um recurso de comando

Você pode usar a operação da API do plano de `DeleteCommand` AWS IoT Core controle ou o FleetWise console de AWS IoT para excluir um recurso de comando.

> **Note**
>
> Uma ação de exclusão é permanente e não pode ser desfeita. O comando será removido permanentemente da sua conta.

Excluir um comando (console)

Para excluir um comando do console, acesse a página Comandos do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o comando que você deseja excluir e, em seguida, escolha Excluir.
2. Confirme que você deseja excluir o comando e escolha Excluir.

Excluir um comando (AWS CLI)

Para excluir um recurso de comando, execute o comando a seguir. *command-id*Substitua pelo identificador do comando que você deseja excluir. O exemplo a seguir mostra como excluir um recurso de comando.

```
aws iot delete-command --command-id command-id
```

Se a solicitação de exclusão for bem-sucedida:

- Se o comando estiver obsoleto por um período maior que o tempo limite máximo de 24 horas, o comando será excluído imediatamente e você verá um HTTP de 204. `statusCode`
- Se o comando não estiver obsoleto ou tiver sido preterido por um período menor que o tempo limite máximo, o comando estará em um `pending deletion` estado e você verá um HTTP de 202. `statusCode` O comando será removido automaticamente da sua conta após o tempo limite máximo de 24 horas.

Iniciar e monitorar execuções de comando

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Depois de criar um recurso de comando, você pode iniciar a execução de um comando no veículo de destino. Depois que o veículo começa a executar o comando, ele pode começar a atualizar o resultado da execução do comando e publicar atualizações de status e informações do resultado nos tópicos reservados do MQTT. Depois, você pode recuperar o status da execução do comando e monitorar o status das execuções em sua conta.

Este tópico mostra como você pode enviar um comando para seu veículo usando o console de IoT AWS CLI ou AWS IoT FleetWise . Também mostra como monitorar e atualizar o status da execução do comando.

Tópicos

- Atualizar o resultado da execução do comando
- Obtenha a execução do comando
- Listar as execuções de comandos em sua conta
- Excluir uma execução de comando

Enviar um comando (console)

Para enviar um comando do console, acesse a página Veículos do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o veículo para o qual você deseja enviar um comando.
2. Selecione Run command.
3. Selecione o ID do comando.
4. Especifique o tempo limite de execução do comando e escolha Executar comando.

Enviar um comando (AWS CLI)

Você pode usar a operação da API do plano de `StartCommandExecution` AWS IoT dados para enviar um comando para um veículo. O veículo então encaminha o comando para um serviço de middleware automotivo (como SOME/IP (Scalable Service-Oriented Middleware over IP)) ou o publica em uma rede veicular (como uma interface de dispositivo de rede de área controladora (CAN)). O exemplo a seguir usa a AWS CLI.

Tópicos

- Considerações ao enviar um comando
- Acessar o endpoint do plano de dados específico da conta
- Envie um exemplo de comando

Considerações ao enviar um comando

Quando você inicia a execução de um comando em AWS IoT FleetWise:

- Você deve provisionar qualquer AWS IoT coisa para o veículo. Para obter mais informações, consulte Provisionar AWS veículos de IoT FleetWise.
- Você já deve ter criado um comando com `AWS-IoT-FleetWise` como namespace e fornecido um `role-Arn` que conceda permissão para criar e executar comandos na IoT AWS . FleetWise Para obter mais informações, consulte Criar um recurso de comando.
- Você pode ignorar o `parameters` campo se optar por usar qualquer valor padrão que tenha sido especificado para os parâmetros ao criar o comando. Se o `mandatory-parameters` não foi especificado no momento da criação, ou se você quiser substituir qualquer valor padrão especificando seus próprios valores para os parâmetros, você deverá especificar o `parameters` campo. Para ver esses exemplos adicionais, consulteCenários de uso de comandos.
- Você pode especificar até três pares de nome-valor para o `mandatory-parameters` campo. No entanto, ao executar o comando no veículo, somente um par nome-valor é aceito, e o `name` campo deve usar o nome totalmente qualificado com o prefixo. `$actuatorPath.`

Acessar o endpoint do plano de dados específico da conta

Antes de executar o comando da API, você deve obter o URL do endpoint específico da conta para o endpoint. `iot:Jobs` Por exemplo, se você executar este comando:

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

Ele exibirá o URL do endpoint específico da conta, conforme mostrado no exemplo de resposta abaixo.

```
{
    "endpointAddress": "<account-specific-prefix>.jobs.iot.<region>.amazonaws.com"
}
```

Envie um exemplo de comando

Para enviar um comando para um veículo, execute o comando a seguir.

- *command-arn*Substitua pelo ARN do comando que você deseja executar. Você pode acessar essas informações na resposta do comando `create-command` da CLI.
- *target-arn*Substitua pelo ARN do dispositivo ou AWS IoT objeto de destino para o qual você deseja executar o comando.

  > **Note**
  >
  > Você pode especificar o ARN de destino de uma AWS IoT coisa (veículo de AWS IoT) FleetWise . Atualmente, não há suporte para grupos e frotas de coisas.

- *endpoint-url*Substitua pelo endpoint específico da conta que você obteve emAcessar o endpoint do plano de dados específico da conta, prefixado por, por exemplo`https://`,. `https://`*123456789012abcd*`.jobs.iot.`*ap-south-1*`.amazonaws.com`
- *value*Substitua *name* e pelo `mandatory-parameters` campo que você especificou ao criar o comando usando a `create-command` CLI.

  O `name` campo é o nome totalmente qualificado, conforme definido no catálogo de sinais com `$actuatorPath.` o prefixo. Por exemplo, `name` pode ser *`$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode`* e `value` pode ser um booleano que indica um status do modo de direção, como*`{"B": false}`*.
- (Opcional) Você também pode especificar um parâmetro adicional,`executionTimeoutSeconds`. Esse campo opcional especifica o tempo em segundos em que o dispositivo deve responder com o resultado da execução. Você pode configurar o tempo limite para um valor máximo de 24 horas.

Quando a execução do comando é criada, um cronômetro é iniciado. Antes que o cronômetro expire, se o status de execução do comando não mudar para um status que o torne terminal, como `SUCCEEDED` ou`FAILED`, o status mudará automaticamente para. `TIMED_OUT`

> **Note**
>
> O dispositivo também pode relatar um `TIMED_OUT` status ou substituir esse status por um status como`SUCCEEDED`, ou `FAILEDREJECTED`, e a execução do comando se tornará terminal. Para obter mais informações, consulte Status do tempo limite de execução do comando.

```
aws iot-jobs-data start-command-execution \
    --command-arn command-arn \
    --target-arn target-arn \
    --execution-timeout-seconds 30 \
    --endpoint-url endpoint-url \
    --parameters '[
        {
            "name": name,
            "value": value
        }
   ]'
```

A operação `StartCommandExecution` da API retorna um ID de execução do comando. Você pode usar esse ID para consultar o status, os detalhes e o histórico da execução do comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
 }
```

Depois de executar o comando, seus dispositivos receberão uma notificação contendo as seguintes informações. O `issued_timestamp_ms` campo corresponde à hora em que a `StartCommandExecution` API foi invocada. O `timeout_ms` corresponde ao valor de tempo limite configurado usando o `executionTimeoutSeconds` parâmetro ao invocar a `StartCommandExecution` API.

```
timeout_ms: 9000000
```

```
issued_timestamp_ms: 1723847831317
```

Atualizar o resultado da execução do comando

Para atualizar o status da execução do comando, seu dispositivo deve ter estabelecido uma conexão MQTT e inscrito no seguinte tópico de solicitação de comandos.

Neste exemplo, *<device-id>* substitua pelo identificador exclusivo do dispositivo de destino, que pode ser o nome `VehicleId` ou a coisa, e *<execution-id>* pelo identificador para a execução do comando.

> **Note**
>
> - A carga deve usar o formato protobuf.
> - É opcional que seus dispositivos se inscrevam nos tópicos `/accepted` e `/rejected` respondam. Seus dispositivos receberão essas mensagens de resposta mesmo que não tenham se inscrito explicitamente.

```
// Request topic
$aws/devices/<DeviceID>/command_executions/+/request/protobuf

// Response topics (Optional)
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/accepted/protobuf
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/rejected/protobuf
```

Seu dispositivo pode publicar uma mensagem no tópico de resposta dos comandos. Depois de processar o comando, ele envia uma resposta codificada em protobuf para esse tópico. O *<DeviceID>* campo deve corresponder ao campo correspondente no tópico da solicitação.

```
$aws/devices/<DeviceID>/command_executions/<ExecutionId>/response/<PayloadFormat>
```

Depois que seu dispositivo publicar uma resposta a esse tópico, você poderá recuperar as informações de status atualizadas usando a `GetCommandExecution` API. O status da execução de um comando pode ser qualquer um dos listados aqui.

- `IN_PROGRESS`

- `SUCCEEDED`
- `FAILED`
- `REJECTED`
- `TIMED_OUT`

Observe que a execução de um comando em qualquer um dos status `SUCCEEDEDFAILED`, e `REJECTED` é terminal, e o status é relatado pelo dispositivo. Quando a execução de um comando é terminal, isso significa que nenhuma atualização adicional será feita em seu status ou campos relacionados. Um `TIMED_OUT` status pode ser relatado pelo dispositivo ou pela nuvem. Se relatado pela nuvem, uma atualização do campo de motivo do status poderá ser feita posteriormente pelo dispositivo.

Por exemplo, o exemplo a seguir mostra um exemplo de mensagem MQTT publicada pelo dispositivo.

> **Note**
>
> Para o status de execução do comando, se seus dispositivos usarem o `statusReason` objeto para publicar as informações de status, você deverá se certificar de que:
>
> - O `reasonCode` usa `[A-Z0-9_-]+` o padrão e não excede 64 caracteres.
> - O `reasonDescription` comprimento não excede 1.024 caracteres. Ela pode usar qualquer caractere, exceto de controle, como novas linhas.

```
{
    "deviceId": "",
    "executionId": "",
    "status": "CREATED",
    "statusReason": {
        "reasonCode": "",
        "reasonDescription": ""
    }
}
```

Para ver um exemplo que mostra como você pode usar o cliente de teste AWS IoT Core MQTT para assinar os tópicos e ver as mensagens de execução do comando, consulte Visualização

de atualizações de comandos usando o cliente de teste MQTT no guia do AWS IoT Core desenvolvedor.

Obtenha a execução do comando

Você pode usar a operação da API do plano de `GetCommandExecution` AWS IoT controle para recuperar informações sobre a execução de um comando. Você já deve ter executado esse comando usando a operação da API `StartCommandExecution`.

Para recuperar os metadados de um comando executado, execute o comando a seguir.

- *`execution-id`*Substitua pelo ID do comando. Você pode acessar essas informações na resposta do comando `start-command-execution` da CLI.
- *`target-arn`*Substitua pelo ARN do veículo ou AWS IoT objeto de destino para o qual você deseja executar o comando.

```
aws iot get-command-execution --execution-id execution-id \
    --target-arn target-arn
```

A operação da `GetCommandExecution` API retorna uma resposta que contém informações sobre o ARN da execução do comando, o status da execução e a hora em que o comando começou a ser executado e quando foi concluído. O código a seguir mostra um exemplo de resposta da solicitação da API.

Para fornecer contexto adicional sobre o status de cada execução de comando, o recurso de comandos fornece um `statusReason` objeto. O objeto contém dois campos `reasonCode` `reasonDescription` e. Usando esses campos, seus dispositivos podem fornecer informações adicionais sobre o status da execução de um comando. Essas informações substituirão qualquer padrão `reasonCode` e `reasonDescription` isso será relatado na nuvem.

Para relatar essas informações, seus dispositivos podem publicar as informações de status atualizadas na nuvem. Então, ao recuperar o status de execução do comando usando a `GetCommandExecution` API, você verá os códigos de status mais recentes.

> **Note**
>
> O campo `completedAt` na resposta de execução corresponde ao momento em que o dispositivo relata o status terminal para a nuvem. No caso de `TIMED_OUT` status, esse campo será definido somente quando o dispositivo informar um tempo limite. Quando o

status `TIMED_OUT` é definido pela nuvem, o status `TIMED_OUT` não é atualizado. Para acessar mais informações sobre o comportamento do tempo limite, consulte Status do tempo limite de execução do comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/LockDoor",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myFrontDoor",
    "status": "SUCCEEDED",
    "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "Parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode":

         { "B": true }
    }'
}
```

Listar as execuções de comandos em sua conta

Use a operação da API HTTP do plano de `ListCommandExecutions` AWS IoT Core controle para listar todas as execuções de comandos em sua conta. O exemplo usa AWS CLI.

Tópicos

- Considerações ao listar as execuções de comandos
- Exemplo de como listar as execuções de comandos

Considerações ao listar as execuções de comandos

Veja a seguir algumas considerações ao usar a `ListCommandExecutions` API.

- Você deve especificar pelo menos o `targetArn` ou o, `commandArn` dependendo se deseja listar as execuções de um comando específico ou de um veículo-alvo. A solicitação de API não pode estar vazia e não pode conter os dois campos na mesma solicitação.

- Você deve fornecer somente as `startedTimeFilter` ou as `completedTimeFilter` informações. A solicitação de API não pode estar vazia e não pode conter os dois campos na mesma solicitação. Você pode usar os `after` campos `before` e do objeto para listar as execuções de comandos que foram criadas ou concluídas em um período específico.
- Os `after` campos `before` e não devem ser maiores que a hora atual. Por padrão, se você não especificar nenhum valor, o `before` campo será a hora atual e o `after` campo será a hora atual - 6 meses. Ou seja, dependendo do filtro que você usa, a API listará todas as execuções que foram criadas ou concluídas nos últimos seis meses.
- Você pode usar o `sort-order` parâmetro para especificar se deseja listar as execuções na ordem crescente. Por padrão, as execuções serão listadas em ordem decrescente se você não especificar esse campo.
- Você não pode filtrar as execuções de comandos com base em seu status ao listar as execuções de comandos para um ARN de comando.

Exemplo de como listar as execuções de comandos

O exemplo a seguir mostra como listar as execuções de comandos na Conta da AWS.

Ao executar o comando, você deve especificar se deseja filtrar a lista para exibir somente execuções de comandos que foram criadas para um dispositivo específico usando o `targetArn` ou execuções para um comando especificado usando o `commandArn`.

Neste exemplo, substitua:

- *`<target-arn>`* pelo nome do recurso da Amazon (ARN) do dispositivo no qual pretende realizar a execução, como `arn:aws:iot:`*`us-east-1`*`:`*`123456789012`*`:thing/`*`b8e4157c98f332cffb37627f`*.
- *`<target-arn>`* pelo nome do recurso da Amazon (ARN) do dispositivo no qual você planeja a execução, como `arn:aws:iot:`*`us-east-1`*`:`*`123456789012`*`:thing/`*`b8e4157c98f332cffb37627f`*.
- *`<after>`* pelo tempo após o qual você deseja listar as execuções criadas, por exemplo, `2024-11-01T03:00`.

```
aws iot list-command-executions \
--target-arn <target-arn> \
--started-time-filter '{after=<after>}' \
```

```
--sort-order "ASCENDING"
```

A execução desse comando gera uma resposta com uma lista das execuções de comando criadas, a hora em que elas começaram a ser realizadas e quando foram concluídas. Ela também fornece informações de status e o objeto `statusReason` que contém informações adicionais sobre o status.


```
{
    "commandExecutions": [
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "b2b654ca-1a71-427f-9669-e74ae9d92d24",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "TIMED_OUT",
            "createdAt": "2024-11-24T14:39:25.791000-08:00",
            "startedAt": "2024-11-24T14:39:25.791000-08:00"
        },
        {
            "commandArn": "arn:aws:iot:us-east-1:123456789012:command/TestMe002",
            "executionId": "34bf015f-ef0f-4453-acd0-9cca2d42a48f",
            "targetArn": "arn:aws:iot:us-east-1:123456789012:thing/
b8e4157c98f332cffb37627f",
            "status": "IN_PROGRESS",
            "createdAt": "2024-11-24T14:05:36.021000-08:00",
            "startedAt": "2024-11-24T14:05:36.021000-08:00"
        }
    ]
}
```


Excluir uma execução de comando

Se você não quiser mais usar uma execução de comando, poderá removê-la permanentemente da sua conta.

> **Note**
>
> A execução de um comando só poderá ser excluída se tiver entrado em um status terminal, como `SUCCEEDED`, `FAILED` ou `REJECTED`.

O exemplo a seguir mostra como excluir a execução de um comando usando o `delete-command-execution` AWS CLI comando. *<execution-id>*Substitua pelo identificador da execução do comando que você está excluindo.

```
aws iot delete-command-execution --execution-id <execution-id>
```

Se a solicitação da API for bem-sucedida, a execução do comando gerará um código de status de 200. Você pode usar a API `GetCommandExecution` para verificar se a execução do comando não existe mais na sua conta.

Exemplo: Usando comandos para controlar o modo de direção de um veículo (AWS CLI)

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

O exemplo a seguir mostra como usar o recurso de comandos usando AWS CLI o. Este exemplo usa um AWS IoT FleetWise veículo como dispositivo de destino para mostrar como você pode enviar um comando para controlar remotamente o modo de direção.

Tópicos

- Visão geral do exemplo do modo de direção do veículo
- Pré-requisitos
- Política do IAM para usar comandos remotos
- Executar AWS IoT comandos (AWS CLI)
- Limpeza

Visão geral do exemplo do modo de direção do veículo

Neste exemplo, você vai:

1. Crie um recurso de comando para a operação usando o `create-command` AWS CLI para alterar o modo de direção do veículo.
2. Recupere informações sobre o comando, como a hora em que ele foi criado ou atualizado pela última vez usando o. `get-command` AWS CLI
3. Envie o comando para o veículo usando o modo de direção como parâmetro obrigatório, que será executado no dispositivo. `start-command-execution` AWS CLI
4. Obtenha o resultado da execução do comando usando `get-command-execution` AWS CLI o. Você pode verificar quando a execução foi concluída e recuperar detalhes adicionais, como o resultado da execução e o tempo necessário para concluir a execução do comando.
5. Execute atividades de limpeza removendo quaisquer comandos e execuções de comandos que você não queira mais usar.

Pré-requisitos

Antes de executar este exemplo:

- Provisione seu AWS IoT FleetWise veículo como AWS IoT algo no AWS IoT registro. Você também deve adicionar um certificado à sua coisa, ativá-lo e anexar uma política à sua coisa. Seu dispositivo pode então se conectar à nuvem e executar os comandos. Para obter mais informações, consulte Provisionar veículos.
- Crie um usuário do IAM e uma política do IAM que conceda a você permissão para realizar as operações de API para usar comandos, conforme mostrado emPolítica do IAM para usar comandos remotos.

Política do IAM para usar comandos remotos

A tabela a seguir mostra um exemplo de política do IAM que concede acesso a todas as operações de API do plano de controle e do plano de dados para o recurso de comandos. O usuário do aplicativo terá permissões para realizar todas as operações da API de comando remoto, conforme mostrado na tabela.

Operação de API

Ação da API	Plano de controle/dados	Protocolo	Description	Recurso
`CreateCommand`	Ambiente de gerenciamento	HTTP	Cria um recurso de comando	• command
`GetCommand`	Ambiente de gerenciamento	HTTP	Recupera informações sobre um comando	• command
`UpdateCommand`	Ambiente de gerenciamento	HTTP	Atualiza as informações sobre um comando ou para descontinuá-lo	• command
`ListCommands`	Ambiente de gerenciamento	HTTP	Lista os comandos na sua conta	• command
`DeleteCommand`	Ambiente de gerenciamento	HTTP	Exclui um comando	• command
`StartCommandExecution`	Plano de dados	HTTP	Inicia a execução de um comando	• command • objeto
`UpdateCommandExecution`	Plano de dados	MQTT	Atualizar a execução de um comando	• command • objeto
`GetCommandExecution`	Ambiente de gerenciamento	HTTP	Recupera informações sobre a execução de um comando	• command • objeto
`ListCommandExecutions`	Ambiente de gerenciamento	HTTP	Lista as execuções de comandos em sua conta	• command • objeto
`DeleteCommandExecution`	Ambiente de gerenciamento	HTTP	Exclui a execução de um comando	• command • objeto

Neste exemplo, substitua:

- *us-east-1*com o seu Região da AWS, comoap-south-1.
- *111122223333* pelo número de sua Conta da AWS , como 57EXAMPLE833.
- *command-idcommand-id1*, e *command-id2* com seu identificador de comando exclusivo, como LockDoor ouTurnOffAC.
- *thing-name*com o nome da sua AWS IoT coisa, comomy_car.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "iot:CreateCommand",
                "iot:GetCommand",
                "iot:ListCommands",
                "iot:UpdateCommand",
                "iot:DeleteCommand"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:command/command-id1",
                "arn:aws:iot:us-east-1:111122223333:command/command-id2"
            ]
        },
        {
            "Action": [
                "iot:GetCommandExecution",
                "iot:ListCommandExecutions",
                "iot:DeleteCommandExecution"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:command/command-id",
                "arn:aws:iot:us-east-1:111122223333:thing/thing-name"
            ]
        },
        {
            "Action": "iot:StartCommandExecution",
            "Effect": "Allow",
```

```
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:command/command-id",
                "arn:aws:iot:us-east-1:111122223333:thing/thing-name"
            ]
        }
    ]
}
```

Executar AWS IoT comandos (AWS CLI)

O seguinte mostra como você pode usar o AWS CLI para executar operações de comandos e alterar o modo de direção do veículo.

1. Crie um recurso de comando para a operação do modo de direção

   Crie o comando que você deseja enviar ao seu dispositivo usando a `create-command` CLI. Neste exemplo, especifique:

   - `command-id` como *`TurnOffSteeringMode`*
   - `role-arn`como "`arn:aws:iam:`*`accountId`*`:role/`*`FwCommandExecutionRole`*" O `role-arn` deve ser fornecido, pois é a função do IAM que concede permissões para criar e executar comandos em seu veículo. Para obter mais informações, consulte Conceda AWS IoT Device Management permissão para gerar a carga útil para comandos com AWS IoT FleetWise.
   - `display-name`como "*`Turn off steering mode`*”
   - `namespace`deve ser `AWS-IoT-FleetWise`
   - `mandatory-parameters`como um par nome-valor, com as "*`$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode`*" e defaultValue `name` como *`{ "S": "true" }`*

   > **Note**
   >
   > Você também pode criar um comando sem especificar nenhum parâmetro obrigatório. Em seguida, você deve especificar os parâmetros a serem usados ao executar o comando usando a `start-command-execution` CLI. Para ver um exemplo, consulte Cenários de uso de comandos.

> **Important**
>
> Ao usar o `AWS-IoT-FleetWise` namespace, você deve garantir que o `Name` campo especificado como parte do `mandatory-parameters` use o `$actuatorPath.` prefixo e que o `Value` campo use o tipo de dados de string.

```
aws iot create-command \
    --command-id TurnOffSteeringMode \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --display-name "Turn off steering mode" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
        "name": "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
        "defaultValue": { "S": "true" }
      }
    ]'
```

A saída a seguir mostra um exemplo de resposta da CLI, onde `ap-south-1` e `123456789012` são exemplos do ID Região da AWS e. Conta da AWS

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode"
}
```

Para obter exemplos adicionais sobre o uso desse comando, consulteCriar um recurso de comando.

2. Recupere informações sobre o comando

   Execute o comando a seguir para recuperar informações sobre o comando, onde `command-id` está o ID do comando na saída da `create-command` operação acima.

> **Note**
>
> Se você criar mais de um comando, poderá usar a `ListCommands` API para listar todos os comandos em sua conta e, em seguida, usar a `GetCommand` API para obter informações adicionais sobre um comando específico. Para obter mais informações, consulte Listar comandos na sua conta.

```
aws iot get-command --command-id TurnOffSteeringMode
```

A execução desse comando gera a resposta a seguir. Você verá a hora em que o comando foi criado e quando foi atualizado pela última vez, todos os parâmetros que você especificou e se o comando está disponível para execução no dispositivo.

```
{
    "commandId": "TurnOffSteeringMode",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "namespace": "AWS-IoT-FleetWise",
    "mandatoryParameters":[
        {
            "name":
 "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode",
            "defaultValue": {"S": "true" }
        }
    ],
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "lastUpdatedAt": "2024-03-23T00:50:10.095000-07:00",
    "deprecated": false
}
```

Para obter exemplos adicionais sobre o uso desse comando, consulteRecuperar informações sobre um comando.

3. Inicie a execução do comando

   Execute o comando a seguir para começar a executar o comando, onde `command-arn` está o comando ARN na saída `get-command` da operação acima. `target-arn`É o ARN do dispositivo de destino para o qual você está executando o comando, por exemplo,. *myVehicle*

Neste exemplo, como você forneceu valores padrão para os parâmetros ao criar o comando, a `start-command-execution` CLI pode usar esses valores ao executar o comando. Você também pode optar por substituir o valor padrão especificando um valor diferente para os parâmetros ao usar a CLI.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:ap-south-1:123456789012:command/TurnOffSteeringMode \
    --target-arn arn:aws:iot:ap-south-1:123456789012:thing/myVehicle
```

A execução desse comando exibe um ID de execução. Você pode usar esse ID para consultar o status, os detalhes e o histórico da execução do comando.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542"
}
```

Para obter exemplos adicionais sobre o uso da CLI, consulte. Enviar um comando (AWS CLI)

4. Recupere informações sobre a execução do comando

Execute o comando a seguir para recuperar informações sobre o comando que você executou no dispositivo de destino. Especifique o`execution-id`, que você obteve como saída da `start-command-execution` operação acima, e o`target-arn`, que é o ARN do dispositivo que você está segmentando.

> **Note**
>
> - Para obter as informações de status mais recentes, seus dispositivos devem ter publicado as informações de status atualizadas no tópico de resposta reservada do MQTT para comandos que usam a API do `UpdateCommandExecution` MQTT. Para obter mais informações, consulte Atualizar o resultado da execução do comando.
> - Se você iniciar mais de uma execução de comando, poderá usar a `ListCommandExecutions` API para listar todas as execuções de comandos em sua conta e, em seguida, usar a `GetCommandExecution` API para obter informações adicionais sobre uma execução específica. Para obter mais informações, consulte Listar as execuções de comandos em sua conta.

```
aws iot get-command-execution \
    --execution-id <"07e4b780-7eca-4ffd-b772-b76358da5542"> \
    --target-arn arn:aws:iot:us-east-1:<account>:thing/myVehicle
```

A execução desse comando retorna informações sobre a execução do comando, o status da execução, a hora em que a execução foi iniciada e a hora em que foi concluída. Por exemplo, a resposta a seguir mostra que a execução do comando foi bem-sucedida no dispositivo de destino e o modo de direção foi desativado.

```
{
    "executionId": "07e4b780-7eca-4ffd-b772-b76358da5542",
    "commandArn": "arn:aws:iot:ap-south-1:123456789012:command/
TurnOffSteeringMode",
    "targetArn": "arn:aws:iot:ap-south-1:123456789012:thing/myVehicle",
    "result": "SUCCEEDED",
     "statusReason": {
        "reasonCode": "65536",
        "reasonDescription": "SUCCESS"
    },
    "result": {
        "KeyName": {
            "S": "",
            "B": true,
            "BIN": null
        }
    },
    "createdAt": "2024-03-23T00:50:10.095000-07:00",
    "completedAt": "2024-03-23T00:50:10.095000-07:00",
    "parameters": '{
         "$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode":
         { "S": "true" }
    }'
}
```

Limpeza

Agora que você criou um comando e o executou em seu dispositivo, se não pretende mais usá-lo, você pode excluí-lo. Todas as execuções de comando pendentes em andamento continuarão sendo executadas sem serem afetadas pela solicitação de exclusão.

> **Note**
>
> Como alternativa, você também pode descontinuar um comando se ele estiver desatualizado e talvez seja necessário usá-lo posteriormente para ser executado no dispositivo de destino.

1. (Opcional) Descontinuar o recurso de comando

   Execute o comando a seguir para descontinuar o comando, onde `command-id` está o ID do comando na saída da `get-command` operação acima.

   ```
   aws iot update-command \
      --command-id TurnOffSteeringMode \
      --deprecated
   ```

   A execução desse comando retorna uma saída que mostra que o comando foi descontinuado. Você também pode usar a CLI para restaurar o comando.

   > **Note**
   >
   > Você também pode usar a `update-command` CLI para atualizar o nome de exibição e a descrição de um comando. Para obter informações adicionais, consulte Atualizar ou descontinuar um recurso de comando.

   ```
   {
       "commandId": "TurnOffSteeringMode",
       "deprecated": true,
       "lastUpdatedAt": "2024-05-09T23:16:51.370000-07:00"
   }
   ```

2. Exclua o comando

   Execute o comando a seguir para excluir o comando, especificado pelo`command-id`.

   > **Note**
   >
   > A ação de exclusão é permanente e não pode ser desfeita.

   ```
   aws iot delete-command --command-id TurnOffSteeringMode
   ```

   Se a solicitação de exclusão for bem-sucedida, você verá um HTTP `statusCode` de 202 ou 204, dependendo se você marcou o comando para descontinuação e quando ele foi descontinuado. Para obter mais informações e um exemplo, consulte Excluir um recurso de comando.

   Você pode usar a `get-command` CLI para verificar se o comando foi removido da sua conta.

3. (Opcional) Exclua as execuções do comando

   Por padrão, todas as execuções de comandos serão excluídas em seis meses a partir da data em que você as criou. Você pode visualizar essas informações usando o `timeToLive` parâmetro da `GetCommandExecution` API.

   Como alternativa, se sua execução de comando se tornou terminal, como quando seu status de execução é de`SUCCEEDED`,`FAILED`, ou`REJECTED`, você pode excluir a execução do comando. Execute o comando a seguir para excluir a execução, onde `execution-id` está o ID de execução na saída da `get-command-execution` operação acima.

   ```
   aws iot delete-command-execution \
               --execution-id "07e4b780-7eca-4ffd-b772-b76358da5542"
   ```

   Você pode usar a `get-command-execution` CLI para verificar se a execução do comando foi removida da sua conta.

Cenários de uso de comandos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Ao usar o recurso de comandos, você pode criar e executar comandos nos seguintes cenários:

- Você pode omitir os parâmetros durante a criação e especificar somente o ID do comando. Nesse caso, você precisa especificar os parâmetros a serem usados ao executar o comando no dispositivo de destino.
- Você pode especificar um ou mais parâmetros e configurar valores padrão para eles ao criar o comando. Fornecer valores padrão ajudará você a evitar o envio de comandos imprecisos.
- Você pode especificar um ou mais parâmetros e configurar valores para eles ao criar o comando. Mais de um parâmetro pode ser fornecido, mas somente um deles será executado, e o `Name` campo desse parâmetro deve usar o `$actuatorPath` prefixo.

Esta seção fornece alguns cenários de uso da `CreateCommand` e da `StartCommandExecution` API e o uso dos parâmetros. Também mostra alguns exemplos de uso de comandos com modelos de estado.

Tópicos

- Criando um comando sem parâmetros
- Criação de um comando com valores padrão para parâmetros
- Criação de um comando com valores de parâmetros
- Usando comandos com modelos de estado

Criando um comando sem parâmetros

O caso de uso a seguir mostra como você pode usar a `CreateCommand` API ou a `create-command` CLI para criar um comando sem parâmetros. Ao criar um comando, você só precisa fornecer um ID de comando e um ARN de função.

Esse caso de uso é especialmente útil em casos de uso recorrentes, como quando você deseja enviar o mesmo comando várias vezes para um veículo. Nesse caso, o comando não está vinculado a um atuador específico e oferece a flexibilidade de executar o comando em qualquer atuador. Em vez disso, você deve especificar os parâmetros em tempo de execução ao executar o comando usando a `StartCommandExecution` API ou a `start-command-execution` CLI, que inclui os atuadores e os valores do sinal físico.

Criando um comando sem **`mandatory-parameters`** entrada

Esse caso de uso mostra como criar um comando sem a entrada obrigatória de parâmetros.

```
aws iot create-command \
    --command-id "UserJourney1" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --description "UserJourney1 - No mandatory parameters" \
    --namespace "AWS-IoT-FleetWise"
```

Executando um comando criado sem **`mandatory-parameters`** entrada

Neste primeiro exemplo, o comando criado acima permite que você execute um comando em qualquer atuador sem restrições. `actuator1`Para definir um valor de 10, execute:

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/UserJourney1 \
    --target-arn arn:aws:iot:region:111122223333:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

Da mesma forma, você pode executar um comando definido `actuator3` como um valor de`true`.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/UserJourney1 \
    --target-arn arn:aws:iot:region:111122223333:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator3": {"S": "true"}
    }'
```

Criação de um comando com valores padrão para parâmetros

Esse comando só permite que você execute um comando no atuador especificado. Fornecer valores padrão ajudará você a evitar o envio de comandos imprecisos. Por exemplo, um `LockDoor` comando que tranca e destranca portas pode ser configurado com um valor padrão para evitar que o comando destranque portas acidentalmente.

Esse caso de uso é especialmente útil quando você deseja enviar o mesmo comando várias vezes e realizar ações diferentes no mesmo atuador, como trancar e destrancar as portas de um veículo. Se você quiser definir o atuador para o valor padrão, não precisará passar nenhum `parameters` para a CLI`start-command-execution`. Se você especificar um valor diferente para o `parameters` na `start-command-execution` CLI, ele substituirá o valor padrão.

Criação de um comando com valores padrão para **`mandatory-parameters`**

O comando a seguir mostra como fornecer um valor padrão para o atuador1.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1",
            "defaultValue": {"S": "0"}
        }
    ]'
```

Executando um comando criado com valores padrão para **`mandatory-parameters`**

O comando `UserJourney2` permite que você execute um comando sem a necessidade de passar um valor de entrada durante o tempo de execução. Nesse caso, a execução em tempo de execução usará os valores padrão especificados durante a criação.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/UserJourney3 \
    --target-arn arn:aws:iot:region:111122223333:thing/target-vehicle
```

Você também pode passar um valor diferente para o mesmo atuador, atuador1, durante o tempo de execução, o que substituirá o valor padrão.

```
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/UserJourney3 \
    --target-arn arn:aws:iot:region:111122223333:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "139"}
    }'
```

Criação de um comando com valores de parâmetros

Esse comando só permite que você execute um comando no atuador especificado. Também força você a definir um valor para o atuador durante o tempo de execução.

Esse caso de uso é especialmente útil quando você deseja que o usuário final execute apenas determinadas ações especificadas em alguns dos atuadores ao executá-los no veículo.

> **Note**
>
> Você pode ter mais do que pares nome-valor para a `mandatory-parameters` entrada, com valores padrão para alguns ou todos eles. Em tempo de execução, você pode então determinar o parâmetro que deseja usar ao executar no atuador, desde que o nome do atuador use o nome totalmente qualificado com o prefixo. `$actuatorPath.`

Criação de comando sem valores padrão para **`mandatory-parameters`**

Esse comando só permite que você execute um comando no atuador especificado. Também força você a definir um valor para o atuador durante o tempo de execução.

```
aws iot create-command \
    --command-id "UserJourney2" \
    --namespace "AWS-IoT-FleetWise" \
    --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \
    --mandatory-parameters '[
        {
            "name": "$actuatorPath.Vehicle.actuator1"
        }
    ]'
```

Executando um comando criado sem valores padrão para **`mandatory-parameters`**

Ao executar o comando, nesse caso, você deve especificar um valor para atuador1. A execução do comando mostrada abaixo definirá com sucesso o valor de `actuator1` para`10`.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/UserJourney2 \
    --target-arn arn:aws:iot:region:111122223333:thing/target-vehicle \
    --parameters '{
        "$actuatorPath.Vehicle.actuator1": {"S": "10"}
    }'
```

Usando comandos com modelos de estado

Você também pode usar as operações de API de comandos para coleta e processamento de dados estaduais. Por exemplo, você pode obter um instantâneo de estado único ou ativar ou desativar modelos de estado para iniciar ou parar de coletar dados de estado do veículo. Os exemplos a seguir mostram como usar o recurso de comandos com modelos de estado. Para obter mais informações, consulte Operações de modelo de estado para a coleta e processamento de dados.

> **Note**
>
> O campo Nome especificado como parte da `mandatory-parameters` entrada deve usar o `$stateTemplate` prefixo.

Exemplo 1: Criação de comandos para modelos de estado com valores padrão

Este exemplo mostra como usar a `create-command` CLI para ativar modelos de estado.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name"
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
```

```
      }
    ]'
```

Da mesma forma, o comando a seguir mostra um exemplo de como você pode usar a `start-command-execution` CLI para modelos de estado.

```
aws iot-data start-command-execution \
    --command-arn arn:aws:iot:region:111122223333:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:region:111122223333:thing/<VEHICLE_NAME> \
    --parameters '{
       "$stateTemplate.name": {"S": "ST345"}
    }'
```

Exemplo 2: Criação de comandos para modelos de estado sem valores padrão

O comando a seguir cria vários modelos de estado sem valores padrão para nenhum dos parâmetros. Isso força você a executar o comando com esses parâmetros e os valores para eles.

```
aws iot create-command \
    --command-id <COMMAND_ID> \
    --display-name "Activate State Template" \
    --namespace AWS-IoT-FleetWise \
    --mandatory-parameters '[
      {
          "name": "$stateTemplate.name",
          "defaultValue": {"S": "ST123"}
      },
      {
          "name": "$stateTemplate.operation",
          "defaultValue": {"S": "activate"}
      },
      {
          "name": "$stateTemplate.deactivateAfterSeconds",
          "defaultValue": {"L": "120"}
      }
    ]'
```

O comando a seguir mostra como você pode usar a `start-command-execution` CLI para o exemplo acima.

```
aws iot-data start-command-execution \
```

```
    --command-arn arn:aws:iot:region:111122223333:command/<COMMAND_ID> \
    --target-arn arn:aws:iot:region:111122223333:thing/<VEHICLE_NAME> \
    --parameters '{
        "$stateTemplate.name": {"S": "ST345"},
        "$stateTemplate.operation": {"S": "activate"},
        "$stateTemplate.deactivateAfterSeconds" : {"L": "120"}
```

Monitorar o último estado conhecido de seus veículos

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode monitorar o último estado conhecido de seus veículos quase em tempo real criando modelos de estado e associando-os aos seus veículos. Veículos associados a modelos de estado transmitem dados de telemetria com uma estratégia de atualização `onChange` ou de `periodic` atualização. Com uma estratégia de atualização contínua, os veículos associados transmitem dados de telemetria quando há uma mudança. Durante uma estratégia de atualização periódica, os veículos associados transmitem dados de telemetria durante um período de tempo especificado.

Com operações sob demanda, você pode solicitar o estado atual do veículo de uma só vez (buscar). Você também pode ativar ou desativar modelos de estado implantados anteriormente para iniciar ou parar de relatar dados de estado do veículo. As últimas operações de estado conhecidas são executadas usando o AWS IoT comando APIs.

Cada modelo de estado contém as seguintes informações.

`name`

O alias exclusivo do modelo de estado.

`signalCatalogArn`

O Amazon Resource Name (ARN) do catálogo de sinais associado ao modelo de estado.

`stateTemplateProperties`

Uma lista de sinais dos quais os dados são coletados. As propriedades do modelo de estado determinam as atualizações específicas do sinal que o veículo envia para a nuvem.

`dataExtraDimensions`

Uma lista de atributos do veículo a serem incluídos nos dados processados codificados pelos buffers de protocolo (Protobuf).

`metadataExtraDimensions`

Uma lista de atributos do veículo a serem publicados com os dados processados como uma propriedade do usuário do MQTT 5.

`id`

Um identificador exclusivo gerado pelo serviço.

Para obter métodos para coletar dados enviados por um veículo que usa o FleetWise software Edge Agent para AWS IoT, consulte. Processe os últimos dados do veículo estadual conhecido usando mensagens MQTT Para obter mais informações sobre como associar um modelo de estado a um veículo, consulteCrie um veículo AWS de IoT FleetWise.

Tópicos

- Crie um modelo de AWS estado de IoT FleetWise
- Atualizar um modelo AWS de estado de IoT FleetWise
- Excluir um modelo AWS de estado de IoT FleetWise
- Obtenha informações AWS do modelo de FleetWise estado da IoT
- Operações de modelo de estado para a coleta e processamento de dados

Crie um modelo de AWS estado de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a FleetWise API ou o console de AWS IoT para criar um modelo de estado. Os modelos de estado fornecem um mecanismo para rastrear o estado de seus veículos. O FleetWise software Edge Agent para AWS IoT executado no veículo coleta e envia atualizações de sinal para a nuvem.

Crie um modelo de estado (console)

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

Para criar um modelo de estado

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, escolha Modelos de estado.
3. Na página Modelos de estado, escolha Criar modelo de estado.
4. Em Detalhes do modelo estadual, insira um nome para o modelo estadual e, opcionalmente, insira uma descrição.
5. Em Escolher sinais, adicione sinais dos quais você deseja obter informações sobre o status do veículo.
6. Escolha Criar modelo de estado.

Depois de criar com sucesso um modelo de estado, você o verá listado na página Modelos de estado. Agora você pode associá-lo a um veículo.

Crie um modelo de estado (AWS CLI)

Você pode usar a operação CreateStateTemplateda API para criar um modelo de estado. O exemplo a seguir usa a AWS CLI.

Para criar um modelo de estado, execute o comando a seguir.

*create-state-template*Substitua pelo nome do arquivo.json que contém a configuração do modelo de estado.

```
aws iotfleetwise create-state-template \
    --cli-input-json file://create-state-template.json
```

Example configuração do modelo de estado

`stateTemplateProperties`deve conter os nomes totalmente qualificados dos sinais.

`dataExtraDimensions`e `metadataExtraDimensions` deve conter os nomes totalmente qualificados dos atributos do veículo. As dimensões especificadas substituem quaisquer valores de dimensão existentes no modelo de estado.

```
{
    "name": "state-template-name",
    "signalCatalogArn": "arn:aws:iotfleetwise:us-east-1:account:signal-catalog/catalog-
name",
```

```
    "stateTemplateProperties": [
        "Vehicle.Signal.One",
        "Vehicle.Signal.Two"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Associar um modelo de FleetWise estado de AWS IoT a um veículo

Associar um modelo de estado a um veículo (console)

Você pode usar o FleetWise console de AWS IoT para adicionar modelos de estado associados a um veículo.

Para associar um modelo de estado

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Veículos.
3. Escolha um veículo na lista para abrir sua página de detalhes.
4. Na guia Modelos de estado, escolha Gerenciar modelos de estado.
5. Escolha Adicionar modelo de estado.
6. Selecione um modelo de estado e escolha seu método de relatório.
    a. Em caso de alteração — O modelo de estado relatará as alterações no estado do veículo.
    b. Periódico — O modelo de estado relatará atualizações no intervalo de tempo especificado.
7. Escolha Salvar alterações.

Associar um modelo de FleetWise estado de AWS IoT a um veículo ()AWS CLI

Associe o modelo de estado criado a um veículo para permitir a coleta de atualizações de estado do veículo para a nuvem. Para fazer isso, use:

- Ao criar um veículo, use o `stateTemplates` campo do `create-vehicle` comando. Para obter mais informações, consulte Crie um veículo AWS de IoT FleetWise.
- Ao atualizar um veículo, use os `stateTemplatesToRemove` campos `stateTemplatesToAdd` ou do `update-vehicle` comando. Para obter mais informações, consulte Atualizar um veículo AWS de IoT FleetWise.

Atualizar um modelo AWS de estado de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação de UpdateStateTemplateAPI ou o FleetWise console de AWS IoT para atualizar um modelo de estado existente.

Atualizar um modelo de estado (console)

Para atualizar um modelo de estado a partir do console, acesse a página Modelos de estado do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o modelo de estado que você deseja atualizar e escolha Editar.
2. Edite os detalhes do modelo de estado e escolha Salvar alterações.

Atualizar um modelo de estado (AWS CLI)

Para atualizar um modelo de estado, execute o comando a seguir.

*`update-state-template`*Substitua pelo nome do arquivo.json que contém a configuração do modelo de estado.

```
aws iotfleetwise update-state-template \
    --cli-input-json file://update-state-template.json
```

Example configuração do modelo de estado

Eles `stateTemplateProperties` devem conter os nomes totalmente qualificados dos sinais.

O `dataExtraDimensions` e `metadataExtraDimensions` deve conter os nomes totalmente qualificados dos atributos do veículo.

```
{
    "identifier": "state-template-name",
    "stateTemplatePropertiesToAdd": [
        "Vehicle.Signal.Three"
    ],
    "stateTemplatePropertiesToRemove": [
        "Vehicle.Signal.One"
    ],
    "dataExtraDimensions": [
        "Vehicle.Attribute.One",
        "Vehicle.Attribute.Two"
    ],
    "metadataExtraDimensions": [
        "Vehicle.Attribute.Three",
        "Vehicle.Attribute.Four"
    ]
}
```

Excluir um modelo AWS de estado de IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação de DeleteStateTemplateAPI ou o FleetWise console de AWS IoT para excluir um modelo de estado.

Excluir um modelo de estado (console)

Para excluir um modelo de estado do console, acesse a página Modelos de estado do FleetWise console de AWS IoT e execute as etapas a seguir.

1. Escolha o modelo de estado que você deseja excluir e, em seguida, escolha Excluir.
2. Confirme que você deseja excluir o modelo de estado e escolha Excluir.

Excluir um modelo de estado (AWS CLI)

Para excluir um modelo de estado, execute o comando a seguir.

*identifier*Substitua pelo nome ou ID do modelo de estado.

```
aws iotfleetwise delete-state-template \
    --identifier idenitfier
```

Obtenha informações AWS do modelo de FleetWise estado da IoT

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode usar a operação GetStateTemplateda API para recuperar informações sobre um modelo de estado. O exemplo a seguir usa a AWS CLI.

*identifier*Substitua pelo nome do modelo de estado.

```
aws iotfleetwise get-state-template \
    --identifier idenitfier
```

Você pode usar a operação ListStateTemplatesda API para recuperar uma lista dos modelos de estado criados. O exemplo a seguir usa a AWS CLI.

```
aws iotfleetwise list-state-templates
```

Se você ativou a criptografia usando uma AWS KMS chave gerenciada pelo cliente, inclua a seguinte declaração de política para que sua função possa invocar as operações `GetStateTemplate` ou a `ListStateTemplates` API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:kms:us-east-1:111122223333:key/KMS_KEY_ID"
            ]
        }
    ]
}
```

Operações de modelo de estado para a coleta e processamento de dados

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

As seções a seguir descrevem como usar modelos de estado para ativar e desativar a coleta de dados, realizar uma operação de busca e processar dados de estado de seus veículos.

Tópicos

- Ative e desative a coleta de dados de estado usando modelos de estado
- Obtenha um instantâneo do estado do veículo usando modelos de estado
- Processe os últimos dados do veículo estadual conhecido usando mensagens MQTT

Ative e desative a coleta de dados de estado usando modelos de estado

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

As seções a seguir descrevem como ativar e desativar a ingestão de dados com modelos de estado usando o. AWS CLI

> **Important**
>
> Antes de começar, verifique se você já criou um modelo de estado e o associou e sua estratégia de atualização a um veículo.

Você deve ativar um modelo de estado para que o Edge Agent possa enviar atualizações de sinal para a nuvem.

Para realizar essas operações com modelos de estado, primeiro crie um recurso de comando e, em seguida, inicie a execução do comando no veículo. A seção a seguir descreve como usar essa API e como ativar e desativar a ingestão de dados.

Tópicos

- Usar a API CreateCommand
- Exemplo: ativar um modelo de estado
- Exemplo: desativar um modelo de estado

Usar a API **`CreateCommand`**

Crie um recurso de comando no namespace `AWS-IoTFleetwise` "" e use os seguintes parâmetros ao criar ou enviar um recurso de comando para um modelo de estado:

- `$stateTemplate.name`— O nome do modelo de estado no qual realizar a operação. O modelo de estado deve ser aplicado ao veículo antes que você possa realizar uma operação. Para obter mais informações, consulte Associar um modelo de FleetWise estado de AWS IoT a um veículo.

- `$stateTemplate.operation`— A operação a ser executada no modelo de estado. Use um dos seguintes valores para esse parâmetro:
  - `activate`— O Edge Agent começa a enviar atualizações de sinal para a nuvem com base no `stateTemplateUpdateStrategy` que você especificou (em alteração ou periódico) ao aplicar o modelo de estado ao veículo. Para obter mais informações, consulte Associar um modelo de FleetWise estado de AWS IoT a um veículo.

    Além disso, você pode definir um tempo de desativação automática do modelo de estado para interromper as atualizações após um período de tempo especificado. Se um horário de desativação automática não for fornecido, os modelos de estado continuarão enviando atualizações até que uma chamada de desativação seja emitida.

    Assim que o `activate` comando for recebido, o dispositivo deverá enviar os sinais especificados no modelo de estado de acordo com a estratégia de atualização. AWS FleetWise A IoT recomenda que, quando um comando de ativação for recebido pelo dispositivo, a primeira mensagem enviada contenha um instantâneo de todos os sinais no modelo de estado. As mensagens subsequentes devem ser enviadas de acordo com a estratégia de atualização.
  - `deactivate`— O Edge Agent para de enviar atualizações de sinal para a nuvem.
  - `fetchSnapshot`— O Edge Agent envia um instantâneo único dos sinais definidos no modelo de estado, independentemente do `stateTemplateUpdateStrategy` que você especificou ao aplicar o modelo de estado ao veículo.
- (Opcional) `$stateTemplate.deactivateAfterSeconds` — O modelo de estado é automaticamente desativado após o tempo especificado. Esse parâmetro só pode ser usado quando o valor do `$stateTemplate.operation` parâmetro é “ativar”. Se esse parâmetro não for especificado ou se o valor desse parâmetro for 0, o Edge Agent continuará enviando atualizações de sinal para a nuvem até que uma operação de “desativação” seja recebida para o modelo de estado. O modelo de estado nunca é desativado automaticamente.

  Valor mínimo: 0, valor máximo: 4294967295.

> **Note**
>
> - A API retorna o sucesso em resposta a uma solicitação de ativação de um modelo que já está no estado ativo.
> - A API retorna o sucesso em resposta a uma solicitação de desativação de um modelo que já está no estado de desativação.

- A solicitação mais recente que você faz em um modelo de estado é aquela que entra em vigor. Por exemplo, se você solicitar a desativação de um modelo de estado em uma hora e, em seguida, fizer uma segunda solicitação para que o mesmo modelo seja desativado em quatro horas, a desativação de quatro horas entrará em vigor por ser a solicitação mais recente.

Important

Uma exceção de validação pode ocorrer em qualquer um dos seguintes cenários:

- É fornecido um modelo de estado que não `ASSOCIATED` está em um veículo.
- É feita uma solicitação para ativar um modelo de estado, mas ele não está `DEPLOYED` em um veículo.
- Uma solicitação é feita para um modelo de estado, mas está `DELETED` em um veículo.

Exemplo: ativar um modelo de estado

Para ativar um modelo de estado, primeiro crie um recurso de comando. Em seguida, você pode enviar o seguinte comando para o veículo no qual deseja ativar o modelo de estado. Este exemplo mostra como você pode especificar valores padrão para os parâmetros ao criar um comando. Esses parâmetros e seus valores são usados ao iniciar a execução do comando para ativar o modelo de estado.

1. Criar um recurso de comando

   Antes de enviar um comando para o veículo, você deve criar um recurso de comando. Você pode especificar valores alternativos para os parâmetros obrigatórios ao enviar o comando para o veículo. Para obter mais informações, consulte Criar um recurso de comando.

   Important

   `$stateTemplate.namee $stateTemplate.operation` os parâmetros devem ser fornecidos como um tipo de dados de string. Se qualquer outro tipo de dados for fornecido ou se algum desses dois parâmetros estiver ausente, a execução do comando falhará com uma exceção de validação. O

`$stateTemplate.deactivateAfterSeconds` parâmetro deve ser fornecido como um tipo de Long dados.

```
aws iot create-command \
    --description "This command activates a state template on a vehicle"
    --command-id ActivateStateTemplate \
    --display-name "Activate State Template" \
    --namespace AWS-IoTFleetWise \
    --mandatory-parameters '[
    {
        "name": "$stateTemplate.name",
        "defaultValue": {"S": "ST123"}
    },
    {
        "name": "$stateTemplate.operation",
        "defaultValue": {"S": "activate"}
    },
    {
        "name": "$stateTemplate.deactivateAfterSeconds",
        "defaultValue": {"L": "120"}
    }
]'
```

2. Inicie a execução do comando no veículo

Depois que o comando for criado, envie o comando para o veículo. Se você não especificou valores para os parâmetros obrigatórios ao criar o recurso de comando, você deve especificá-los agora. Para obter mais informações, consulte Enviar um comando (AWS CLI).

Important

Certifique-se de usar o endpoint da API do plano de dados de AWS IoT trabalhos específico da conta para a operação da API.

```
aws iot-jobs-data start-command-execution \
    --endpoint-url <endpoint-url> \
    --command-arn arn:aws:iot:region:111122223333:command/ActivateStateTemplate \
    --target-arn arn:aws:iot:region:111122223333:thing/<VEHICLE_NAME>
```

3. Recupere o status da operação do modelo de estado

   Depois de iniciar a execução do comando, você pode usar a `GetCommandExecution` API para recuperar o modelo de estado.

   ```
   aws iot get-command-execution --execution-id <EXECUTION_ID>
   ```

Exemplo: desativar um modelo de estado

Para desativar um modelo de estado, primeiro crie um recurso de comando. Em seguida, você pode enviar o seguinte comando para o veículo no qual deseja desativar o modelo de estado. Este exemplo mostra como você pode especificar valores padrão para os parâmetros ao criar um comando. Esses parâmetros e seus valores são usados ao iniciar a execução do comando para desativar o modelo de estado.

1. Criar um recurso de comando

   Antes de enviar um comando para o veículo, você deve criar um recurso de comando. Você pode especificar valores alternativos para os parâmetros obrigatórios ao enviar o comando para o veículo. Para obter mais informações, consulte Criar um recurso de comando.

   ```
   aws iot create-command \
       --description "This command deactivates a state template on a vehicle"
       --command-id DeactivateStateTemplate \
       --display-name "Deactivate State Template" \
       --namespace AWS-IoTFleetWise \
       --mandatory-parameters '[
       {
           "name": "$stateTemplate.name",
           "defaultValue": {"S": "ST123"}
       },
       {
           "name": "$stateTemplate.operation",
           "defaultValue": {"S": "deactivate"}
       }
   ]'
   ```

2. Inicie a execução do comando no veículo

   Depois que o comando for criado, envie o comando para o veículo. Se você não especificou valores para os parâmetros obrigatórios ao criar o recurso de comando, você deve especificá-los agora. Para obter mais informações, consulte Enviar um comando (AWS CLI).

   ```
   aws iot-jobs-data start-command-execution \
       --endpoint-url <endpoint-url> \
       --command-arn arn:aws:iot:region:111122223333:command/DeactivateStateTemplate \
       --target-arn arn:aws:iot:region:111122223333:thing/<VEHICLE_NAME>
   ```

3. Recupere o status da operação do modelo de estado

   Depois de iniciar a execução do comando, você pode usar a `GetCommandExecution` API para recuperar o modelo de estado.

   ```
   aws iot get-command-execution  --execution-id <EXECUTION_ID>
   ```

Obtenha um instantâneo do estado do veículo usando modelos de estado

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Você pode obter o último estado conhecido de um veículo usando a operação da API do plano de `CreateCommand` AWS IoT Core controle ou o console de AWS FleetWise IoT.

> **Important**
>
> Uma exceção de validação pode ocorrer em qualquer um dos seguintes cenários:
>
> - É fornecido um modelo de estado que não `ASSOCIATED` está em um veículo.
> - É feita uma solicitação para ativar um modelo de estado, mas ele não está `DEPLOYED` em um veículo.

- Uma solicitação é feita para um modelo de estado, mas está `DELETED` em um veículo.

Obter um instantâneo do estado do veículo (console)

Você pode usar o FleetWise console de AWS IoT para buscar o último estado conhecido de um veículo. AWS A IoT FleetWise criará um comando para você buscar dados.

Para obter o estado de um veículo

1. Abra o console AWS de IoT. FleetWise
2. No painel de navegação, selecione Veículos.
3. Escolha um veículo na lista para abrir sua página de detalhes.
4. Na guia Modelos de estado, escolha Buscar dados.
5. Selecione a função do IAM que concede FleetWise permissões de AWS IoT para enviar um comando e buscar dados. Consulte Controle de acesso.
6. Escolha Buscar estado.

Obtenha um instantâneo do estado do veículo ()AWS CLI

Para obter um instantâneo do estado, primeiro crie um recurso de comando. Em seguida, você pode enviar o seguinte comando para o veículo para o qual deseja obter o instantâneo do estado. Para obter mais informações sobre como usar a `CreateCommand` API e seus parâmetros, consulteUsar a API `CreateCommand`.

1. Criar um recurso de comando

   O exemplo a seguir mostra como criar o recurso de comando para realizar a operação de busca. Você pode especificar valores alternativos para os parâmetros obrigatórios ao enviar o comando para o veículo. Para obter mais informações, consulte Criar um recurso de comando.

   ```
   aws iot create-command \
       --command-id <COMMAND_ID> \
       --display-name "FetchSnapshot State Template" \
       --namespace AWS-IoTFleetWise \
       --mandatory-parameters '[
         {
             "name": "$stateTemplate.name",
   ```

```
            "defaultValue": {"S": "ST123"}
        },
        {
            "name": "$stateTemplate.operation",
            "defaultValue": {"S": "fetchSnapshot"}
        }
    ]'
```

Resposta:

```
{
    "commandId": "<COMMAND_ID>",
    "commandArn": "arn:aws:iot:<REGION>:111122223333:command/<COMMAND_ID>"
}
```

2. Inicie a execução do comando para obter um instantâneo do estado

   Depois que o comando for criado, envie o comando para o veículo. Se você não especificou valores para os parâmetros obrigatórios ao criar o recurso de comando, você deve especificá-los agora. Para obter mais informações, consulte Enviar um comando (AWS CLI).

   ```
   aws iot-jobs-data start-command-execution \
       --command-arn arn:aws:iot:region:111122223333:command/<COMMAND_ID> \
       --target-arn arn:aws:iot:region:111122223333:thing/<VEHICLE_NAME>
   ```

   Resposta:

   ```
   {
       "executionId": "<UNIQUE_UUID>"
   }
   ```

3. Recupere o status da operação do modelo de estado

   Depois de iniciar a execução do comando, você pode usar a `GetCommandExecution` API para recuperar o modelo de estado.

   ```
   aws iot get-command-execution --execution-id <EXECUTION_ID>
   ```

Processe os últimos dados do veículo estadual conhecido usando mensagens MQTT

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Para receber atualizações do seu veículo e processar seus dados, assine o seguinte tópico do MQTT. Para obter mais informações, consulte MQTT topics no Guia do desenvolvedor do AWS IoT Core .

```
$aws/iotfleetwise/vehicles/$vehicle_name/last_known_state/$state_template_name/data
```

As últimas mensagens de atualização do sinal de estado conhecido podem ser recebidas fora de ordem, pois o MQTT não garante o pedido. Qualquer cliente que use o MQTT para receber e processar dados do veículo deve lidar com isso. As últimas mensagens de atualização do sinal de estado conhecido seguem o protocolo de mensagens MQTT 5.

O cabeçalho de cada mensagem MQTT tem as seguintes propriedades de usuário:

- Nome do veículo — Um identificador exclusivo dos veículos.
- stateTemplateName— Um identificador exclusivo do último modelo de estado conhecido.

Além disso, você pode especificar os atributos do veículo a serem incluídos no cabeçalho da mensagem do MQTT especificando o parâmetro da `metadataExtraDimensions` solicitação ao atualizar ou criar um modelo de estado. (Consulte Modelos de estado.)

As propriedades do usuário no cabeçalho da mensagem MQTT são úteis para rotear mensagens para destinos diferentes sem inspecionar a carga.

A carga útil da mensagem MQTT contém dados coletados dos veículos. Você pode especificar os atributos do veículo a serem incluídos na carga de mensagens do MQTT especificando o parâmetro de `extraDimensions` solicitação ao criar ou atualizar um modelo de estado (consulte). Crie um modelo de AWS estado de IoT FleetWise As dimensões extras enriquecem os dados coletados dos veículos ao associar dimensões extras a eles.

A carga útil da mensagem MQTT é codificada em buffers de protocolo (Protobuf) e o cabeçalho da mensagem MQTT contém um indicador de tipo de conteúdo definido como application/octet-stream. O esquema de codificação do Protobuf é o seguinte:

```
// Copyright Amazon.com, Inc. or its affiliates. All Rights Reserved.
// SPDX-License-Identifier: Apache-2.0

syntax = "proto3";

option java_package = "com.amazonaws.iot.autobahn.schemas.lastknownstate";
package Aws.IoTFleetWise.Schemas.CustomerMessage;

message LastKnownState {

  /*
   * The absolute timestamp in milliseconds since Unix Epoch of when the event was
 triggered in vehicle.
   */
  uint64 time_ms = 1;

  /*
   * This field is deprecated, use signals instead
   */
  repeated Signal signal = 2 [ deprecated = true ];

  repeated Signal signals = 3;

  repeated ExtraDimension extra_dimensions = 4;
}

message Signal {

  /*
   * The Fully Qualified Name of the signal is the path to the signal plus the signal's
 name.
   * For example, Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  /*
   * The FWE reported signal value can be one of the following data types.
```

```
   */
  oneof SignalValue {
    double double_value = 2;

    bool boolean_value = 3;

    sint32 int8_value = 4;

    uint32 uint8_value = 5;

    sint32 int16_value = 6;

    uint32 uint16_value = 7;

    sint32 int32_value = 8;

    uint32 uint32_value = 9;

    sint64 int64_value = 10;

    uint64 uint64_value = 11;

    float float_value = 12;
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
    string string_value = 13;
  }
}

message ExtraDimension {
  /*
   * The Fully Qualified Name of the attribute is the path to the attribute plus the
 attribute's name.
   * For example, Vehicle.Model.Color
   * The fully qualified name can have up to 150 characters. Valid characters: a-z, A-
Z, 0-9, : (colon), and _ (underscore).
   */
  string name = 1;

  oneof ExtraDimensionValue {
    /*
     * An UTF-8 encoded or 7-bit ASCII string
     */
```

```
    string string_value = 2;
  }
}
```

Em que:

- `time_ms`:

  O registro de data e hora absoluto (em milissegundos desde o Unix Epoch) de quando o evento foi acionado no veículo. O software Edge Agent usa o relógio do veículo para esse registro de data e hora.
- `signal`:

  Uma matriz de `Signal` s que contém as informações do sinal: `name` (string) e `signalValue` que suporta os seguintes tipos de dados - `doublebool`,`int8`,`uint8`,,`int16`,`uint16`,`int32`,`uint32`,`int64`,`uint64`,`float`,`string`.
- `extra_dimensions`:

  Uma matriz `ExtraDimensions` que contém informações de atributos do veículo: `name` (string) e `extraDimensionValue` que atualmente suporta apenas o tipo de `string` dados.

Tutorial: Configurar a coleta de dados independente da rede usando uma interface de decodificação personalizada

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Introdução

Este tutorial descreve como configurar a AWS FleetWise IoT para coletar dados e executar comandos usando a coleta de dados independente de rede, que utiliza uma interface de decodificação personalizada. Com a coleta de dados independente da rede, você pode usar seus próprios métodos para decodificar sinais antes de enviá-los ao destino de dados especificado. Isso economiza tempo, pois você não precisa criar decodificadores de sinal especificamente para AWS IoT FleetWise. Você pode decodificar um subconjunto de sinais usando sua própria implementação ou pode usá-lo `defaultForUnmappedSignals` ao criar ou atualizar um manifesto do decodificador. Isso também oferece flexibilidade para coletar sinais e gatilhos em uma ampla variedade de fontes dentro do veículo.

Este tutorial é destinado a sinais de veículos que não estão em uma interface padrão de rede de área de controle (barramento CAN). Por exemplo, dados codificados em um formato ou esquema personalizado no veículo.

Configuração do ambiente

Este tutorial pressupõe que você tenha seguido as etapas para configurar seus ambientes para acessar a nuvem de AWS FleetWise IoT e a APIs implementação e a base de código do Edge.

Modelos de dados

A próxima seção ilustra como modelar as propriedades do veículo usando uma interface de decodificação personalizada. Isso se aplica à coleta de dados, bem como aos casos de uso de

comandos. Também se aplica a qualquer modelagem de fonte de dados subjacente usada no veículo, por exemplo, IDLs.

No exemplo, há duas propriedades do veículo: um sensor do veículo (posição atual do veículo) para coletar e um atuador do veículo (ar condicionado) para controlar remotamente. Ambos são definidos neste esquema:

```
// Vehicle WGS84 Coordinates
double Latitude;
double Longitude;

// Vehicle AC
Boolean ActivateAC;
```

A próxima etapa é importar essas definições para a AWS IoT FleetWise usando a interface de decodificação personalizada. APIs

Atualizações do catálogo do Signal

Importe essas definições em seu catálogo de sinais. Se você já tem um catálogo de sinais na AWS IoT FleetWise , use a API de atualização diretamente. Se você não tiver um, primeiro crie um catálogo de sinais e depois chame a API de atualização.

Primeiro, você deve criar a representação VSS desses sinais do veículo. O VSS é usado como uma taxonomia para representar dados de veículos na IoT. AWS FleetWise Crie um arquivo json chamado 'vehicle-signals.json' com o seguinte conteúdo:

```
// vehicle-signals.json
// Verify that branches and nodes are unique in terms of fully qualified name
// in the signal catalog.
[
 {
    "branch": {
      "fullyQualifiedName": "Vehicle",
      "description": "Vehicle Branch"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.CurrentLocation",
      "description": "CurrentLocation"
```

```
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "description": "Latitude"
    }
  },
  {
    "sensor": {
      "dataType": "DOUBLE",
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "description": "Longitude"
    }
  },
  {
    "actuator": {
      "fullyQualifiedName": "Vehicle.ActivateAC",
      "description": "AC Controller",
      "dataType": "BOOLEAN"
    }
  }
]
```

Se você não tiver um catálogo de sinais instalado, precisará invocar: `create-signal-catalog`

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise create-signal-catalog \
        --name my-signal-catalog \
        --nodes "${VEHICLE_NODES}"
```

Se você já tem um catálogo de sinais, pode adicionar esses sinais usando a `update-signal-catalog` API:

```
VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise update-signal-catalog \
        --name my-signal-catalog \
        --nodes-to-add "${VEHICLE_NODES}"
```

Modelo e decodificador do veículo

Depois de inserir os sinais no catálogo de sinais, a próxima etapa é criar um modelo de veículo e instanciar esses sinais. Para isso, você usa o `create-model-manifest` `create-decoder-manifest` APIs e.

Primeiro, formate os nomes dos sinais que você deseja inserir no modelo do veículo:

```
# Prepare the signals for insertion into the vehicle model.
VEHICLE_NODES=`cat vehicle-signals.json`
VEHICLE_NODES=`echo ${VEHICLE_NODES} | jq -r ".[] | .actuator,.sensor
 | .fullyQualifiedName" | grep Vehicle\\.`
VEHICLE_NODES=`echo "${VEHICLE_NODES}" | jq -Rn [inputs]`
# This is how the vehicle model input looks.
echo $VEHICLE_NODES
# [ "Vehicle.CurrentLocation.Latitude",
#   "Vehicle.CurrentLocation.Longitude",
#   "Vehicle.ActivateAC" ]
# Create the vehicle model with those signals.
aws iotfleetwise create-model-manifest \
    --name my-model-manifest \
    --signal-catalog-arn arn:xxxx:signal-catalog/my-signal-catalog \
    --nodes "${VEHICLE_NODES}"

# Activate the vehicle model.
 aws iotfleetwise update-model-manifest \
    --name my-model-manifest --status ACTIVE
```

Agora, use a interface de decodificação personalizada para criar um manifesto do decodificador.

> **Note**
>
> Você só precisa criar interfaces e sinais de rede se quiser especificar algo personalizado IDs, o que não faz parte deste exemplo.
> Para obter informações sobre mapeamento de informações de decodificação quando o nome totalmente qualificado (FQN) difere da ID do sinal de decodificação personalizado, consulte o Guia do desenvolvedor do Edge Agent.

```
// Create a network interface that is of type : CUSTOM_DECODING_INTERFACE
// custom-interface.json
```

```
[
  {
    "interfaceId": "NAMED_SIGNAL",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  },
  {
    "interfaceId": "AC_ACTUATORS",
    "type": "CUSTOM_DECODING_INTERFACE",
    "customDecodingInterface": {
      "name": "NamedSignalInterface"
    }
  }
]
// custom-decoders.json
// Refer to the fully qualified names of the signals, make them of
// type CUSTOM_DECODING_SIGNAL, and specify them as part of the same interface ID
// that was defined above.
[
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Longitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Longitude"
      }
    },
    {
      "fullyQualifiedName": "Vehicle.CurrentLocation.Latitude",
      "interfaceId": "NAMED_SIGNAL",
      "type": "CUSTOM_DECODING_SIGNAL",
      "customDecodingSignal": {
        "id": "Vehicle.CurrentLocation.Latitude"
      }
    },
    {
        "fullyQualifiedName": "Vehicle.ActivateAC",
        "interfaceId": "AC_ACTUATORS",
        "type": "CUSTOM_DECODING_SIGNAL",
        "customDecodingSignal": {
          "id": "Vehicle.ActivateAC"
        }
```

```
    }
]
# Create the decoder manifest.
 CUSTOM_INTERFACE=`cat custom-interface.json`
 CUSTOM_DECODERS=`cat custom-decoders.json`

aws iotfleetwise create-decoder-manifest \
   --name my-decoder-manifest \
   --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
   --network-interfaces "${CUSTOM_INTERFACE}" \
   --signal-decoders "${CUSTOM_DECODERS}"

 # Activate the decoder manifest.
 aws iotfleetwise update-decoder-manifest \
    --name my-decoder-manifest \
    --status ACTIVE
```

Neste ponto, você modelou totalmente esses sinais na AWS IoT FleetWise. Em seguida, você cria o veículo e o associa ao modelo que você criou. Você usa a `create-vehicle` API para isso:

```
aws iotfleetwise create-vehicle \
    --decoder-manifest-arn arn:xxx:decoder-manifest/my-decoder-manifest \
    --association-behavior ValidateIotThingExists \
    --model-manifest-arn arn:xxx:model-manifest/my-model-manifest \
    --vehicle-name "my-vehicle"
```

A próxima etapa é focar na base de código do AWS IoT FleetWise Edge e escrever a extensão de código necessária.

> **Note**
>
> Para obter informações sobre a implementação do Edge, consulte o Guia do Desenvolvedor do Edge Agent.

Enviar comando

Agora, compile o software (certifique-se de adicionar seus cabeçalhos e arquivos C++ ao CMake arquivo) e depois volte para a nuvem APIs para testar um comando nesse atuador:

```
// Create a command targeting your vehicle.
```

```
aws iot create-command --command-id activateAC \
    --namespace "AWS-IoT-Fleetwise" \
    --endpoint-url endpoint-url \
    --role-arn ${SERVICE_ROLE_ARN} \
    --mandatory-parameters '[ { "name": "$actuatorPath.Vehicle.ActivateAC",
 "defaultValue": {"B": "false"} } ]' \
// You will receive the command ARN.

{
    "commandId": "activateAC",
    "commandArn": "arn:aws:iot:xxx:command/activateAC"
}

// You can send the command to activate the AC targeting your vehicle.

JOBS_ENDPOINT_URL=`aws iot describe-endpoint --endpoint-type iot:Jobs | jq -
j .endpointAddress`
aws iot-jobs-data start-command-execution \
    --command-arn arn:aws:iot:xxx:command/activateAC \
    --target-arn arn:xxx:vehicle/my-vehicle \
    --parameters '{ "$actuatorPath.Vehicle.ActivateAC" : {"B": "true"}}' \
    --endpoint-url https://${JOBS_ENDPOINT_URL}
// You will receive the corresponding execution ID.
{
    "executionId": "01HSK4ZH6ME7D43RB2BV8JC51D"
}

// If you have the AWS IoT FleetWise Edge Agent running, you can see the logs.
[AcCommandDispatcher.cpp:26] [setActuatorValue()]:
[Actuator Vehicle.ActivateAC executed successfully for command ID
 01HSK4ZH6ME7D43RB2BV8JC51D]
```

Use AWS CLI e AWS SDKs com a AWS IoT FleetWise

Esta seção fornece informações sobre como fazer solicitações de FleetWise API de AWS IoT. Para obter mais informações sobre FleetWise operações e tipos de dados de AWS IoT, consulte a Referência da API de AWS FleetWise IoT.

Para usar a AWS IoT FleetWise com uma variedade de linguagens de programação, use o AWS SDKs, que contém as seguintes funcionalidades automáticas:

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

Para acesso à linha de comando, use a AWS IoT FleetWise com o. AWS CLI Você pode controlar a AWS IoT FleetWise e seus outros serviços a partir da linha de comando e automatizá-los por meio de scripts.

Solução de problemas de AWS IoT FleetWise

Use as informações e soluções de solução de problemas nesta seção para ajudar a resolver problemas com a AWS IoT FleetWise.

As informações a seguir podem ajudá-lo a solucionar problemas comuns com a AWS IoT FleetWise.

Tópicos

- AWS Problemas manifestos do FleetWise decodificador de IoT
- Edge Agent para problemas AWS de software de IoT FleetWise
- Armazenar e encaminhar problemas

AWS Problemas manifestos do FleetWise decodificador de IoT

Solucione problemas com manifesto do decodificador.

Diagnosticar chamadas de API do manifesto do decodificador

Erro	Diretrizes de solução de problemas
`UpdateOperationFailure.ConflictingDecoderUpdate`	O mesmo manifesto do decodificador tem várias solicitações de atualização. Espere e tente novamente.
`UpdateOperationFailure.InternalFailure`	InternalFailure é lançado como uma exceção encapsulada. O problema em si depende da exceção encapsulada.
`UpdateOperationFailure.ActiveDecoderUpdate`	O manifesto do decodificador está em um estado `Active` e não pode ser atualizado. Altere o estado do manifesto do decodificador para `DRAFT` e tente novamente.
`UpdateOperationFailure.ConflictingModelUpdate`	AWS A IoT FleetWise está tentando se validar 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
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND`	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.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE`	Atualize o modelo do veículo para que ele fique no estado `ACTIVE` e tente novamente.
`UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_NOT_FOUND`	AWS A IoT não FleetWise consegue encontrar o modelo do veículo associado ao manifesto do decodificador. Verifique o nome do recurso da Amazon (ARN) do modelo do veículo e tente novamente.
`UpdateOperationFailure.ModelManifestValidationResponse (FailureReason.MODEL_DATA_ENTRIES_READ_FAILURE`	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.
`UpdateOperationFailure.ValidationFailure`	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.

Erro	Diretrizes de solução de problemas
`UpdateOperationFailure.KmsKeyAccessDenied`	Há um problema de permissão na tecla 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.
`UpdateOperationFailure.DecoderDoesNotExist`	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
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)`	AWS A 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.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)`	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
	correspondem à definição do catálogo de sinais correspondente.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)`	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.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	AWS A IoT FleetWise encontrou um sinal definido como STRUCT no catálogo de sinais sem um sinal structuredMessageDefinition definido na solicitação do manifesto do decodificador. Certifique-se de que cada estrutura esteja definida como a structuredMessageDefinition na solicitação de atualização do manifesto do decodificador.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	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 seu structFullyQualified nome de campo definido. Ele também precisa de um nó STRUCT com isso fullyQualifiedName.
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	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.

Erro	Diretrizes de solução de problemas
`InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)`	Um objeto de matriz no catálogo de sinais não foi definido como uma structuredMessageList definição na solicitação de atualização do manifesto do decodificador. Verifique se todas as propriedades da matriz estão definidas como structuredMessageList Definição na solicitação de atualização do manifesto do decodificador.

Edge Agent para problemas AWS de software de IoT FleetWise

Solucione problemas do software Edge Agent.

Problemas

- Problema: o software Edge Agent não inicia.
- Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [Falha ao iniciar a biblioteca de persistência]
- Problema: O software Edge Agent não coleta diagnósticos integrados (OBD) II PIDs e códigos de problemas de diagnóstico (). DTCs
- Problema: o FleetWise software Edge Agent para AWS IoT não coleta dados da rede ou não consegue aplicar regras de inspeção de dados.
- Problema: [ERROR] [AwsIotConnectivityModule: :connect]: [Falha na conexão com erro] ou [WARN] [AwsIotChannel: :send]: [Nenhuma conexão MQTT ativa.]

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 FleetWise software Edge Agent para AWS IoT está usando o 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 FleetWise software Edge Agent para AWS IoT.

1. Abra o console AWS de IoT. FleetWise
2. Na página inicial do serviço, na FleetWise seção Introdução à AWS IoT, escolha Explore 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: *network-interface-id*Substitua 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.

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

- `mqttConnection::privateKeyFilename`— O caminho para o arquivo da chave privada do veículo.
- Você costumava AWS IoT Core provisionar o veículo. Para obter mais informações, consulte Provisionar AWS veículos de IoT FleetWise.

Para obter mais informações, consulte Perguntas frequentes sobre o AWS IoT Device SDK for C++.

Problema: [ERROR] [Io TFleetWiseEngine: :connect]: [Falha ao iniciar a biblioteca de persistência]

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 diagnósticos integrados (OBD) II PIDs 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 seu veículo suportar a arbitragem estendida da Controller Area Network (CAN bus) IDs, verifique se `obdInterface:useExtendedIds` está configurado para`true`.

Problema: o FleetWise software Edge Agent para AWS IoT 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	Quota	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]: [Falha na conexão com erro] ou [WARN] [AwsIotChannel: :send]: [Nenhuma conexão MQTT ativa.]

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 cada minuto e espera por três minutos. Se não houver resposta, o software Edge Agent restabelecerá automaticamente a conexão com a nuvem.

Armazenar e encaminhar problemas

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

Problema: receber um **`AccessDeniedException`** com todas as permissões necessárias do IAM

Solução: o recurso de armazenamento e encaminhamento para particionamento de dados em campanhas requer acesso fechado por meio da listagem de permissões. Entre em contato com a equipe de serviço para garantir que seus recursos tenham permissões adequadas por meio da listagem de permissões.

Problema: os dados enviados para AWS IoT Jobs ignoram o **`endTime`**

Solução: Você especificou um inválido `endtime` no documento do trabalho. Por exemplo, `endtime` não segue o formato ISO 8601 UTC). Nos registros do AWS IoT FleetWise agente, pode haver uma declaração em nível de aviso que diz,. `Malformed IoT Job endTime:` **`customer configured endTime.`** `Not setting endTime`

Problema: o upload de dados para AWS IoT Jobs tem um status de **`REJECTED`** execução.

Solução: Você especificou um inválido `campaignArn` no documento do trabalho. Por exemplo, se você especificar um ARN para uma campanha que não está sendo executada em um veículo, pode haver uma declaração de nível de erro que diz, `CampaignArn value in the received job document does not match the ARN of a Store and Forward campaign` nos registros do agente. AWS IoT FleetWise

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
- AWS atualizações de políticas gerenciadas para o 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 o conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para saber mais sobre a privacidade de dados, consulte as Data Privacy FAQ. Para saber mais sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and RGPD no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com Centro de Identidade do AWS IAM 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 AWS os recursos. Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail. Para obter informações sobre o uso de CloudTrail trilhas para capturar AWS atividades, consulte Como trabalhar com CloudTrail trilhas no Guia AWS CloudTrail do usuário.
- 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 sensíveis armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-3 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para saber mais sobre os endpoints FIPS disponíveis, consulte Federal Information Processing Standard (FIPS) 140-3.

É altamente recomendável que nunca sejam colocadas informações confidenciais ou sensíveis, como endereços de e-mail de clientes, em tags ou campos de formato livre, como um campo Nome.

Isso inclui quando você trabalha com a AWS IoT FleetWise ou outra coisa Serviços da AWS usando o console, a API ou. AWS CLI AWS SDKs Quaisquer dados inseridos em tags ou em 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, é fortemente recomendável que não sejam incluídas informações de credenciais no URL para validar a solicitação nesse 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 em repouso na AWS IoT FleetWise

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. Para obter mais informações, consulte Proteção de dados com criptografia no Guia do usuário do Amazon S3 e Proteção de dados no Timestream for. LiveAnalytics

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 obter mais informações, consulte the section called “Provisionar veículos”.

Criptografia de dados na AWS IoT FleetWise

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 que estão hospedadas em FleetWise gateways de AWS IoT e podem ser acessadas pela rede local. Eles APIs são expostos por meio de uma conexão TLS apoiada por um certificado de servidor de propriedade do conector IoT AWS Edge. FleetWise Para autenticação do cliente, eles APIs usam 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 em repouso na AWS IoT FleetWise
- Gerenciamento de chaves na AWS IoT FleetWise

Criptografia em repouso na AWS IoT FleetWise

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 esta chave sem custo adicional.
- Chave gerenciada pelo cliente – A chave é armazenada na sua conta e é você que a cria, detém 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 as chaves KMS, como as seguintes:

- Estabelecer e manter políticas de chaves, políticas do IAM e concessões
- Ativação e desativação das chaves

- Alternar os materiais de criptografia das chaves
- Adicionar tags da
- Criar aliases que se referem as chaves
- 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. AWS A 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 usar sua AWS KMS chave.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "Allow use of the key",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
  ]
}
```

> **Important**
>
> Ao adicionar as novas seções à sua política de chaves do KMS, não altere nenhuma seção existente na política. AWS A 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 GetEncryptionConfigurationda 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 PutEncryptionConfigurationAPI para redefinir o tipo de criptografia para`FLEETWISE_DEFAULT_ENCRYPTION`.

   ```
   aws iotfleetwise put-encryption-configuration \
         --encryption-type FLEETWISE_DEFAULT_ENCRYPTION
   ```

3. Use a operação PutEncryptionConfigurationda API para reativar o tipo de criptografia paraKMS_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 na AWS IoT FleetWise.

Gerenciamento de chaves na AWS IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

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 dos seguintes recursos.

AWS Recurso de IoT FleetWise	Tipo de dados	Campos criptografados em repouso com chaves gerenciadas pelo cliente
Catálogo de sinais		descrição
	Atributo	description, allowedValues, defaultValue, min, max
	Actuator	description, allowedValues, min, max

AWS Recurso de IoT FleetWise	Tipo de dados	Campos criptografados em repouso com chaves gerenciadas pelo cliente
	Sensor	description, allowedValues, min, max
Modelo do veículo (manifesto do modelo)		descrição
Manifesto do decodificador		descrição
	CanInterface	protocolName, protocolVersion
	ObdInterface	requestMessageId, dtcRequestInterval Segundos hasTransmissionEcu, Padrão OBD, Segundos, pidRequestInterval useExtendedIds
	CanSignal	factor,, isSigned isBigEndian, comprimento, messageID, offset, startBit
	ObdSignal	ByteLength, offset, pid, escalabilidade, modo de serviço pidResponseLength, startByte,, bitMaskLength bitRightShift
Veículo		attributes
Campaign		descrição
	conditionBasedCollectionEsquema	expressão, minimumTriggerInterval Ms conditionLanguageVersion, TriggerMode
	TimeBasedCollectionScheme	periodMs
Modelo de estado		descrição

> **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 — Escolha 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 PutEncryptionConfigurationAPI 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.

- *kms_key_id*Substitua pela ID da chave KMS.

```
aws iotfleetwise put-encryption-configuration \
      --encryption-type KMS_BASED_ENCRYPTION \
      --kms-key-id kms_key_id
```

Example resposta

```
{
 "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 O principal FleetWise serviço de AWS IoT `iotfleetwise.amazonaws.com` na declaração de política de chaves do KMS permite que a AWS IoT acesse FleetWise a chave do KMS.

```
{
  "Sid": "Allow FleetWise to encrypt and decrypt data when customer managed KMS key
 based encryption is enabled",
```

```
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*"
}
```

Como prática recomendada de segurança, adicione `aws:SourceArn` e `aws:SourceAccount` condicione as chaves à política de chaves do KMS. A chave de condição global do IAM `aws:SourceArn` ajuda a garantir que a AWS IoT FleetWise use a chave KMS somente para o recurso Amazon Resource Names () específico do serviço. ARNs

Se você definir o valor de`aws:SourceArn`, ele sempre deverá ser`arn:aws:iotfleetwise:us-east-1:account_id:*`. Isso permite que a chave KMS acesse todos os recursos de AWS FleetWise IoT para isso. Conta da AWS AWS A IoT FleetWise oferece suporte a uma chave KMS por conta para todos os recursos nela contidos. Região da AWS Usar qualquer outro valor para ou não usar o `SourceArn` caractere curinga (*) para o campo de recurso ARN impede que a AWS FleetWise IoT acesse a chave KMS.

O valor de `aws:SourceAccount` é o ID da sua conta, que é usado para restringir ainda mais a chave KMS para que ela só possa ser usada para sua conta específica. Se você adicionar `aws:SourceAccount` e `aws:SourceArn` condicionar chaves à chave KMS, certifique-se de que a chave não seja usada por nenhum outro serviço ou conta. Isso ajuda a evitar falhas.

A política a seguir inclui um principal de serviço (um identificador para um serviço), bem como `aws:SourceAccount` uma `aws:SourceArn` configuração para uso com base no Região da AWS e na ID da sua conta.

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
```

```
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "aws:SourceAccount": "AWS-account-ID"
    },
    "ArnLike": {
      "aws:SourceArn": "arn:aws:iotfleetwise:region:AWS-account-ID:*"
    }
  }
}
```

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. AWS A 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.

Permissões para AWS KMS criptografia

Se você habilitou a AWS KMS criptografia, deverá especificar permissões na política de função para poder chamar a AWS IoT FleetWise APIs. A política a seguir permite acesso a todas as FleetWise ações de AWS IoT, bem como permissões AWS KMS específicas.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:*",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

A declaração de política a seguir é necessária para que sua função invoque a criptografia APIs. Esta declaração de política permite `GetEncryptionConfiguration` ações `PutEncryptionConfiguration` e ações da AWS IoT FleetWise.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:GetEncryptionConfiguration",
        "iotfleetwise:PutEncryptionConfiguration",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
```

```
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Recuperação após a exclusão da AWS KMS chave

Se você excluir uma AWS KMS chave depois de ativar a criptografia com a AWS IoT FleetWise, deverá redefinir sua conta excluindo todos os dados antes de usar a IoT AWS novamente. FleetWise Você pode usar a lista e excluir as operações da API para limpar os recursos da sua conta.

Para limpar recursos em sua conta

1. Use a lista APIs com o `listResponseScope` parâmetro definido como`METADATA_ONLY`. Isso fornece uma lista de recursos, incluindo nomes de recursos e outros metadados, como registros de data ARNs e hora.
2. Use delete APIs para remover recursos individuais.

Você deve limpar os recursos na seguinte ordem.

1. Campanhas
   a. Liste todas as campanhas com o `listResponseScope` parâmetro definido como`METADATA_ONLY`.
   b. Exclua as campanhas.
2. Frotas e veículos
   a. Liste todas as frotas com o `listResponseScope` parâmetro definido como. `METADATA_ONLY`
   b. Liste todos os veículos de cada frota com o `listResponseScope` parâmetro definido como`METADATA_ONLY`.
   c. Desassocie todos os veículos de cada frota.
   d. Exclua as frotas.

   e. Exclua os veículos.
3. Manifestos de decodificadores
   a. Liste todos os manifestos do decodificador com o `listResponseScope` parâmetro definido como. `METADATA_ONLY`
   b. Exclua todos os manifestos do decodificador.
4. Modelos de veículos (manifestos de modelos)
   a. Liste todos os modelos de veículos com o `listResponseScope` parâmetro definido como`METADATA_ONLY`.
   b. Exclua todos os modelos de veículos.
5. Modelos de estado
   a. Liste todos os modelos de estado com o `listResponseScope` parâmetro definido como`METADATA_ONLY`.
   b. Exclua todos os modelos de estado.
6. Catálogos de sinais
   a. Liste todos os catálogos de sinais.
   b. Exclua todos os catálogos de sinais.

Controlando o acesso com AWS IoT FleetWise

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na 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 ou às mensagens MQTT usadas para enviar dados de veículos.

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 permissão para enviar e receber dados sobre um tópico do MQTT
- Conceder AWS IoT FleetWise acesso a um destino do Amazon S3
- Conceda AWS IoT FleetWise acesso a um destino do Amazon Timestream
- Conceda AWS IoT Device Management permissão para gerar a carga útil para comandos com AWS IoT FleetWise

Conceder AWS IoT FleetWise permissão para enviar e receber dados sobre um tópico do MQTT

Quando você usa um tópico MQTT, seus veículos enviam dados usando o agente de mensagens AWS IoT MQTT. Você deve conceder AWS IoT FleetWise permissão para assinar o tópico do MQTT que você especificar. Se você também usa AWS IoT regras para agir ou rotear dados para outros destinos, deve anexar políticas a uma função do IAM para permitir o encaminhamento de dados AWS IoT FleetWise para as regras de IoT.

Além disso, seus outros aplicativos ou dispositivos podem se inscrever no tópico que você especificar para receber dados do veículo quase em tempo real, e esses aplicativos ou dispositivos devem receber permissões e acesso conforme necessário.

Para obter mais informações sobre o uso do MQTT e as funções e permissões necessárias, consulte:

- Protocolos de comunicação do dispositivo
- Regras para AWS IoT
- Conceder a uma AWS IoT regra o acesso que ela exige
- Passe permissões de função

Antes de começar, verifique o seguinte:

> **Important**
>
> - Você deve usar a mesma AWS região ao criar recursos de campanha de veículos para AWS IoT FleetWise. Se você mudar de AWS região, poderá ter problemas para acessar os recursos.
> - AWS A IoT FleetWise está disponível no Leste dos EUA (Norte da Virgínia) e na Europa (Frankfurt).

Você pode usar o AWS CLI para criar uma função do IAM com uma política de confiança para mensagens MQTT. Para criar um perfil do IAM, execute o comando a seguir.

Para criar um perfil do IAM com uma política de confiança

- *IotTopicExecutionRole*Substitua pelo nome da função que você está criando.
- *trust-policy*Substitua pelo arquivo JSON que contém a política de confiança.

```
aws iam create-role --role-name IotTopicExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "mqttTopicTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleetwise.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn": [
            "arn:aws:iotfleetwise:region:123456789012:campaign/campaign-name"
          ],
          "aws:SourceAccount": [
```

```
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

Crie uma política de permissões para dar FleetWise permissões à AWS IoT para publicar mensagens no tópico do MQTT que você especificou. Para criar uma política de permissões, execute o seguinte comando.

Para criar uma política de permissões

- *AWSIoTFleetwiseAccessIotTopicPermissionsPolicy*Substitua pelo nome da política que você está criando.
- *permissions-policy*Substitua pelo nome do arquivo JSON que contém a política de permissões.

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessIotTopicPermissionsPolicy --
policy-document file://permissions-policy.json
```

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Action": [
          "iot:Publish"
        ],
        "Resource": [
          "topic-arn"
        ]
      }
    ]
}
```

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.
   - *IotTopicExecutionRole*Substitua pelo nome da função do IAM que você criou.

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-
name IotTopicExecutionRole
```

Para obter mais informações, consulte Gerenciamento de acesso para recursos do AWS no Guia do usuário do IAM.

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 uma 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.

JSON

```
{
  "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": "123456789012"
        }
      }
    }
  ]
}
```

A política de bucket a seguir se aplica a todas as campanhas em uma conta em uma região da AWS .

JSON

```
{
  "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:123456789012:campaign/*",
          "aws:SourceAccount": "123456789012"
        }
      }
    }
  ]
}
```

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 obter mais informações, consulte Desativação ACLs de todos os novos buckets e imposição da propriedade de objetos no Guia do usuário do Amazon Simple Storage Service.

Conceda 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.

> **Note**
>
> O Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

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), Europa (Frankfurt) e Ásia-Pacífico (Mumbai).
> - 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 Amazon Timestream Developer Guide.
- Você deve ter uma tabela criada no banco de dados do Timestream especificado. Para ver um tutorial, consulte Criar uma tabela no Amazon Timestream Developer Guide.

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.
- *trust-policy*Substitua pelo arquivo.json que contém a política de confiança.

```
aws iam create-role --role-name TimestreamExecutionRole --assume-role-policy-document
 file://trust-policy.json
```

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:123456789012:campaign/campaign-name"
           ],
           "aws:SourceAccount": [
            "123456789012"
          ]
        }
      }
    }
  ]
}
```

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.
- *permissions-policy*Substitua pelo nome do arquivo JSON que contém a política de permissões.

```
aws iam create-policy --policy-name AWSIoTFleetwiseAccessTimestreamPermissionsPolicy --
policy-document file://permissions-policy.json
```

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.

Conceda AWS IoT Device Management permissão para gerar a carga útil para comandos com AWS IoT FleetWise

Quando você usa o recurso de comandos para iniciar a execução de um comando, AWS IoT Device Management buscará o comando e os parâmetros do comando da solicitação recebida. Em seguida,

ele exige permissões para acessar os FleetWise recursos de AWS IoT para validar a solicitação e gerar a carga útil. A carga útil é então enviada ao veículo AWS IoT Device Management por meio do MQTT para o tópico de solicitação de comando no qual seu veículo se inscreveu.

Primeiro, você deve criar uma função do IAM que AWS IoT Device Management conceda as permissões necessárias para gerar a carga. Em seguida, forneça o ARN dessa função à `CreateCommand`API usando o `roleArn` campo. Veja a seguir alguns exemplos de políticas.

> **Important**
>
> Para a função do IAM, você deve usar a Região da AWS mesma em que criou o veículo e os recursos de comando. Se você mudar Região da AWS, poderá ter problemas para acessar os recursos.

A função do IAM precisa ter a seguinte política de confiança.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "RemoteCommandsTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Conceda permissões a todos os veículos (coisas de IoT)

O exemplo a seguir mostra como conceder permissões para gerar a carga útil para todos os veículos registrados como AWS IoT itens.

> **Note**
>
> - Essa política pode ser excessivamente permissiva. Use o princípio do privilégio mínimo para garantir que você conceda somente as permissões necessárias.
> - Em vez disso, para negar permissões, altere `"Effect": "Allow"` para `"Effect": "Deny"` a política do IAM.

Neste exemplo, substitua:

- *region*com você Região da AWS onde você está usando os recursos de AWS IoT. FleetWise
- *111122223333*com seu Conta da AWS número.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        }
    ]
}
```

Conceda permissão a um veículo específico (coisa de IoT)

O exemplo a seguir mostra como conceder permissões para gerar a carga útil de um veículo específico registrado como uma AWS IoT coisa.

Neste exemplo, substitua:

- *region*com você Região da AWS onde você está usando os recursos de AWS IoT. FleetWise
- *111122223333*com seu Conta da AWS número.
- *<VEHICLE_NAME>*com o nome IoT do seu veículo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:us-east-1:111122223333:thing/<VEHICLE_NAME>"
        }
    ]
}
```

Conceda permissões para veículos e sinais específicos

O exemplo a seguir mostra como conceder permissões para gerar a carga útil do atuador para um veículo específico.

Neste exemplo, substitua:

- *region*com você Região da AWS onde você está usando os recursos de AWS IoT. FleetWise
- *111122223333*com seu Conta da AWS número.
- *<VEHICLE_NAME>*com o nome IoT do seu veículo.
- *<SIGNAL_FQN>*com o nome do sinal, como*<Vehicle.actuator2>*.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "arn:aws:iot:us-
east-1:111122223333:thing/<VEHICLE_NAME>",
            "Condition": {
                "ForAnyValue:StringEquals": {
```

```
                    "iotfleetwise:Signals": [
                        "<SIGNAL_FQN>"
                    ]
                }
            }
        }
    ]
}
```

Conceda permissões para veículos e modelos de estado específicos

O exemplo a seguir mostra como conceder permissões para gerar a carga útil para um modelo específico de veículo e estado.

Neste exemplo, substitua:

- *region*é Região da AWS onde você está usando os recursos de AWS IoT. FleetWise
- *111122223333*é o seu Conta da AWS número.
- *<VEHICLE_NAME>*é o nome da IoT para seu veículo.
- *<STATE_TEMPLATE_ID>*com o identificador do seu modelo de estado.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": [
                "arn:aws:iot:us-east-1:111122223333:thing/<VEHICLE_NAME>",
                "arn:aws:iotfleetwise:us-east-1:111122223333:state-
template/<STATE_TEMPLATE_ID>"
            ]
        }
    ]
}
```

Conceda permissões para usar chaves KMS gerenciadas pelo cliente

Se você habilitou as chaves KMS gerenciadas pelo cliente para AWS IoT FleetWise, o exemplo a seguir mostra como conceder permissões para gerar a carga.

Neste exemplo, substitua:

- *region*com você Região da AWS onde você está usando os recursos de AWS IoT. FleetWise
- *111122223333*com seu Conta da AWS número.
- *<KMS_KEY_ID>*com o ID da sua chave KMS.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iotfleetwise:GenerateCommandPayload",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "kms:Decrypt",
            "Resource": "arn:aws:kms:us-east-1:111122223333:key/<KMS_KEY_ID>"
        }
    ]
}
```

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
- Autenticação com identidades
- Gerenciar o 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
- AWS Referência de ações e recursos FleetWise da API de IoT

Público

A forma como você usa AWS Identity and Access Management (IAM) difere com base na sua função:

- Usuário do serviço: solicite permissões ao seu administrador se você não conseguir acessar os atributos (consulte Solução de problemas AWS de FleetWise identidade e acesso à IoT).
- Administrador do serviço: determine o acesso do usuário e envie solicitações de permissão (consulte Como a AWS IoT FleetWise funciona com o IAM)
- Administrador do IAM: escreva políticas para gerenciar o acesso (consulte Exemplos de políticas baseadas em identidade para IoT AWS FleetWise)

Autenticação com identidades

A autenticação é como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado como usuário do IAM ou assumindo uma função do IAM. Usuário raiz da conta da AWS

Você pode fazer login como uma identidade federada usando credenciais de uma fonte de identidade como Centro de Identidade do AWS IAM (IAM Identity Center), autenticação de login único ou credenciais. Google/Facebook Para ter mais informações sobre como fazer login, consulte Como fazer login em sua Conta da AWS no Guia do usuário do Início de Sessão da AWS .

Para acesso programático, AWS fornece um SDK e uma CLI para assinar solicitações criptograficamente. Para ter mais informações, consulte AWS Signature Version 4 para solicitações de API no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar um Conta da AWS, você começa com uma identidade de login chamada usuário Conta da AWS raiz que tem acesso completo a todos Serviços da AWS os recursos. É altamente recomendável não usar o usuário-raiz em tarefas diárias. Consulte as tarefas que exigem credenciais de usuário-raiz em Tarefas que exigem credenciais de usuário-raiz no Guia do usuário do IAM.

Identidade federada

Como prática recomendada, exija que os usuários humanos 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 corporativo, provedor de identidade da web ou Directory Service que acessa Serviços da AWS usando credenciais de uma fonte de identidade. As identidades federadas assumem funções que oferecem credenciais temporárias.

Para o gerenciamento de acesso centralizado, recomendamos Centro de Identidade do AWS IAM. Para saber mais, consulte O que é o IAM Identity Center? no Guia do usuário do Centro de Identidade do AWS IAM .

Usuários e grupos do IAM

Um usuário do IAM é uma identidade com permissões específicas para uma única pessoa ou aplicação. É recomendável usar credenciais temporárias, em vez de usuários do IAM com credenciais de longo prazo. Para obter mais informações, consulte Exigir que usuários humanos usem a federação com um provedor de identidade para acessar AWS usando credenciais temporárias no Guia do usuário do IAM.

Um grupo do IAM especifica um conjunto de usuários do IAM e facilita o gerenciamento de permissões para grandes conjuntos de usuários. Para ter mais informações, consulte Casos de uso de usuários do IAM no Guia do usuário do IAM.

Perfis do IAM

Uma perfil do IAM é uma identidade com permissões específicas que oferece credenciais temporárias. Você pode assumir uma função mudando de um usuário para uma função do IAM (console) ou chamando uma operação de AWS API AWS CLI ou. Para saber mais, consulte Métodos para assumir um perfil no Manual do usuário do IAM.

Os perfis do IAM são úteis para acesso de usuário federado, permissões de usuário do IAM temporárias, acesso entre contas, acesso entre serviços e aplicações em execução no Amazon EC2. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Gerenciar o acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política define permissões quando associada a uma identidade ou recurso. AWS avalia essas políticas quando um diretor faz uma solicitação. A maioria das políticas é armazenada AWS como documentos JSON. Para ter mais informações sobre documentos de política JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Por meio de políticas, os administradores especificam quem tem acesso a que, definindo qual entidade principal pode realizar ações em quais recursos e sob quais condições.

Por padrão, usuários e perfis não têm permissões. Um administrador do IAM cria políticas do IAM e as adiciona aos perfis, os quais os usuários podem então assumir. As políticas do IAM definem permissões, independentemente do método usado para realizar a operação.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissão JSON que você anexa a uma identidade (usuário, grupo ou perfil). Essas políticas controlam quais ações as identidades podem realizar, em quais recursos e sob quais condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

As políticas baseadas em identidade podem ser políticas em linha (incorporadas diretamente em uma única identidade) ou políticas gerenciadas (políticas autônomas anexadas a várias identidades). Para saber como escolher entre uma política gerenciada e políticas 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. Entre os exemplos estão políticas de confiança de perfil do IAM e 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. É necessário especificar uma entidade principal em uma política baseada em recursos.

Políticas baseadas em recursos são políticas em linha localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais que podem definir o máximo de permissões concedidas por tipos de políticas mais comuns:

- Limites de permissões: definem o número máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM. Para saber mais sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — Especifique as permissões máximas para uma organização ou unidade organizacional em AWS Organizations. Para saber mais, consulte Políticas de controle de serviço no Guia do usuário do AWS Organizations .
- Políticas de controle de recursos (RCPs) — Defina o máximo de permissões disponíveis para recursos em suas contas. Para obter mais informações, consulte Políticas de controle de recursos (RCPs) no Guia AWS Organizations do usuário.
- Políticas de sessão: políticas avançadas transmitidas como um parâmetro durante a criação de uma sessão temporária para um perfil ou um usuário federado. Para saber mais, 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

Recurso do IAM	AWS Suporte à IoT FleetWise
Políticas baseadas em identidade	Sim
Políticas baseadas em recurso	Não
Ações de políticas	Sim
Recursos de políticas	Sim
Chaves de condição de políticas	Sim
ACLs	Não
ABAC (tags em políticas)	Parcial
Credenciais temporárias	Sim
Permissões de entidade principal	Sim
Perfis de serviço	Não
Funções vinculadas ao serviço	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

Compatível com políticas baseadas em identidade: sim

As políticas baseadas em identidade são documentos de políticas de permissões JSON que podem ser anexados a uma identidade, como usuário do IAM, grupo de usuários ou perfil. Essas políticas controlam quais ações os usuários e perfis podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Definir permissões personalizadas do IAM com as políticas gerenciadas pelo cliente no Guia do Usuário do IAM.

Com as políticas baseadas em identidade do IAM, é possível especificar ações e recursos permitidos ou negados, assim como as condições sob as quais as ações são permitidas ou negadas. Para saber mais sobre todos os elementos que podem ser usados em uma política JSON, consulte Referência de elemento de 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

Compatibilidade com 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 especificado pode executar nesse atributo e em que condições. É necessário 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, é possível especificar uma conta inteira ou as entidades do IAM em outra conta como a entidade principal em uma política baseada em recursos. Consulte mais informações em Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

Ações políticas para AWS IoT FleetWise

Compatível com 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 podem ser usadas para permitir ou negar acesso em uma política. 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 políticas 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

Compatível com recursos 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 de política JSON `Resource` especifica o objeto ou os objetos aos quais a ação se aplica. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Para ações que não oferecem compatibilidade com permissões em nível de recurso, 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 de serviço: 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 `Condition` especifica quando as instruções são executadas com base em critérios definidos. É possível criar expressões condicionais que usem agentes de condição, como “igual a” ou “menor que”, para fazer a condição da política corresponder aos valores na solicitação. 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 AWS IoT FleetWise

Suportes ACLs: Não

As listas de controle de acesso (ACLs) controlam quais diretores (membros da conta, usuários ou funções) têm permissões para acessar um recurso. 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

Compatível com ABAC (tags em políticas): parcial

O controle de acesso por atributo (ABAC) é uma estratégia de autorização que define permissões com base em atributos chamados de tags. Você pode anexar tags a entidades e AWS recursos

do IAM e, em seguida, criar políticas ABAC para permitir operações quando a tag do diretor corresponder à tag no recurso.

Para controlar o acesso baseado em tags, forneça informações sobre as tags no elemento de condição de uma política usando as `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` ou chaves de condição `aws:TagKeys`.

Se um serviço for compatível com as três chaves de condição para cada tipo de recurso, o valor será Sim para o serviço. Se um serviço for compatível com as três chaves de condição somente para alguns tipos de recursos, o valor será Parcial

Para saber mais sobre o ABAC, consulte Definir permissões com autorização do ABAC no Guia do usuário do IAM. Para visualizar um tutorial com etapas para configurar o ABAC, consulte Usar controle de acesso por atributo (ABAC) no Guia do usuário do IAM.

> **Note**
>
> AWS FleetWise Só há suporte para IoT`iam:PassRole`, o que é necessário para a operação da `CreateCampaign` API.

Usando credenciais temporárias com IoT AWS FleetWise

Compatível com credenciais temporárias: sim

As credenciais temporárias fornecem acesso de curto prazo aos AWS recursos e são criadas automaticamente quando você usa a federação ou troca de funções. AWS recomenda que você gere credenciais temporárias dinamicamente em vez de usar chaves de acesso de longo prazo. Para ter mais informações, consulte Credenciais de segurança temporárias no IAM e Serviços da Serviços da AWS que funcionam com o IAM no Guia do usuário do IAM.

Permissões principais entre serviços para IoT AWS FleetWise

Compatibilidade com o recurso de encaminhamento de sessões de acesso (FAS): sim

As sessões de acesso direto (FAS) usam as permissões do principal chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) de fazer solicitações aos serviços posteriores. Para obter detalhes da política ao fazer solicitações de FAS, consulte Sessões de acesso direto.

Funções de serviço para AWS IoT FleetWise

Compatível com perfis de serviço: não

O perfil de serviço é um perfil do IAM que um serviço assume para executar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para saber mais, 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

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 editar as permissões para perfis vinculados ao serviço.

Para obter detalhes sobre como criar ou gerenciar perfis vinculados a serviços, consulte Serviços da AWS que funcionam com o IAM. Encontre um serviço na tabela que inclua um `Yes` na coluna Perfil vinculado ao serviço. Escolha o link Sim para visualizar a documentação do perfil vinculado a 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 obter mais informações, consulte Monitore AWS a IoT com a Amazon FleetWise 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 Wise AWSServiceRoleForIoTFleet— uma política gerenciada pela AWS que é usada para todas as permissões do out-of-the-box AWS IoT. FleetWise

A função vinculada ao serviço da AWSService RoleForIo TFleet Wise confia nos seguintes serviços para assumir a função:

- `IoTFleetWise`

A política de permissões de função nomeada AWSIoTFleetwiseServiceRolePolicypermite que o AWS IoT FleetWise conclua as seguintes ações nos recursos especificados:

- Ação: `cloudwatch:PutMetricData` no recurso *

Para obter informações sobre alterações nessa política, consulte as atualizações AWSIo TFleetwise ServiceRolePolicy da política.

A função vinculada ao serviço tem permissões para publicar métricas nos seguintes CloudWatch namespaces:

- `AWS/IoTFleetWise`— Para métricas específicas do serviço

- `AWS/Usage`— Para métricas de uso

Você deve configurar permissões para que uma entidade do IAM (por exemplo, um usuário, grupo ou função) crie, edite ou exclua um perfil vinculado 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 um perfil vinculado ao 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 obter mais informações, consulte Defina suas configurações AWS de IoT FleetWise.

Criação de uma função vinculada ao serviço no AWS FleetWise IoT (console)

Não é necessário criar manualmente um perfil vinculado ao 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 vinculada ao serviço da AWSService RoleForIo TFleet Wise 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 saber mais, 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. Para conceder permissão aos usuários para executar ações nos recursos que eles precisam, um administrador do IAM pode criar políticas do IAM.

Para aprender a criar uma política baseada em identidade do IAM ao usar esses documentos de política em JSON de exemplo, consulte Criar políticas do IAM (console) no Guia do usuário do IAM.

Para obter detalhes sobre ações e tipos de recursos definidos pela AWS IoT FleetWise, incluindo o formato do 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

- Práticas recomendadas de política
- Usando o console AWS de IoT FleetWise
- Permitir que os usuários visualizem suas próprias permissões
- Acessar recursos no Amazon Timestream

Práticas recomendadas 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 sua 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 saber mais, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções 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 recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para saber mais 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: é possível adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, é possível 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 CloudFormation. Para saber mais, consulte Elementos da política JSON do IAM: condição no Guia 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 práticas para ajudar a criar políticas seguras e funcionais. Para saber mais, 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 MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para saber mais, consulte Configuração de acesso à API protegido por MFA no Guia do Usuário do IAM.

Para saber mais 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 Caso crie uma política baseada em identidade mais restritiva 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 à 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 informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como criar uma política que permita 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:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
```

```
    ]
}
```

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, consulteDefina suas configurações AWS de IoT FleetWise.

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 Console de gerenciamento da AWS 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

É possível criar um perfil que os usuários de outras contas ou pessoas fora da organização podem usar para acessar seus recursos. É possível especificar quem é confiável para assumir o perfil. Para

serviços que oferecem suporte a 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 perfis e políticas baseadas em recurso para acesso entre contas, consulte Acesso a recursos entre contas no IAM no Guia do usuário do IAM.

AWS Referência de ações e recursos FleetWise da API de IoT

Ao gerenciar o acesso usando políticas para uma identidade do IAM, você pode usar a tabela a seguir como referência. A tabela lista cada FleetWise API de AWS IoT, as ações correspondentes para as quais você pode conceder permissões para realizar a ação e o AWS recurso para o qual você pode conceder as permissões.

Especifique as ações no campo `Action` da política e o valor do recurso no campo `Resource` da política. Para especificar uma ação, use o `iotfleetwise:` prefixo seguido pelo nome da ação. Por exemplo, .`iotfleetwise:CreateSignalCatalog`

Atualmente, a AWS IoT FleetWise oferece suporte às seguintes ações, recursos e chaves de condição.

Use a barra de rolagem para ver o resto da tabela.

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
AssociateVehicleFleet	IoT em termos de frota: Associate VehicleFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id* avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id*
CreateCampaign	IoT em termos de frota: CreateCampaign	avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name* avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*
CreateDecoderManifest	IoT em termos de frota: CreateDecoderManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name*
CreateFleet	IoT em termos de frota: CreateFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id* arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
CreateModelManifest	IoT em termos de frota: CreateModelManifest	avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
CreateSignalCatalog	IoT em termos de frota: CreateSignalCatalog	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
CreateStateTemplate	IoT em termos de frota: CreateStateTemplate	arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id* arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
CreateVehicle	IoT em termos de frota: CreateVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
BatchCreateVehicle	IoT em termos de frota: CreateVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
DeleteCampaign	IoT em termos de frota: DeleteCampaign	avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name*
DeleteDecoderManifest	IoT em termos de frota: DeleteDecoderManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
DeleteFleet	IoT em termos de frota: DeleteFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*
DeleteModelManifest	IoT em termos de frota: DeleteModelManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
DeleteSignalCatalog	IoT em termos de frota: DeleteSignalCatalog	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
DeleteStateTemplate	IoT em termos de frota: DeleteStateTemplate	arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*
DeleteVehicle	IoT em termos de frota: DeleteVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id*
DisassociateVehicleFleet	IoT em termos de frota: DisassociateVehicleFleet	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*
GetCampaign	IoT em termos de frota: GetCampaign	avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name*
GetDecoderManifest	IoT em termos de frota: GetDecoderManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
GetEncryptionConfiguration	IoT em termos de frota: GetEncryptionConfiguration	

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
GetFleet	IoT em termos de frota: GetFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*
GetLoggingOptions	IoT em termos de frota: GetLoggingOptions	
GetModelManifest	IoT em termos de frota: GetModelManifest	avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name*
GetRegisterAccountStatus	IoT em termos de frota: GetRegisterAccountStatus	
GetSignalCatalog	IoT em termos de frota: GetSignalCatalog	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
GetStateTemplate	IoT em termos de frota: GetStateTemplate	arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*
GetVehicle	IoT em termos de frota: GetVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id*
GetVehicleStatus	IoT em termos de frota: GetVehicleStatus	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id*
ImportDecoderManifest	IoT em termos de frota: ImportDecoderManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
ImportSignalCatalog	IoT em termos de frota: ImportSignalCatalog	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
ListCampaigns	IoT em termos de frota: ListCampaigns	
ListDecoderManifestNetworkInterfaces	IoT em termos de frota: ListDecoderManifestNetworkInterfaces	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
ListDecoderManifests	IoT em termos de frota: ListDecoderManifests	
ListDecoderManifestSignals	IoT em termos de frota: ListDecoderManifestSignals	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
ListFleets	IoT em termos de frota: ListFleets	
ListFleetsForVehicle	IoT em termos de frota: ListFleetsForVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id*
ListModelManifestNodes	IoT em termos de frota: ListModelManifestNodes	avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name*
ListModelManifests	IoT em termos de frota: ListModelManifests	

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
ListSignalCatalogNodes	IoT em termos de frota: ListSignalCatalogNodes	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
ListSignalCatalogs	IoT em termos de frota: ListSignalCatalogs	
ListStateTemplates	IoT em termos de frota: ListStateTemplates	
ListVehicles	IoT em termos de frota: ListVehicles	
ListVehiclesInFleet	IoT em termos de frota: ListVehiclesInFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
ListTagsForResource	IoT em termos de frota: ListTagsForResource	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name* avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id* avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*
PutEncryptionConfiguration	IoT em termos de frota: PutEncryptionConfiguration	
PutLoggingOptions	IoT em termos de frota: PutLoggingOptions	
RegisterAccount	IoT em termos de frota: RegisterAccount	

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
TagResource	IoT em termos de frota: TagResource	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name* avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id* avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
UntagResource	IoT em termos de frota: UntagResource	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name* avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id* avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name* arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*
UpdateCampaign	IoT em termos de frota: UpdateCampaign	avise ::iotfleetwise: ::campaign/ *partition region account_id campaign-name*
UpdateDecoderManifest	IoT em termos de frota: UpdateDecoderManifest	avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
UpdateFleet	IoT em termos de frota: UpdateFleet	avise ::iotfleetwise: ::fleet/ *partition region account_id fleet-id*
UpdateModelManifest	IoT em termos de frota: UpdateModelManifest	avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name*

AWS Ações de IoT FleetWise	Permissões obrigatórias	Recursos
UpdateSignalCatalog	IoT em termos de frota: UpdateSignalCatalog	arn ::iotfleetwise: *partition* ::signal-catalog/ *region account_id name*
UpdateStateTemplate	IoT em termos de frota: UpdateStateTemplate	arn ::iotfleetwise: ::state-template/ *partition region account_id state-template-id*
UpdateVehicle	IoT em termos de frota: UpdateVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*
BatchUpdateVehicle	IoT em termos de frota: UpdateVehicle	avisar ::iotfleetwise: ::veículo/ *partition region account_id vehicle-id* avise ::iotfleetwise: *partition* ::model-manifest/ *region account_id name* avise ::iotfleetwise: ::decoder-manifest/ *partition region account_id name*

AWS atualizações de políticas gerenciadas para o AWS IoT FleetWise

Veja detalhes sobre as atualizações das políticas AWS gerenciadas do AWS IoT FleetWise desde que esse serviço começou a monitorar essas mudanças. Para receber alertas automáticos sobre alterações nessa página, assine o feed RSS na página de histórico de documentos do AWS FleetWise IoT.

AWSIoTFleetwiseServiceRolePolicy

Essa política permite que o AWS IoT FleetWise publique métricas CloudWatch na Amazon em seu nome.

Alteração	Descrição	Data
`AWSIoTFleetwiseServiceRolePolicy` : atualizar para uma política existente	Foram adicionadas permissões para publicar métricas de uso no AWS/Usage namespace , além das permissões de `AWS/IoTFleetWise` namespace existentes.	13 de junho de 2025
`AWSIoTFleetwiseServiceRolePolicy` — O AWS IoT FleetWise começou a monitorar as mudanças	O AWS IoT FleetWise começou a monitorar as alterações dessa política.	27 de setembro de 2022

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 regulamentos aplicáveis. Para obter mais informações sobre sua responsabilidade de conformidade ao usar Serviços da AWS, consulte a documentação AWS de segurança.

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 altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de data center 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 suporta 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.
> Amazon Timestream não está disponível na região Ásia-Pacífico (Mumbai).

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 oferecer compatibilidade com:

- Transport Layer Security (TLS). Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Conjuntos de criptografia com perfect forward secrecy (PFS) como DHE (Ephemeral Diffie-Hellman) ou ECDHE (Ephemeral Elliptic Curve Diffie-Hellman). A maioria dos sistemas modernos, como Java 7 e versões posteriores, comporta esses modos.

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 ou específicos da Amazon Virtual Private Cloud (Amazon VPC). VPCs 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 VPC endpoint é representado por uma ou mais interfaces de rede elástica (ENIs) com endereços IP privados em suas sub-redes VPC.

A interface VPC endpoint conecta sua VPC diretamente à AWS FleetWise IoT sem um gateway de internet, dispositivo NAT, conexão VPN ou conexão. 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 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 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 Endpoints da VPC da interface (AWS PrivateLink) no Manual do Usuário do 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.

> **Note**
>
> AWS A IoT é FleetWise compatível com todos os endpoints de VPC com o modo de pilha dupla. Para obter informações sobre endpoints de serviço, consulte endpoints e cotas de AWS FleetWise IoT.

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 que usam 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. AWS *123456789012* 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 *fleetWise* usuário do IAM no. Conta da AWS *123456789012*

```
{
    "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 práticas recomendadas 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 Melhores práticas 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 “Práticas recomendadas 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 nos FleetWise recursos IDs de AWS IoT, por exemplo, em nomes de campanhas, manifestos de decodificadores, modelos de veículos e catálogos de sinais ou de veículos e frotas. IDs

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. Console de gerenciamento da AWS Para obter mais informações, consulte the section called “CloudTrail troncos”.

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 .

Monitore a AWS IoT FleetWise

O monitoramento é uma parte importante da manutenção da confiabilidade, disponibilidade e desempenho da AWS IoT FleetWise e de suas outras AWS soluções. AWS fornece as seguintes ferramentas de monitoramento para monitorar a AWS IoT FleetWise, relatar quando algo está errado e realizar ações automáticas quando apropriado:

- A Amazon CloudWatch monitora seus AWS recursos e os aplicativos nos quais você executa 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 CloudWatch rastrear o uso da CPU ou outras métricas de suas instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.
- O Amazon CloudWatch Logs pode ser usado para monitorar, armazenar e acessar seus arquivos de log das instâncias do Amazon EC2 e de outras fontes. CloudTrail CloudWatch Os registros podem monitorar as informações nos arquivos de log e notificá-lo quando determinados limites forem atingidos. É possível também arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por sua 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 ligaram AWS, o endereço IP de origem a partir do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

Monitore AWS a IoT com a Amazon FleetWise CloudWatch

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

As CloudWatch métricas da Amazon são uma forma de monitorar seus AWS recursos e o desempenho deles. AWS A IoT FleetWise envia métricas para. CloudWatch Você pode usar o Console de gerenciamento da AWS AWS CLI, o ou uma API para listar as métricas para as quais a

AWS IoT FleetWise envia. CloudWatch Para obter mais informações, consulte o Guia CloudWatch do usuário da Amazon.

> **Important**
>
> Você deve definir as configurações para que a AWS IoT FleetWise possa enviar métricas para o. CloudWatch Para obter mais informações, consulte Defina suas configurações AWS de IoT FleetWise.

O namespace `AWS/IoTFleetWise` inclui as métricas a seguir.

Métricas de sinal

Métrica	Description
IllegalMessageFromEdge	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise não correspondia ao formato exigido. Unidades: contagem Dimensões: nenhuma Estatísticas válidas: soma
MessageThrottled	Uma mensagem enviada do veículo para a AWS IoT FleetWise foi estrangulada. Isso ocorre porque você excedeu os limites de serviço dessa conta na região atual. Unidades: contagem Dimensões: nenhuma Estatísticas válidas: soma
ModelingError	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo.

Métrica	Description
	Unidades: contagem Dimensões: ModelName, StateTemplateName (opcional), SignalCatalogName (opcional)
DecodingError	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que falham na decodificação do manifesto do decodificador do veículo. Unidades: contagem Dimensões: DecoderName Estatísticas válidas: soma
MessageSizeLimitExceeded	Uma mensagem enviada do veículo para a AWS IoT FleetWise foi descartada. Isso ocorre porque você excedeu o tamanho máximo de um limite de serviço de mensagens para essa conta na região atual. Unidades: contagem Dimensões: nenhuma Estatísticas válidas: soma
CallCount	O número de mensagens ingeridas durante o período especificado. Unidades: contagem Dimensões: AccountID

Métrica	Description
CheckInThrottled	Um check-in enviado do veículo para a AWS IoT FleetWise foi acelerado. Isso ocorre porque você excedeu o limite de serviço dessa conta na região atual. Unidades: contagem Dimensões: VehicleName Estatísticas válidas: soma
VehicleAttributeNotFound	Uma mensagem enviada do veículo e recebida pela AWS IoT não FleetWise pôde ser enriquecida com os atributos especificados do veículo. Unidades: contagem Dimensões: nome da campanha (opcional), stateTemplateName (opcional), nome do veículo Estatísticas válidas: soma

Métricas do veículo

Métrica	Description
VehicleNotFound	Uma mensagem recebida pela AWS IoT FleetWise, em que o veículo é desconhecido. Unidades: contagem Dimensões: nenhuma Estatísticas válidas: soma

Métricas de implantação

Métrica	Description
PayloadSize	Tamanho da mensagem enviada da AWS IoT FleetWise para o veículo. Unidades: contagem Dimensões: VehicleName, ResourceTypes (StateTemplates, Campanhas, DecoderMa nifest)
PayloadSizeLimitExceeded	Uma mensagem enviada da AWS IoT FleetWise para o veículo excedeu o tamanho máximo de um limite de serviço de carga útil para essa conta na região atual. Unidades: contagem Dimensões: VehicleName Estatísticas válidas: soma

Métricas de campanha

Métrica	Description
CampaignInvalid	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que a campanha não é válida. Unidades: contagem Dimensões: CampaignName Estatísticas válidas: soma

Métrica	Description
CampaignNotFound	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, onde a campanha é desconhecida. Unidades: contagem Dimensões: CampaignName Estatísticas válidas: soma

Métricas do modelo estadual

Métrica	Description
NoStateTemplatesAssociated	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, na qual nenhum modelo de estado está associado ao veículo. Unidades: contagem Estatística válida: soma

Métricas de destino dos dados da campanha

Métrica	Description
TimestreamWriteError	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Amazon Timestream. Unidades: contagem Dimensões: DatabaseName, TableName Estatísticas válidas: soma

Métrica	Description
S3 WriteError	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3). Unidades: contagem Dimensões: BucketName Estatísticas válidas: soma
S3 ReadError	AWS A IoT não FleetWise conseguiu ler uma chave de objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3). Unidades: contagem Dimensões: BucketName Estatísticas válidas: soma

AWS KMS Principais métricas gerenciadas pelo cliente

Métrica	Description
KMSKeyAccessDenied	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Timestrea m ou no bucket do Amazon S3 devido a um erro de acesso negado à chave. AWS KMS Unidades: contagem Dimensões: KMSKey ID Estatísticas válidas: soma

Monitore AWS a IoT com o FleetWise Amazon Logs CloudWatch

> **Important**
>
> O acesso a determinados FleetWise recursos de AWS IoT está atualmente bloqueado. Para obter mais informações, consulte AWS Disponibilidade de regiões e recursos na AWS IoT FleetWise.

O Amazon CloudWatch Logs monitora os eventos que ocorrem em seus recursos e alerta você se houver 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.

Exibir FleetWise registros AWS de IoT no console CloudWatch

> **Important**
>
> Antes de ver o grupo de FleetWise registros de AWS IoT no CloudWatch console, verifique se o seguinte é verdadeiro:
>
> - Você ativou o login na AWS IoT FleetWise. Para obter mais informações sobre registro em log, consulte Configurar o AWS registro de IoT FleetWise.
> - Já existem entradas de registro gravadas por AWS IoT operações.

Para ver seus FleetWise registros de AWS IoT no console 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 a criação de expressões de filtro, consulte Sintaxe de filtros e padrões 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	Description
MODELING_ERROR	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que não são validados em relação ao modelo do veículo. Atributos: nome do veículo, nome da campanha (opcional), signalID (opcional) signalCatalogName, valor do sinal (opcional), signalValueRange mínimo (opcional), signalVal ueRange máximo (opcional), (opcional), signalIDs, modelManifestName stateTemp lateName

Tipo de evento	Description
ILLEGAL_MESSAGE_FROM_EDGE	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise não correspondia ao formato exigido. Atributos: nome do veículo, nome da campanha, signalCatalogName
DECODING_ERROR	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise contém sinais que falham na decodificação do manifesto do decodificador do veículo. Atributos: CampaignName,, signalCat alogName decoderManifestName, (opcional) SignalName, (opcional) S3URI
MESSAGE_THROTTLED	Uma mensagem enviada do veículo para a AWS IoT FleetWise foi estrangulada. Isso ocorre porque você excedeu os limites de serviço dessa conta na região atual. Atributos: accountId, vehicleName, message, eventType, logLevel, timestamp
TAMANHO_LIMITE_DA MENSAGEM EXCEDIDO	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise excede o tamanho máximo de um limite de serviço de mensagens. Atributos: AccountId, nome do veículo
CHECKIN_THROTTLED	Um check-in enviado do veículo para a AWS IoT FleetWise foi acelerado. Isso ocorre porque você excedeu o limite de serviço dessa conta na região atual. Atributos: Nome do veículo

Tipo de evento	Description
ATRIBUTO_DO_VEÍCULO_NÃO_ENCONTRADO	Uma mensagem enviada do veículo e recebida pela AWS IoT não FleetWise pôde ser enriquecida com os atributos especificados do veículo. Atributos: CampaignName (opcional), stateTemplateName (opcional), VehicleName, vehicleAttributeNames

Tipos de eventos de veículos

Tipo de evento	Description
VEHICLE_NOT_FOUND	Uma mensagem recebida pela AWS IoT FleetWise, em que o veículo era desconhecido. Atributos: nome do veículo, nome da campanha (opcional), (opcional) stateTemplateName

Tipos de eventos de implantação

Tipo de evento	Description
LIMITE DE TAMANHO DA CARGA ÚTIL EXCEDIDO	Uma mensagem enviada da AWS IoT FleetWise para o veículo excedeu o limite máximo de tamanho do serviço. Atributos: nome do veículo, nome da campanha (opcional), (opcional) stateTemplateName

Tipos de evento de campanha

Tipo de evento	Description
CAMPAIGN_NOT_FOUND	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, onde a campanha era desconhecida. Atributos: vehicleName (opcional), campaignName
CAMPAIGN_INVALID	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que a campanha não era válida. Atributos: vehicleName (opcional), campaignName

Tipos de eventos de destino de dados da campanha

Tipo de evento	Description
TIMESTREAM_WRITE_ERROR	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Amazon Timestream. Atributos: nome do veículo, nome da campanha,, timestreamDatabaseName timestreamTableName
S3_WRITE_ERROR	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo no bucket do Amazon Simple Storage Service (Amazon S3). Atributos: campaignName, destinationName
S3_READ_ERROR	AWS A IoT não FleetWise conseguiu ler uma chave de objeto do veículo no bucket do Amazon Simple Storage Service (Amazon S3).

Tipo de evento	Description
	Atributos: campaignName, destinationName

Tipos de eventos de modelo de estado

Tipo de evento	Description
STATE_TEMPLATE_NOT_FOUND	Uma mensagem enviada do veículo e recebida pela AWS IoT FleetWise, em que o modelo de estado era desconhecido. Atributos: nome do veículo (opcional), stateTemplateName

Tipos de eventos AWS KMS principais gerenciados pelo cliente

Tipo de evento	Description
KMS_KEY_ACCESS_DENIED	AWS A IoT não FleetWise conseguiu gravar uma mensagem do veículo na tabela Timestrea m ou no bucket do Amazon S3 devido a um erro de acesso negado à chave. AWS KMS Atributos: kmsKeyId (opcional), resourceArn (opcional)

Atributos

Todas as entradas de CloudWatch registros incluem os seguintes atributos:

accountId

Sua Conta da AWS identidade.

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 data e hora de milissegundos da época em que a IoT AWS processou o registro. FleetWise

Atributos opcionais

CloudWatch As entradas de registros incluem opcionalmente esses atributos, dependendo de`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 sinais de erro IDs.

signalName

O nome do sinal.

signalTimestampEpochSenhora

A marcação de data/hora do sinal de erro.

signalValue

O valor do sinal de erro.

signalValueRangeMáximo

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.

vehicleAttributeNames

Uma lista de nomes de atributos de veículos que não puderam ser encontrados.

Configurar o AWS registro de IoT FleetWise

Você pode enviar seus dados de FleetWise registro de AWS IoT para um grupo de CloudWatch registros. CloudWatch Os registros oferecem visibilidade caso a AWS IoT FleetWise falhe em processar mensagens de veículos. Por exemplo, isso pode acontecer devido a uma configuração

incorreta ou a outros erros do cliente. Você receberá notificações sobre quaisquer erros para poder identificar e mitigar problemas.

Antes de enviar registros para CloudWatch, você deve criar um grupo de CloudWatch registros. Configure o grupo de registros com a mesma conta e na mesma região que você usou com a AWS IoT FleetWise. Ao habilitar o login na AWS IoT FleetWise, forneça o nome do grupo de registros. Depois que o registro é ativado, a AWS IoT FleetWise entrega os registros ao grupo de CloudWatch registros em fluxos de registros.

Você pode visualizar os dados de log enviados da AWS IoT FleetWise no CloudWatch console. Para obter mais informações sobre como configurar um grupo de CloudWatch registros e visualizar dados de registros, consulte Trabalhando com grupos de registros. Para obter mais informações sobre como configurar registros para AWS serviços, consulte AWS serviços que publicam registros em CloudWatch Logs no Guia do usuário do Amazon CloudWatch Logs.

Permissões para publicar registros em CloudWatch

A configuração do registro em CloudWatch log para um grupo de registros requer as configurações de permissões descritas nesta seção. Para obter informações sobre o gerenciamento de permissões, consulte Gerenciamento de acesso para AWS recursos no Guia do usuário do IAM.

Com essas permissões, você pode alterar a configuração de registro, configurar a entrega de CloudWatch registros e recuperar informações sobre seu grupo de registros.

JSON

```
{
    "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 AWS recursos, isso é indicado na política com uma `"Resource"` configuração de`"*"`. Isso significa que as ações são permitidas em todos os AWS recursos que cada ação suporta.

Configurar o registro usando o console

Esta seção descreve como usar o FleetWise console de AWS IoT para configurar o registro.

Para usar o FleetWise console de AWS IoT para configurar o registro

1. Abra o console AWS de IoT. FleetWise
2. No painel esquerdo, escolha Settings (Configurações).
3. Na seção Registro em log da página Configurações, selecione Editar.
4. Na seção de CloudWatch registro, insira o grupo Registro.
5. Selecione Enviar para salvar as alterações.

Depois de ativar o registro, você pode visualizar seus dados de registro no CloudWatch console.

Configurar o registro usando a CLI

Esta seção descreve como configurar o registro em log para AWS IoT FleetWise usando a CLI.

Você também pode realizar esse procedimento com a API usando os métodos na AWS API que correspondem aos comandos da CLI mostrados aqui. Você pode usar a operação da GetLoggingOptionsAPI para buscar a configuração atual e a operação da PutLoggingOptionsAPI para modificar a configuração.

Para usar a CLI para configurar o registro para IoT AWS 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
```

em que:

logType

O tipo de registro para enviar dados para o CloudWatch Logs. Para desativar o registro em log, altere o valor para `OFF`.

logGroupName

O grupo de CloudWatch registros para o qual a operação envia dados. Certifique-se de criar o nome do grupo de registros antes de habilitar o registro para AWS IoT FleetWise.

Depois de habilitar o registro, consulte Pesquisar entradas de registro usando a AWS CLI.

Registre chamadas da FleetWise API de AWS IoT usando AWS CloudTrail

AWS A IoT FleetWise é integrada com AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, função ou AWS serviço na IoT AWS . FleetWise CloudTrail captura todas as chamadas de API para AWS FleetWise IoT como eventos. As chamadas capturadas incluem chamadas do FleetWise console de AWS IoT e chamadas de código para as operações da API de AWS FleetWise IoT. Se você criar uma trilha, poderá habilitar a entrega contínua de CloudTrail eventos para um bucket do Amazon S3, incluindo eventos para IoT AWS . FleetWise Se

você não configurar uma trilha, ainda poderá ver os eventos mais recentes no CloudTrail console no Histórico de eventos. Usando as informações coletadas por CloudTrail, você pode determinar a solicitação que foi feita à 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 isso CloudTrail, consulte o Guia AWS CloudTrail do usuário.

AWS Informações de IoT em FleetWise CloudTrail

CloudTrail é ativado em sua AWS conta quando você cria a conta. Quando a atividade ocorre na AWS IoT FleetWise, essa atividade é registrada em um CloudTrail evento junto com outros eventos de AWS serviço no histórico de eventos. É possível visualizar, pesquisar e baixar eventos recentes em sua AWS conta. Para obter mais informações, consulte Visualização de eventos com histórico de CloudTrail eventos.

Para um registro contínuo dos eventos em sua AWS conta, incluindo eventos de AWS IoT FleetWise, crie uma trilha. Uma trilha permite CloudTrail entregar arquivos de log para 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 eventos de todas as regiões na AWS partição e entrega os arquivos de log ao bucket do Amazon S3 que você especificar. Além disso, você pode configurar outros AWS serviços para analisar e agir com base nos dados de eventos coletados nos CloudTrail registros. Para saber mais, consulte:

- Visão geral da criação de uma trilha
- CloudTrail serviços e integrações suportados
- Configurando notificações do Amazon SNS para CloudTrail
- Recebendo arquivos de CloudTrail log de várias regiões
- Recebendo arquivos de CloudTrail log de várias contas

Todas as FleetWise ações de AWS IoT são registradas CloudTrail e documentadas na Referência da API de IoT AWS . FleetWise Por exemplo, chamadas para as `GetModelManifest` ações `CreateCampaignAssociateVehicleFleet`, e geram entradas nos arquivos de CloudTrail log.

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 usuário do IAM.

- Se a solicitação foi feita com credenciais de segurança temporárias de um perfil ou de um usuário federado.
- Se a solicitação foi feita por outro AWS serviço.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail .

Entenda as entradas do arquivo de FleetWise log AWS de IoT

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log para um bucket do Amazon S3 que você especificar. CloudTrail os arquivos de log 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 da solicitação e assim por diante. CloudTrail os arquivos de log não são um rastreamento de pilha ordenado das chamadas públicas de API, portanto, eles não aparecem em nenhuma ordem específica.

O exemplo a seguir mostra uma entrada de CloudTrail registro que demonstra a *AssociateVehicleFleet* operação.


```
{
      "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 AWS desenvolvedor de IoT FleetWise

A tabela a seguir descreve as versões da documentação para AWS IoT FleetWise.

Alteração	Descrição	Data
Atualização de monitoramento	Atualizou o FleetWise monitoramento de AWS IoT para incluir mais métricas e opções de registro. Para obter mais informações, consulte Monitorar a AWS IoT FleetWise com Amazon CloudWatch e Monitorar a IoT com o Amazon AWS FleetWise Logs. CloudWatch	23 de dezembro de 2025
Atualização da política de função vinculada ao serviço	Atualizado `AWSIoTFleetwiseServiceRolePolicy` para incluir permissões para publicar métricas de uso no `AWS/Usage` namespace. Para obter mais informações, consulte as atualizações AWSIo TFleetwise ServiceRolePolicy da política.	13 de junho de 2025
Expansão de região	AWS A IoT agora FleetWise está disponível na região Ásia-Pacífico (Mumbai) (somente acesso fechado).	21 de novembro de 2024
Disponibilidade geral limitada de novos recursos	AWS A IoT FleetWise agora oferece suporte ao acesso	21 de novembro de 2024

	fechado para campanhas para armazenar e encaminhar dados, configurar um tópico do MQTT como destino de dados e coletar dados de códigos de diagnóstico de problemas . Agora, ele também suporta acesso fechado para coleta de dados independente de rede usando interfaces de decodificação personalizadas, configurando comandos e monitorando o último estado conhecido dos veículos.	
Enviar dados da campanha para um tópico do MQTT	AWS A IoT FleetWise agora suporta o envio de dados coletados durante campanhas para um tópico do MQTT especificado por você, além da capacidade de armazenar os dados no Amazon S3 ou no Amazon Timestream.	1º de maio de 2024

Prévia dos dados de sistemas de visão	Você pode usar a visualização prévia dos dados do sistema de visão da 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
AWS KMS chaves gerenciadas pelo cliente	AWS A IoT FleetWise agora oferece suporte a chaves gerenciadas pelo AWS KMS cliente. Você pode usar a chave KMS para criptografar dados do lado do servidor relacionados aos FleetWise recursos de AWS IoT (catálogo de sinais, modelo do veículo, manifesto do decodificador, veículos e configurações da campanha de coleta de dados) armazenados. Nuvem AWS	16 de outubro de 2023

Armazenamento de objetos no Amazon S3	AWS A IoT FleetWise agora suporta 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.	1º de junho de 2023
Disponibilidade geral	Este é o lançamento público da AWS IoT FleetWise.	27 de setembro de 2022
Lançamento inicial	Esta é a versão prévia do Guia do FleetWise Desenvolvedor de AWS IoT.	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.