



开发人员指南

# AWS 物联网 FleetWise



# AWS 物联网 FleetWise: 开发人员指南

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

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

# Table of Contents

什么是 AWS IoT FleetWise ? .....	1
优势 .....	2
使用案例 .....	2
重要提示 .....	3
你是 AWS 物联网新手 FleetWise 吗? .....	3
访问 AWS 物联网 FleetWise .....	4
AWS 物联网的定价 FleetWise .....	4
相关服务 .....	4
重要概念 .....	4
重要概念 .....	5
AWS 物联网的特点 FleetWise .....	8
支持的 AWS 区域 .....	9
设置 AWS 物联网 FleetWise .....	11
设置你的 AWS 账户 .....	11
注册获取 AWS 账户 .....	11
创建具有管理访问权限的用户 .....	11
开始使用控制台 .....	13
配置您的设置 .....	13
配置设置 ( 控制台 ) .....	13
配置设置 (AWS CLI) .....	14
IPv6 与 AWS 物联网配合使用 FleetWise .....	16
IPv6 控制平面端点的先决条件 .....	16
IPv6 对 AWS PrivateLink 端点的支持 .....	16
测试 IPv6 地址兼容性 .....	16
在 IAM 策略中使用 IPv6 地址 .....	17
使用双堆栈端点 .....	18
开始使用 .....	20
简介 .....	20
先决条件 .....	20
步骤 1 : 为 AWS 物联网设置边缘代理软件 FleetWise .....	21
步骤 2 : 创建车辆型号 .....	22
步骤 3 : 创建解码器清单 .....	24
步骤 4 : 配置解码器清单 .....	25
步骤 5 : 创建车辆 .....	25

步骤 6：创建活动 .....	27
步骤 7：清除 .....	28
后续步骤 .....	28
摄取数据 .....	29
模型车辆 .....	32
信号目录 .....	34
配置信号 .....	36
创建信号目录 .....	42
导入信号目录 .....	47
更新信号目录 .....	57
删除信号目录 .....	60
获取信号目录信息 .....	61
车辆型号 .....	62
创建车辆型号 .....	63
更新车辆模型 .....	70
删除车辆型号 .....	72
获取车辆型号信息 .....	73
解码器清单 .....	74
配置接口和信号 .....	76
创建解码器清单 .....	79
更新解码器清单 .....	88
删除解码器清单 .....	91
获取解码器清单信息 .....	93
管理车辆 .....	95
预配置车辆 .....	96
对车辆进行身份验证 .....	96
授权车辆 .....	98
保留的主题 .....	100
创建车辆 .....	103
创建车辆（控制台） .....	103
创建车辆（AWS CLI） .....	105
创建多辆车 .....	108
更新车辆 .....	109
更新多辆车辆 .....	112
删除车辆 .....	113
删除车辆（控制台） .....	114

删除车辆 (AWS CLI) .....	114
获取车辆信息 .....	115
管理车队 .....	117
创建实例集 .....	118
将车辆与车队关联 .....	119
解除车辆与车队的关联 .....	120
更新舰队 .....	121
删除实例集 .....	122
验证舰队已删除 .....	122
获取车队信息 .....	123
通过广告活动管理数据 .....	126
创建活动 .....	131
创建活动 (控制台) .....	131
创建活动 (AWS CLI) .....	139
AWS 物联网 FleetWise 活动的逻辑表达式 .....	144
更新市场活动 .....	145
删除市场活动 .....	146
创建活动 (控制台) .....	146
删除活动 (AWS CLI) .....	147
确认广告活动已删除 .....	147
获取活动信息 .....	147
存储并转发 .....	148
创建数据分区 .....	149
上传广告活动数据 .....	152
使用 AWS IoT 任务上传数据 .....	153
收集诊断故障代码数据 .....	154
诊断故障代码关键字 .....	155
为诊断故障代码创建数据收集活动 .....	157
诊断故障代码用例 .....	160
可视化车辆数据 .....	163
处理发送到 MQTT 主题的车辆数据 .....	163
在 Timestream 中处理车辆数据 .....	164
可视化存储在 Timestream 中的车辆数据 .....	165
在 Amazon S3 中处理车辆数据 .....	166
亚马逊 S3 对象格式 .....	167
分析存储在 Amazon S3 中的车辆数据 .....	168

命令 .....	169
命令概念 .....	170
命令关键概念 .....	170
命令执行状态 .....	172
车辆和指挥系统 .....	176
工作流程概述 .....	176
车辆工作流程 .....	177
命令工作流程 .....	180
( 可选 ) 命令通知 .....	181
创建和管理命令 .....	182
创建命令资源 .....	183
检索关于命令的信息 .....	185
在你的账户中列出命令 .....	186
更新或弃用命令资源 .....	186
删除命令资源 .....	188
启动和监控命令执行 .....	189
更新命令执行结果 .....	192
获取命令执行 .....	194
列出您账户中的命令执行情况 .....	195
删除命令执行 .....	197
示例：使用命令 .....	197
车辆转向模式示例概述 .....	198
先决条件 .....	198
用于使用远程命令的 IAM 政策 .....	198
运行 AWS IoT 命令 (AWS CLI) .....	201
清理 .....	205
命令使用场景 .....	206
创建不带参数的命令 .....	207
使用参数的默认值创建命令 .....	208
使用参数值创建命令 .....	209
使用带有状态模板的命令 .....	210
最后一个已知状态 .....	213
创建状态模板 .....	214
将物 AWS 联网 FleetWise 状态模板与车辆关联 .....	215
更新状态模板 .....	216
删除状态模板 .....	218

获取状态模板信息 .....	218
状态模板操作 .....	219
激活和停用状态数据收集 .....	220
获取车辆状态快照 .....	224
使用 MQTT 消息处理上次已知的状态车辆数据 .....	226
配置与网络无关的数据收集 .....	231
简介 .....	231
环境设置 .....	231
数据模型 .....	231
信号目录更新 .....	232
车辆模型和解码器 .....	233
发送命令 .....	236
AWS CLI 和 SDKs .....	238
问题排查 .....	239
解码器清单问题 .....	239
边缘代理问题 .....	242
问题：边缘代理软件无法启动。 .....	242
问题：[错误] [IoTFleetWiseEngine:: connect]: [无法初始化持久性库] .....	243
问题：Edge Agent 软件不收集板载诊断 (OBD) II PIDs 和诊断故障代码 (DTCs)。 .....	244
问题：适用于 AWS IoT FleetWise 软件的 Edge Agent 无法从网络收集数据或无法应用数据 检查规则。 .....	244
问题：[错误] [AwsIotConnectivityModule:: connect]: [连接失败并出现错误] 或 [警告] [AwsIotChannel:: send]: [没有活动的 MQTT 连接。] .....	245
存储和转发问题 .....	245
问题：收到AccessDeniedException具有所有必需的 IAM 权限的 .....	245
问题：上传到 AWS IoT Jobs 的数据忽略了 endTime .....	245
问题：上传到 AWS IoT 作业的数据处于REJECTED执行状态。 .....	245
安全性 .....	246
数据保护 .....	246
AWS 物联网中的静态加密 FleetWise .....	247
传输中加密 .....	248
AWS 物联网中的数据加密 FleetWise .....	248
控制访问权限 .....	259
授 AWS IoT FleetWise 予发送和接收 MQTT 主题数据的权限 .....	260
授予 AWS IoT FleetWise 对 Amazon S3 目的地的访问权限 .....	263
授予 AWS IoT FleetWise 访问亚马逊 Timestream 目的地的权限 .....	265

授 AWS IoT Device Management 予使用以下命令生成有效负载的权限 AWS IoT FleetWise .....	269
身份和访问管理 .....	273
受众 .....	274
使用身份进行身份验证 .....	274
使用策略管理访问 .....	275
AWS 物联网如何 FleetWise 与 IAM 配合使用 .....	277
基于身份的策略示例 .....	283
问题排查 .....	286
API 权限参考 .....	288
托管策略更新 .....	296
AWSIoTfleetwiseserviceRolePolicy .....	296
合规性验证 .....	297
恢复能力 .....	297
基础结构安全性 .....	298
FleetWise 通过接口 VPC 终端节点连接到 AWS 物联网 .....	298
配置和漏洞分析 .....	301
安全最佳实践 .....	302
授予可能的最低权限 .....	302
不要记录敏感信息 .....	302
AWS CloudTrail 用于查看 API 调用历史记录 .....	302
使设备时钟保持同步 .....	302
监控 AWS 物联网 FleetWise .....	303
使用监控 CloudWatch .....	303
使用 CloudWatch 日志进行监控 .....	309
在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志 .....	309
配置日志记录 .....	315
CloudTrail 日志 .....	318
AWS 中的物联网 FleetWise 信息 CloudTrail .....	318
了解 日志文件条目 .....	319
文档历史记录 .....	321
.....	cccxxiv

# 什么是 AWS 物联网 FleetWise ?

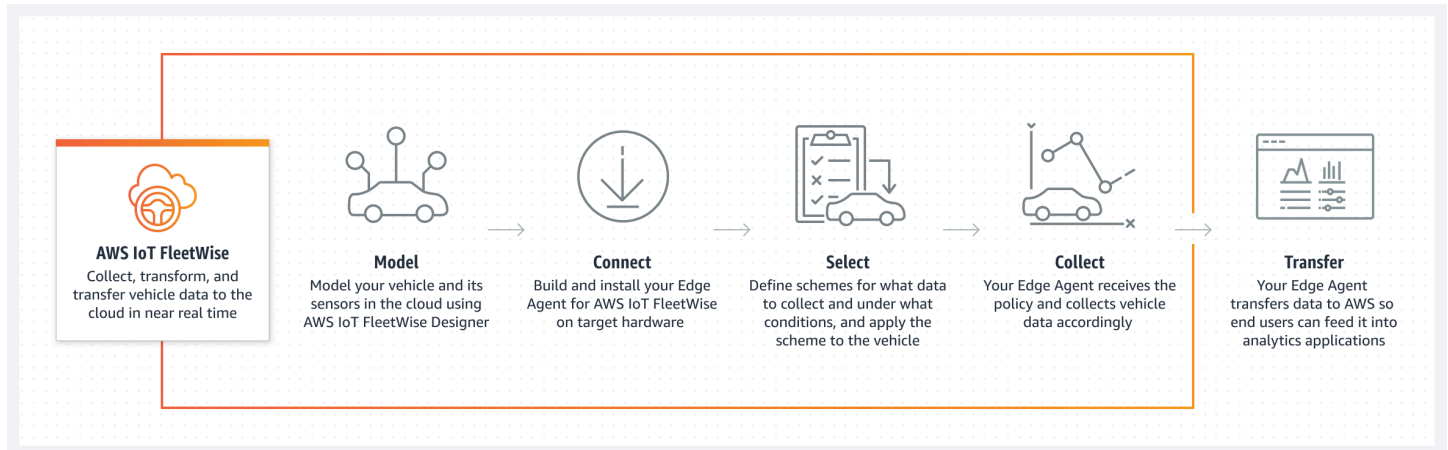
## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

AWS 物联网 FleetWise 是一项托管服务，可用于收集车辆数据并将其整理到云端。您可以使用收集的数据来改善车辆质量、性能和自动驾驶性能。借助 AWS 物联网 FleetWise，您可以从使用不同协议和数据格式的车辆收集和整理数据。AWS 物联网 FleetWise 有助于将低级消息转换为人类可读的值，并标准化云端的数据格式以进行数据分析。您也可以制定数据收集方案，控制从车辆中收集哪些数据以及何时将该数据传输到云端。

当车辆数据存储在云端时，您可以将其用于分析车队健康状况的应用程序。这些数据可以帮助您识别潜在维护问题，使车载信息娱乐系统更加智能，并通过分析和机器学习 (ML) 改进自动驾驶和驾驶辅助系统等先进技术。

下图显示了 AWS 物联网的基本架构 FleetWise。



## 主题

- [优势](#)
- [使用案例](#)
- [重要提示](#)
- [你是 AWS 物联网新手 FleetWise 吗？](#)
- [访问 AWS 物联网 FleetWise](#)

- [AWS 物联网的定价 FleetWise](#)
- [相关服务](#)
- [AWS 物联网的主要概念和特点 FleetWise](#)
- [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)

## 优势

AWS 物联网的主要优势 FleetWise 是：

### 更智能地收集车辆数据

通过智能数据收集提高数据相关性，这种数据收集方式仅将您需要的数据发送到云端进行分析。

### 轻松分析标准化的车队范围数据

无需开发自定义数据收集或日志记录系统，即可分析来自队列的标准化数据。

### 自动同步云端数据

统一查看从标准传感器（遥测数据）和视觉系统（来自摄像头、雷达和激光雷达的数据）收集的数据，并在云中自动保持同步。AWS 物联网 FleetWise 使结构化和非结构化视觉系统数据、元数据和标准传感器数据在云端自动同步。这简化了整理事件全景视图并获得见解的过程。

### 将数据存储于边缘并在最佳条件下转发

通过临时存储车辆上的数据来降低传输成本。您可以在指定的最佳条件下将所选数据转发到云端，例如当车辆连接到 Wi-Fi 时。

#### Note

视觉系统数据目前为预览版，可能会发生变化。

## 使用案例

您可以使用 AWS 物联网的场景 FleetWise 包括以下几种：

### 火车 AI/ML 模型

通过从生产车辆收集数据，持续改进用于自动驾驶和高级驾驶辅助系统的机器学习模型。

## 增强数字化客户体验

使用来自信息娱乐系统的数据，使车载视听内容和应用内见解更具相关性。

## 保持车队健康

使用来自车队数据的见解来监控电动汽车电池的运行状况和充电水平，管理维护计划，以及分析油耗等。

## 创建和管理命令

使用命令从云端对车辆执行命令。您可以远程向车辆发送命令，车辆将在几秒钟内执行命令。例如，您可以配置命令来锁定车辆的车门或设置温度。

## 创建和管理状态模板

状态模板为车主提供了一种跟踪其车辆状态的机制。在车辆上运行的 AWS 物联网 FleetWise 边缘代理收集信号更新并将其发送到云端。

# 重要提示

通过您使用 AWS 物联网收集的车辆数据 FleetWise 仅用于信息目的（包括帮助您训练基于云的人工智能和机器学习模型），您不得使用 AWS 物联网 FleetWise 来控制或操作车辆功能。对于超出 AWS 物联网 FleetWise 预期目的且以任何违反适用车辆法规的方式使用而可能产生的所有责任，您应承担全部责任。

FleetWise 应根据您的用例评估通过您使用 AWS 物联网收集的车辆数据的准确性，包括为了履行适用的车辆安全法规（例如安全监控和报告义务）下可能承担的任何合规义务。此类评估应包括通过其他行业标准手段和来源（例如车辆驾驶员的报告）收集和审查信息。您和您的最终用户对基于您对 AWS 物联网的使用所做的所有决定、给出的建议、采取的行动以及未能采取行动承担全部责任 FleetWise。

## 你是 AWS 物联网新手 FleetWise 吗？

如果您不熟悉 AWS IoT FleetWise，我们建议您先阅读以下章节：

- [AWS 物联网的主要概念和特点 FleetWise](#)
- [设置 AWS 物联网 FleetWise](#)
- [教程：AWS 物联网入门 FleetWise](#)
- [将 AWS 物联网 FleetWise 数据摄取到云端](#)

# 访问 AWS 物联网 FleetWise

您可以使用 AWS 物联网 FleetWise 控制台或 API 访问 AWS 物联网 FleetWise。

## AWS 物联网的定价 FleetWise

车辆通过 MQTT 消息将数据发送到云端。您每月月底为在 AWS 物联网中创建的车辆付款 FleetWise。您还需要为从车辆上收集的消息付费。有关定价的最新信息，请参阅 [AWS IoT FleetWise 定价](#) 页面。要了解有关 MQTT 消息协议的更多信息，请参阅《AWS IoT Core 开发人员指南》中的 [MQTT](#)。

## 相关服务

AWS 物联网与以下 AWS 服务 FleetWise 集成，以提高您的云解决方案的可用性和可扩展性。

- AWS IoT Core— 注册和控制将车辆数据上传到 AWS 物联网 FleetWise 并远程向车辆发送命令的 AWS IoT 设备。有关更多信息，请参阅《AWS IoT 开发人员指南》中的 [什么是 AWS IoT](#)。
- Amazon Timestream – 使用时间序列数据库存储和分析您的车辆数据。有关更多信息，请参阅《Amazon Timestream 开发人员指南》中的 [什么是 Amazon Timestream](#)。
- Amazon S3 – 使用对象存储服务来存储和管理您的车辆数据。有关更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的 [什么是 Amazon S3](#)。

## AWS 物联网的主要概念和特点 FleetWise

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下各节概述了物 AWS 联网 FleetWise 服务组件及其交互方式。

阅读本简介后，请参阅 [设置 AWS 物联网 FleetWise](#) 部分以了解如何设置 AWS 物联网 FleetWise。

### 主题

- [重要概念](#)
- [AWS 物联网的特点 FleetWise](#)

## 重要概念

AWS 物联网 FleetWise 提供了一个车辆建模框架，供您在云端对车辆及其传感器和执行器进行建模。为了实现车辆与云之间的安全通信，AWS 物联网 FleetWise 还提供了参考实现，以帮助您开发可安装在车辆中的边缘代理软件。您可以在云端定义数据收集方案，并将其部署到您的车辆上。在您的车辆中运行的边缘代理软件使用数据收集方案来控制收集哪些数据以及何时将其传输到云端。

以下是 AWS 物联网的核心概念 FleetWise。

### 信号

信号是用于定义是否包含车辆数据及其元数据的基本结构。信号可以是属性、分支、传感器或执行器。例如，您可以创建一个传感器来接收车载温度值并存储其元数据，包括传感器名称、数据类型和单位。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 信号目录](#)。

### 属性

属性表示通常不会变化的静态信息，例如制造商和生产日期。

### 分支

分支表示嵌套结构中的信号。分支演示信号层次结构。例如，Vehicle 分支有一个子分支 Powertrain。Powertrain 分支有一个子分支 combustionEngine。要找到 combustionEngine 分支，请使用 Vehicle.Powertrain.combustionEngine 表达式。

### 传感器

传感器数据报告车辆的当前状态，并随着车辆状态（例如液位、温度、振动或电压）的变化而随着时间的推移变化。

### 执行器

执行器数据报告车辆设备（例如电机、加热器和门锁）的状态。更改车辆设备的状态可以更新执行器数据。例如，您可以定义一个代表加热器的执行器。开启或关闭加热器时，执行器会收到新的数据。

### 自定义结构

自定义结构（也称为结构体）表示复杂或高阶数据结构。它便于对来自同一来源的数据进行逻辑绑定或分组。结构用于在原子操作中读取或写入数据，例如表示复杂数据类型或高阶形状。

在信号目录中，使用对结构数据类型的引用而不是基元数据类型来定义结构类型的信号。结构可用于所有类型的信号，包括传感器、属性、执行器和视觉系统数据类型。如果发送或接收结构

类型的信号，AWS IoT FleetWise 预计所有包含的项目都具有有效值，因此所有项目都是必填项。例如，如果结构包含项目 `Vehicle.Camera.Image.height`、`Vehicle.Camera.Image.width` 和 `Vehicle.Camera.Image.data`，预计发送的信号包含所有这些项目的值。

### Note

视觉系统数据目前为预览版，可能会发生变化。

## 自定义属性

自定义属性代表复杂数据结构的成员。该属性的数据类型可以是基元结构，也可以是其他结构。

当使用结构和自定义属性表示高阶形状时，始终将预期的高阶形状定义为树结构并将其视为树结构。自定义属性用于定义所有叶节点，而结构用于定义所有非叶节点。

## 信号目录

信号目录包含一系列信号。信号目录中的信号可用于对使用不同协议和数据格式的车辆进行建模。例如，有两辆汽车由不同的汽车制造商制造：一辆使用控制区域网络（CAN 总线）协议；另一辆使用车载诊断（OBD）协议。您可以在信号目录中定义一个传感器来接收车载温度值。该传感器可用于表示两辆车中的热电偶。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 信号目录](#)。

## 车辆型号（模型清单）

车辆型号是声明性结构，可用于标准化车辆形式并定义车辆中信号之间的关系。车辆型号要求同一类型的多辆车辆具有一致的信息。您可以添加信号来创建车辆型号。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆模型](#)。

## 解码器清单

解码器清单包含车辆型号中每个信号的解码信息。车辆中的传感器和执行器传输低级消息（二进制数据）。借助解码器清单，物 AWS 联网 FleetWise 能够将二进制数据转换为人类可读的值。每个解码器清单都与车辆型号相关联。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

## 网络接口

包含有关车载网络使用的协议的信息。AWS 物联网 FleetWise 支持以下协议。

### 控制器局域网（CAN 总线）

一种协议，用于定义电子控制单元之间如何传输数据（ECUs）。ECUs 可以是发动机控制单元、安全气囊或音频系统。

## 板载诊断 (OBD) II

一种进一步开发的协议，用于定义相互之间 ECUs 如何传输自我诊断数据。它提供了许多标准的诊断故障代码 (DTCs)，可帮助识别您的车辆出了什么问题。

### 车辆中间件

车辆中间件定义为一种网络接口。车辆中间件的示例包括机器人操作系统 (ROS 2) 和可扩展的面向服务的 IP 中间件 (SOME/IP)。

#### Note

AWS 物联网 FleetWise 支持用于视觉系统数据的 ROS 2 中间件。

### 自定义接口

您也可以使用自己的接口来解码边缘的信号。这可以节省您的时间，因为您无需在云中创建解码规则。

### 信号解码器

提供特定信号的详细解码信息。车辆模型中指定的每个信号都必须与信号解码器配对。如果解码器清单包含 CAN 网络接口，则它必须包含 CAN 解码器信号。如果解码器清单包含 OBD 网络接口，则必须包含 OBD 信号解码器。

如果解码器清单还包含车辆中间件接口，则它必须包含消息信号解码器。或者，如果解码器清单包含自定义解码接口，则它还必须包含自定义解码信号。

### 车辆

您的实体车辆（例如汽车或卡车）的虚拟表示。车辆是车辆型号的实例。使用相同车辆型号创建的车辆会继承相同的一组信号。每辆车对应一个 AWS IoT 物品。

### 车队

车队表示一组车辆。在轻松管理车队之前，必须将各个车辆与车队关联起来。

### 活动

包含数据收集方案。您可以在云端定义活动并将其部署到车辆或车队中。活动会向边缘代理软件发送有关如何选择、收集数据并将其传输到云端的指令。

## 数据分区

在活动中配置分区数据以临时存储信号数据。您可以配置何时以及如何将数据转发到云端。

## 数据收集方案

数据收集方案为边缘代理软件提供有关如何收集数据的说明。目前，AWS 物联网 FleetWise 支持基于条件的收集方案和基于时间的收集方案。

### 基于条件的收集方案

使用逻辑表达式来识别要收集哪些数据。满足条件时，边缘代理软件会收集数据。例如，如果表达式为 `$variable.myVehicle.InVehicleTemperature >35.0`，则边缘代理软件会收集大于 35.0 的温度值。

### 基于时间的收集方案

指定用于确定收集数据的频率的时间段，以毫秒为单位。例如，如果时间段为 10,000 毫秒，则边缘代理软件会每隔 10 秒收集一次数据。

## 命令

命令从云端对车辆执行命令。您可以远程向车辆发送命令，车辆将在几秒钟内执行命令。例如，您可以配置命令来锁定车辆的车门或设置温度。

该命令是由管理的资源 AWS IoT Device Management。它包含可重复使用的配置，这些配置在向车辆发送命令执行时适用。有关更多信息，请参阅《AWS IoT Core 开发人员指南》中的[AWS IoT 命令](#)。

## 状态模板

状态模板为车主提供了一种跟踪其车辆状态的机制。在车辆上运行的 Edge Agent 软件代理收集信号更新并将其发送到云端。每个状态模板都包含从中收集数据的信号列表。

## AWS 物联网的特点 FleetWise

以下是 AWS 物联网的主要功能 FleetWise。

### 车辆建模

创建车辆的虚拟表示，并应用通用格式来组织车辆信号。AWS 物联网 FleetWise 支持[车辆信号规范 \(VSS\)](#)，可用于标准化车辆信号。

## 基于计划的数据收集

定义仅将高价值车辆数据传输到云端的方案。您可以定义基于条件的方案来控制要收集的数据，例如高于 40 度的车内温度值。您还可以定义基于时间的方案，以控制收集数据的频率。

### 适用于 AWS 物联网 FleetWise 软件的边缘代理

在车辆中运行的边缘代理软件可推动车辆与云端之间的通信。当车辆连接到云端时，边缘代理软件会持续接收数据收集方案并相应地收集数据。

## AWS 物联网中的区域和功能可用性 FleetWise

有关支持 AWS IoT 的 AWS 区域列表 FleetWise，请参阅[物联网 AWS 网 FleetWise 终端节点和配额](#)。AWS 物联网 FleetWise 功能的区域支持有所不同。

### Note

目前，亚太地区（孟买）地区和某些 AWS 物联网 FleetWise 功能的访问受到限制。要申请访问该 AWS 地区和所有封闭功能，请联系您的账户经理或[AWS Support Center](#)。

下表显示了按区域划分的功能支持：

功能/区域	美国东部（弗吉尼亚州北部）	欧洲地区（法兰克福）	亚太地区（孟买）注意：仅限门禁
<a href="#">信号目录</a>	支持	是	受限
<a href="#">车辆型号</a>	支持	是	受限
<a href="#">解码器清单</a>	支持	是	受限
<a href="#">车辆</a>	支持	是	受限
<a href="#">舰队</a>	支持	是	受限
<a href="#">活动</a>	支持	是	受限
<a href="#">视觉系统数据（预览版）</a>	支持	是	受限

功能/区域	美国东部 ( 弗吉尼亚州北部 )	欧洲地区 ( 法兰克福 )	亚太地区 ( 孟买 ) 注意：仅限门禁
<a href="#">MQTT 主题作为活动数据目的地</a>	受限	受限	受限
<a href="#">存储并转发</a>	受限	受限	受限
<a href="#">命令</a>	受限	受限	受限
<a href="#">最后一个已知状态</a>	受限	受限	受限
<a href="#">使用自定义解码接口收集与网络无关的数据</a>	受限	受限	受限
<a href="#">获取诊断故障代码 (DTC) *</a>	受限	受限	受限

\*DTC 提取提供了一系列超出基本 DTC 数据检索的功能。此功能包括自定义功能，使您能够在边缘定义函数，并在基于条件的广告系列表达式中按名称调用它们。此外，它还支持无界字符串的集合，从而提供灵活的字符串数据类型处理。Edge Agent 可以定期提取数据，也可以根据特定条件触发获取数据，从而增强其在数据收集过程中的适应性和效率。有关更多信息，请参阅 [Edge Agent 开发人员指南](#) 中的 [自定义函数指南](#) 和 [DTC 数据收集参考实现](#)。

# 设置 AWS 物联网 FleetWise

在首次使用 AWS IoT FleetWise oT 之前，请完成以下各节中的步骤。

## 主题

- [设置你的 AWS 账户](#)
- [开始使用控制台](#)
- [配置您的 AWS 物联网 FleetWise 设置](#)
- [FleetWise 使用向 AWS 物联网发出请求 IPv6](#)

## 设置你的 AWS 账户

完成以下任务以注册 AWS 并创建管理用户。

### 注册获取 AWS 账户

如果您没有 AWS 账户，请完成以下步骤来创建一个。

#### 报名参加 AWS 账户

1. 打开<https://portal.aws.amazon.com/billing/注册>。
2. 按照屏幕上的说明操作。

在注册时，将接到电话或收到短信，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建 AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务和资源。作为最佳安全实践，请为用户分配管理访问权限，并且只使用根用户来执行[需要根用户访问权限的任务](#)。

AWS 注册过程完成后会向您发送一封确认电子邮件。您可以随时前往 <https://aws.amazon.com/> 并选择“我的账户”，查看您当前的账户活动并管理您的账户。

### 创建具有管理访问权限的用户

注册后，请保护您的安全 AWS 账户 AWS 账户根用户 AWS IAM Identity Center，启用并创建管理用户，这样您就可以不会使用 root 用户执行日常任务。

## 保护你的 AWS 账户根用户

1. 选择 Root 用户并输入您的 AWS 账户 电子邮件地址，以账户所有者的身份登录。[AWS 管理控制台](#)在下一页上，输入您的密码。

要获取使用根用户登录方面的帮助，请参阅《AWS 登录 用户指南》中的 [Signing in as the root user](#)。

2. 为您的根用户启用多重身份验证 ( MFA )。

有关说明，请参阅 [IAM 用户指南中的为 AWS 账户 根用户启用虚拟 MFA 设备 \( 控制台 \)](#)。

## 创建具有管理访问权限的用户

1. 启用 IAM Identity Center。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 [启用 AWS IAM Identity Center](#)。

2. 在 IAM Identity Center 中，为用户授予管理访问权限。

有关使用 IAM Identity Center 目录 作为身份源的教程，请参阅《[用户指南](#)》[IAM Identity Center 目录中的使用默认设置配置AWS IAM Identity Center 用户访问权限](#)。

## 以具有管理访问权限的用户身份登录

- 要使用您的 IAM Identity Center 用户身份登录，请使用您在创建 IAM Identity Center 用户时发送到您的电子邮件地址的登录 URL。

有关使用 IAM Identity Center 用户[登录的帮助](#)，请参阅[AWS 登录 用户指南中的登录 AWS 访问门户](#)。

## 将访问权限分配给其他用户

1. 在 IAM Identity Center 中，创建一个权限集，该权限集遵循应用最低权限的最佳做法。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 [Create a permission set](#)。

2. 将用户分配到一个组，然后为该组分配单点登录访问权限。

有关说明，请参阅《AWS IAM Identity Center 用户指南》中的 [Add groups](#)。

**Note**

您可以在 AWS IoT FleetWise 中使用服务相关角色。服务相关角色由 AWS 物联网预定义 FleetWise ，包括 AWS 物联网向亚马逊 CloudWatch 发送指标 FleetWise 所需的权限。有关更多信息，请参阅 [在 AWS IoT 中使用服务相关角色 FleetWise](#)。

## 开始使用控制台

如果您尚未登录您的，请登录 AWS 账户，然后打开 [AWS IoT FleetWise 控制台](#)。要开始使用 AWS 物联网 FleetWise，请创建车辆模型。车辆型号可以标准化您的车辆格式。

1. 打开 [AWS 物联网 FleetWise 控制台](#)。
2. 在“入门”中 AWS IoT FleetWise，选择“开始”。

有关创建车辆型号的更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。

## 配置您的 AWS 物联网 FleetWise 设置

您可以使用 AWS 物联网 FleetWise 控制台或 API 来配置亚马逊 CloudWatch 日志指标和亚马逊 CloudWatch 日志的设置，并使用加密数据 AWS 托管式密钥。

借助 CloudWatch 指标，您可以监控 AWS 物联网 FleetWise 和其他 AWS 资源。您可以使用 CloudWatch 指标来收集和跟踪指标，例如确定是否超过了服务限制。有关 CloudWatch 指标的更多信息，请参阅 [FleetWise 使用 Amazon 监控 AWS 物联网 CloudWatch](#)。

通过 CloudWatch 日志，AWS 物联网 FleetWise 将日志数据发送到 CloudWatch 日志组，您可以在其中使用它来识别和缓解任何问题。有关 CloudWatch 日志的更多信息，请参阅 [配置 AWS 物联网 FleetWise 日志](#)。

通过数据加密，AWS 物联网 FleetWise 使用 AWS 托管式密钥 来加密数据。您也可以选择使用创建和管理密钥 AWS KMS。有关加密的更多信息，请参阅 [AWS 物联网中的数据加密 FleetWise](#)。

## 配置设置（控制台）

如果您尚未登录您的，请登录 AWS 账户，然后打开 [AWS IoT FleetWise 控制台](#)。

1. 打开 [AWS 物联网 FleetWise 控制台](#)。
2. 在左侧窗格中，选择设置。

3. 在指标中，选择启用。AWS IoT FleetWise 会自动将 CloudWatch 托管策略附加到服务相关角色并启用 CloudWatch 指标。
4. 在日志记录中，选择编辑。
  - a. 在 CloudWatch 日志记录部分，输入日志组。
  - b. 选择提交可保存更改。
5. 在加密部分，选择编辑。
  - a. 选择要使用的密钥类型。有关更多信息，请参阅 [AWS 物联网中的密钥管理 FleetWise](#)。
    - i. 使用 AWS 密钥 — AWS 物联网 FleetWise 拥有并管理密钥。
    - ii. 选择其他 AWS Key Management Service 密钥 — 由 AWS KMS keys 您管理账户中的密钥。
  - b. 选择提交可保存更改。

## 配置设置 (AWS CLI)

在中 AWS CLI，注册账户以配置设置。

### 账户注册的 IAM 权限设置

要成功调用 RegisterAccount API，您需要在 IAM 策略文档 iam:CreateServiceLinkedRole 中包含内容。此 API 会在您的账户中创建服务相关角色，用于向您的账户发布 AWS IoT FleetWise 指标。CloudWatch 要验证账户是否成功注册，请调用 GetRegisterAccountStatus API 并确保注册状态为 REGISTRATION\_SUCCESS。

以下示例显示了用于设置 RegisterAccount 和权限的策略文档示例 GetRegisterAccountStatus：

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

```
    "Resource": [  
      "*"   
    ]   
  }   
]   
}
```

1. 要配置设置，请运行以下命令。

```
aws iotfleetwise register-account
```

2. 要验证您的设置，请运行以下命令来检索注册状态。

#### Note

服务相关角色仅用于向发布 AWS 物联网 FleetWise 指标。CloudWatch 有关更多信息，请参阅 [在 AWS IoT 中使用服务相关角色 FleetWise](#)。

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

### Example 响应

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

注册状态可以是下列项之一：

- REGISTRATION\_SUCCESS— AWS 资源已成功注册。

- REGISTRATION\_PENDING— AWS 物联网 FleetWise 正在处理注册请求。该过程可能需要五分钟才能完成。
- REGISTRATION\_FAILURE— AWS 物联网 FleetWise 无法注册 AWS 资源。请稍后重试。

## FleetWise 使用向 AWS 物联网发出请求 IPv6

您可以通过 FleetWise 互联网协议版本 6 与 AWS IoT 进行通信 IPv4 并管理您的资源。IPv6 双栈端点支持通过 IPv6 和 IPv4 向 AWS 物联网发 FleetWise APIs 出的请求。通过通信不收取任何额外费用 IPv6。

该 IPv6 协议是具有其他安全功能的下一代 IP 标准。它提供 128 位长地址空间，同时 IPv4 具有 32 位长地址。IPv4 可以生成  $4.29 \times 10^9$  个地址，而 IPv6 可以有  $3.4 \times 10^{38}$  个地址。

### IPv6 控制平面端点的先决条件

IPv6 将自动为控制平面端点启用协议支持。使用控制面板客户端的端点时，必须提供[服务器名称指示 \(SNI\) 扩展名](#)。客户端可以使用 SNI 扩展来指示正在联系的服务器的名称，以及它使用的是常规端点还是双栈端点。请参阅[使用双堆栈端点](#)。

### IPv6 对 AWS PrivateLink 端点的支持

AWS 物联网 FleetWise 支持使用与接口 VPC 终端节点进行 IPv6 通信 AWS PrivateLink。

### 测试 IPv6 地址兼容性

如果您使用的是 use Linux/Unix 或 Mac OS X，则可以使用 curl 命令测试是否可以访问双栈端点，如下示例所示：IPv6

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

您获得的信息如下例所示。如果您是通过连接的 IPv6，则连接的 IP 地址将是一个 IPv6 地址。

```
* 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
```

如果你使用的是微软 Windows 7 或 Windows 10，你可以测试是否 IPv4 可以通过 IPv6 或使用 ping 命令访问双栈端点，如下例所示。

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

## 在 IAM 策略中使用 IPv6 地址

在使用资源之前，必须确保用 IPv6 于 IP 地址筛选的任何 IAM 策略都包含 IPv6 地址范围。有关使用 IAM 管理访问权限的更多信息，请参阅 [适用于 AWS 物联网的 Identity and Access 管理 FleetWise](#)。

筛选 IP 地址的 IAM 策略使用 [IP 地址条件运算符](#)。以下策略使用 IP IPv4 地址条件运算符确定允许的地址 54.240.143.\* 范围。由于所有 IPv6 地址都超出了允许的范围，因此此策略禁止使用 IPv6 地址进行通信。

### 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"}
      }
    }
  ]
}
```

要包括 IPv6 地址，您可以修改策略的条件元素以同时允许 IPv4 (54.240.143.0/24) 和 IPv6 (2001::1234:5678::DB8:/64) 地址范围，如以下示例所示。

```
"Condition": {
  "IpAddress": {
    "aws:SourceIp": [
```

```

    "54.240.143.0/24",
    "2001:DB8:1234:5678::/64"
  ]
}
}

```

## 使用双堆栈端点

AWS 物联网 FleetWise 双栈端点支持通过 IPv6 和 IPv4 向 AWS 物联网 FleetWise APIs 发出请求。当您向双栈终端节点发出请求时，它会自动解析为 IPv4 或 IPv6 地址。在双栈模式下，同时接受 IPv4 和 IPv6 客户端连接。

如果您使用的是 REST API，则可以使用 FleetWise 终端节点名称 (URI) 直接访问 AWS 物联网终端节点。AWS IoT 仅 FleetWise 支持区域双栈终端节点名称，这意味着您必须在名称中指定 AWS 区域。

下表显示了使用 FleetWise 时 AWS 物联网控制平面端点的格式 IPv4 和双栈模式。有关这些终端节点的更多信息，请参阅[AWS 物联网 FleetWise 终端节点](#)。

端点	IPv4 地址	双栈模式
控制面板	iotfleetwise. <region>.amazonaws.co	iotfleetwise. <region>.api.aws

使用 AWS CLI 和 AWS SDKs，您可以使用 `AWS_USE_DUALSTACK_ENDPOINT` 环境变量或 `use_dualstack_endpoint` 参数（共享配置文件设置）更改为双堆栈端点。您也可以在配置文件中直接将双栈端点指定为 AWS 物联网 FleetWise 端点的替代项。有关更多信息，请参阅 [Dual-stack and FIPS endpoints](#)。

使用时 AWS CLI，您可以像在 Config 文件 `true` 中的配置文件中 `use_dualstack_endpoint` 一样设置 AWS 配置值。这会将命令发出的所有 AWS IoT FleetWise 请求定向到指定区域的双栈终端节点。您可以在配置文件或命令中使用 `--region` 选项指定区域。

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

与其对所有命令使用双栈端点，不如使用这些端点执行特定命令：

- 您可以通过为特定命令设置 `--endpoint-url` 参数来使用双栈端点。例如，在以下命令中，您可以将替换 `<endpoint-url>` 为 `iotfleetwise.<region>.api.aws`。

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

- 您可以在 Config 文件中设置单独的 AWS 配置文件。例如，创建一个设置为 true 的 `use_dualstack_endpoint` 的配置文件和一个未设置的配置文件 `use_dualstack_endpoint`。在运行命令时，根据是否需要使用双堆栈端点来指定要使用的配置文件。

# 教程：AWS 物联网入门 FleetWise

借助 AWS 物联网 FleetWise，您可以收集、转换和传输车辆数据。使用本节中的教程开始使用 AWS IoT FleetWise。

要了解有关 AWS IoT 的更多信息，请参阅以下主题 FleetWise：

- [将 AWS 物联网 FleetWise 数据摄取到云端](#)
- [对 AWS 物联网 FleetWise 车辆进行建模](#)
- [管理 AWS 物联网 FleetWise 车辆](#)
- [在物联网中 AWS 管理车队 FleetWise](#)
- [通过活动收集 AWS 物联网 FleetWise 数据](#)

## 简介

使用 AWS 物联网 FleetWise 近乎实时地收集、转换自动驾驶汽车的独特数据格式并将其传输到云端。您可以访问车队范围的见解。这可以帮助您高效地检测和缓解车辆运行状况问题，高效地传输高价值数据信号和远程诊断问题，同时还能降低成本。

本教程向您展示如何开始使用 AWS 物联网 FleetWise。您将了解如何创建车辆型号（模型清单）、解码器清单、车辆和活动。

有关 AWS IoT 的关键组件和概念的更多信息 FleetWise，请参阅[AWS 物联网的主要概念和特点 FleetWise](#)。

估计时间：约 45 分钟。

### Important

您需要为本演示创建和使用的 AWS 物联网 FleetWise 资源付费。有关更多信息，请参阅 [AWS IoT FleetWise](#) 定价页面 FleetWise 中的 AWS 物联网。

## 先决条件

要完成本入门教程，首先需要满足以下条件：

- 一个 AWS 账户。如果您没有 AWS 账户，请参阅《AWS 账户管理 参考指南》AWS 账户中的[创建](#)。
- 访问支持 AWS 物联网 FleetWise 的。AWS 区域 目前，美国东部（弗吉尼亚北部）和欧洲（法兰克福）支持 AWS 物联网 FleetWise。您可以使用中的区域选择器切换 AWS 管理控制台 到其中一个区域。有关更多信息，请参阅 [AWS IoT FleetWise 终端节点和配额](#)。
- Amazon Timestream 资源：
  - Amazon Timestream 数据库。有关更多信息，请参阅《Amazon Timestream 开发人员指南》中的[创建数据库](#)。
  - 在 Amazon Timestream 中创建的用于保存数据的 Amazon Timestream 表。有关更多信息，请参阅《Amazon Timestream 开发人员指南》中的[创建表](#)。
- 边缘代理软件演示。（设置演示的说明将在下一步中介绍。）
  - 您可以使用 Explore Edge Agent 快速入门演示来探索 AWS 物联网 FleetWise 并学习如何为 AWS 物联网开发边缘代理软件 FleetWise。此演示使用 CloudFormation 模板。它将引导您查看边缘代理软件参考实现，开发边缘代理，然后在 Amazon EC2 Graviton 上部署边缘代理软件并生成车辆数据示例。该演示还提供了一个脚本，您可以使用该脚本在云端创建信号目录、车辆模型、解码器清单、车辆、车队和活动。
  - 要下载演示，请导航到 [AWS IoT FleetWise 控制台](#)。在服务主页上的“AWS 物联网入门”FleetWise 部分，选择“探索 Edge Agent”。

## 步骤 1：为 AWS 物联网设置边缘代理软件 FleetWise

### Note

此步骤中的 CloudFormation 堆栈使用遥测数据。您也可以使用视觉系统数据创建 CloudFormation 堆栈。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。视觉系统数据目前为预览版，可能会发生变化。


适用于 AWS 物联网的 Edge Agent 软件 FleetWise 可促进车辆与云之间的通信。该软件接收来自数据收集方案的说明如何从联网车辆收集数据的指令。




要设置边缘代理软件，请在常规信息中执行以下操作：

1. 打开[启动 CloudFormation 模板](#)。

- 在快速创建堆栈页面上，在堆栈名称中输入您的 AWS 物联网 FleetWise 资源堆栈的名称。堆栈是一个友好的名称，它以前缀形式出现在此 CloudFormation 模板创建的资源名称上。
- 在参数下，输入与堆栈相关的参数的自定义值。
  - Fleetsize - 可以通过更新 Fleetsize 参数来增加车队中的车辆数量。
  - Io TCore 区域-您可以通过更新 Io Region 参数来指定创建 AWS IoT 事物的 TCore 区域。您必须使用与创建 AWS 物联网 FleetWise 车辆相同的区域。有关更多信息 AWS 区域，请参阅[区域和区域-Amazon Elastic Compute Cloud](#)。
- 在功能部分中，选中复选框以确认 CloudFormation 创建了 IAM 资源。
- 选择创建堆栈，然后等待大约 15 分钟，等到堆栈的状态显示为 CREATE\_COMPLETE。
- 要确认堆栈已创建，请选择堆栈信息选项卡，刷新视图，然后查找 CREATE\_COMPLETE。

fwdemo

Overview 

Stack ID	Description
 <a href="#">arn:aws:cloudformation:us-east-1:012345678912:stack/fwdemo/bd04af20-a269-11ed-bf1d-0a56266679b7</a> 	-
Status	Status reason
 CREATE_COMPLETE	-

### Important

您需要为本演示创建和使用的 AWS 物联网 FleetWise 资源付费。有关更多信息，请参阅 [AWS IoT FleetWise](#) 定价页面 FleetWise 中的 AWS 物联网。

## 步骤 2：创建车辆型号

### Important

您无法在 AWS 物联网 FleetWise 控制台中创建带有视觉系统数据信号的车辆模型。可改用 AWS CLI。

可以使用车辆模型来标准化车辆的格式，并帮助定义所创建的车辆中信号之间的关系。创建车辆模型时还会创建信号目录。信号目录是标准化信号的集合，可以重用这些目录来创建车辆型号。。信号是用于定义是否包含车辆数据及其元数据的基本结构。目前，AWS 物联网 FleetWise 服务 AWS 区域 每个账户仅支持一个信号目录。这有助于确保处理的车队数据保持一致。

## 创建车辆型号

1. 打开 AWS 物联网 FleetWise 控制台。
2. 在导航窗格中，选择车辆型号。
3. 在车辆型号页面上，选择创建车辆型号。
4. 在一般信息部分，输入您的车辆型号的名称（例如 Vehicle1）和可选描述。然后选择下一步。
5. 从信号目录中选择一个或多个信号。您可以在搜索目录中按名称筛选信号，也可以从列表中选择信号。例如，您可以选择轮胎压力和制动压力信号，以便收集与这些信号相关的数据。选择下一步。
6. 选择您的 .dbc 文件并从本地设备上传。选择下一步。

### Note

根据本教程，可以下载 [.dbc 文件示例](#) 以供在此步骤中上传。

7. 在车辆型号中添加属性，然后选择下一步。
  - a. 名称 - 输入车辆属性的名称，例如制造商名称或制造日期。
  - b. 数据类型 - 在数据类型菜单上，选择一种数据类型。
  - c. 单位 - (可选) 输入单位值，例如千米或摄氏度。
  - d. 路径 - (可选) 输入信号路径的名称，例如 Vehicle.Engine.Light.点(.)表示它是子信号。
  - e. 默认值 - (可选) 输入默认值。
  - f. 描述 - (可选) 输入属性的描述。
8. 审核您的配置。当您准备好后，选择 Create (创建)。系统将显示一条通知，说明您的车辆型号已成功创建。

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

AWS IoT FleetWise > Vehicle models > Demo

## demo

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

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

### Summary [Info](#)

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

## 步骤 3：创建解码器清单

解码器清单与您创建的车型号关联。它们包含有助于 AWS 物联网 FleetWise 解码车辆数据并将其从二进制格式转换为可供分析的人类可读值的信息。网络接口和解码器信号是帮助配置解码器清单的组件。网络接口包含有关您的车辆网络使用的 CAN 或 OBD 协议的信息。解码器信号为特定信号提供解码信息。

### 创建解码器清单

1. 打开 AWS 物联网 FleetWise 控制台。
2. 在导航窗格中，选择车型号。
3. 在车型号部分，选择要用于创建解码器清单的车型号。
4. 选择创建解码器清单。

## 步骤 4：配置解码器清单

### 配置解码器清单

#### Important

您无法使用物 AWS 联 FleetWise 网控制台在解码器清单中配置视觉系统数据信号。可改用 AWS CLI。有关更多信息，请参阅 [创建解码器清单 \(AWS CLI\)](#)。

1. 为了帮助您识别解码器清单，请为其输入名称和可选描述。然后选择下一步。
2. 要添加一个或多个网络接口，请选择 CAN\_INTERFACE 或 OBD\_INTERFACE 类型。
  - 板载诊断 (OBD) 接口-如果您想要一个定义电子控制单元之间如何通信自诊断数据的协议，请选择此接口类型 ( ) ECUs。该协议提供了许多标准的诊断故障代码 (DTCs)，可以帮助您解决车辆问题。
  - 控制器局域网 (CAN 总线) 接口-如果您想要一个定义数据之间如何通信的协议，请选择此接口类型 ECUs。ECUs 可以是发动机控制单元、安全气囊或音频系统。
3. 输入网络接口的名称。
4. 要向网络接口添加信号，请从列表选择一个或多个信号。
5. 为在上一步中添加的信号选择解码器信号。要提供解码信息，请上传 .dbc 文件。车辆模型中的每个信号都必须与解码器信号 ( 可从列表中选择 ) 配对。
6. 要添加另一个网络接口，请选择添加网络接口。添加网络接口后，选择下一步。
7. 检查配置，然后选择创建。系统将显示一条通知，说明您的解码器清单已成功创建。

## 步骤 5：创建车辆

在 AWS 物联网中 FleetWise，车辆是现实生活中的实体车辆的虚拟代表。使用相同车辆模型创建的所有车辆都继承一组相同的信号，并且您创建的每个车辆都对应于一个新创建的 IoT 事物。您必须将所有车辆与解码器清单相关联。

### 先决条件

1. 确认您已经创建了车辆模型和解码器清单。此外，还要确认车辆模型的状态为活动。
  - a. 要验证车辆模型的状态是否为激活，请打开 AWS 物联网 FleetWise 控制台。

- b. 在导航窗格中，选择车辆型号。
- c. 在摘要部分的状态下，查看您的车辆状态。

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

AWS IoT FleetWise > Vehicle models > Demo

## demo

Duplicate Create vehicle Create decoder manifest

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

### Summary [Info](#)

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

## 创建车辆

1. 打开 AWS FleetWise 控制台。
2. 在导航窗格中，选择车辆。
3. 选择创建车辆。
4. 要定义车辆属性，请输入车辆名称，然后选择模型清单（车辆型号）和解码器清单。
5. （可选）要定义车辆属性，请输入键值对，然后选择添加属性。
6. （可选）要为您的 AWS 资源添加标签，请添加标签，然后选择添加新标签。
7. 选择下一步。
8. 要配置车辆证书，您可以上传自己的证书，也可以选择自动生成新证书。建议您自动生成证书，以便更快地进行设置。如果您已经有证书，可以选择改用您的证书。
9. 下载公钥和私钥文件，然后选择下一步。
10. 要将策略附加到车辆证书，可以输入现有策略名称或创建新策略。要创建新策略，请选择创建策略，然后选择下一步。
11. 审核您的配置。完成后，选择创建车辆。

## 步骤 6：创建活动

在 AWS 物联网中 FleetWise，活动用于促进数据从车辆的选择、收集和传输到云端。活动包含数据收集方案，这些方案向边缘代理软件提供有关如何使用基于条件的收集方案或基于时间的收集方案收集数据的指令。

### 创建市场活动

1. 打开 AWS 物联网 FleetWise 控制台。
2. 在导航窗格中，选择活动。
3. 选择创建市场活动。
4. 输入活动名称和可选描述。
5. 要配置活动的数据收集方案，可以手动定义数据收集方案，或者从本地设备上传 .json 文件。上传 .json 文件可自动定义数据收集方案。
  - a. 要手动定义数据收集方案，请选择定义数据收集方案，然后选择要用于活动的数据收集方案类型。您可以选择基于条件的收集方案或基于时间的收集方案。
  - b. 如果选择基于时间的收集方案，则必须指定活动收集车辆数据的时长。
  - c. 如果选择基于条件的数据收集方案，则必须指定一个表达式来识别要收集哪些数据。确保将信号的名称指定为变量、比较运算符和比较值。
  - d. ( 可选 ) 选择表达式的语言版本，或将其保留为默认值 1。
  - e. ( 可选 ) 指定两个数据收集事件之间的触发间隔。
  - f. 要收集数据，请选择边缘代理软件的触发模式条件。默认情况下，适用于 AWS 物联网的 Edge Agent FleetWise 软件始终在满足条件时收集数据。或者，它只在第一次满足条件时收集数据，即首次触发时。
  - g. ( 可选 ) 可以选择更多高级方案选项。
6. 要指定数据收集方案将从中收集数据的信号，请从菜单中搜索信号的名称。
7. ( 可选 ) 可以选择最大样本数或最小采样间隔。您还可以添加更多信号。
8. 选择下一步。
9. 定义您希望活动将数据传输到的存储目标。您可以将数据存储到 Amazon S3 或 Amazon Timestream 中。
  - a. Amazon S3 — 选择 AWS IoT FleetWise 有权访问的 S3 存储桶。
  - b. Amazon Timestream - 选择 Timestream 数据库和表名。输入允许向 Timestream AWS IoT FleetWise 发送数据的 IAM 角色。

10. 选择下一步。
11. 从搜索框中选择车辆属性或车辆名称。
12. 输入与您为车辆选择的属性或名称相关的值。
13. 选择您的活动将从中收集数据的车辆。然后选择下一步。
14. 检查活动配置，然后选择创建活动。您或您的团队必须将活动部署到车辆上。

## 步骤 7：清除

为避免对您在本教程中使用的资源收取更多费用，请删除 CloudFormation 堆栈和所有堆栈资源。

### 删除 CloudFormation 堆栈

1. 打开 [CloudFormation 控制台](#)。
2. 从堆栈列表中，选择您在步骤 1 中创建的堆栈。
3. 选择删除。
4. 要确认删除，选择 Delete (删除)。删除堆栈约需 15 分钟时间。

## 后续步骤

1. 您可以处理和可视化活动收集到的车辆数据。有关更多信息，请参阅 [可视化 AWS 物联网 FleetWise 车辆数据](#)。
2. 您可以对 AWS 物联网问题进行故障排除和解决 FleetWise。有关更多信息，请参阅 [AWS 物联网故障排除 FleetWise](#)。

# 将 AWS 物联网 FleetWise 数据摄取到云端

## Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

适用于 AWS 物联网的 Edge Agent FleetWise 软件在车辆中安装和运行时，旨在促进车辆与云之间的安全通信。

## Note

- AWS 物联网 FleetWise 不适用于任何可能导致严重人身伤害或死亡或造成环境或财产损失的危险环境或关键系统的运行，或与之相关联。通过您使用 AWS 物联网收集的车辆数据 FleetWise 仅供参考，您不得使用 AWS 物联网 FleetWise 来控制或操作车辆功能。
- FleetWise 应根据您的用例评估通过您使用 AWS 物联网收集的车辆数据的准确性，包括为了履行适用的车辆安全法规（例如安全监控和报告义务）下可能承担的任何合规义务。此类评估应包括通过其他行业标准手段和来源（例如车辆驾驶员的报告）收集和审查信息。

要将数据摄取到云，请执行以下操作：

1. 开发并安装适用于 AWS 物联网的 Edge Agent FleetWise 软件，并将其安装到您的车辆中。有关如何使用 Edge Agent 软件的更多信息，请执行以下操作下载 [适用于 AWS IoT 的 Edge Agent FleetWise 软件开发人员指南](#)。
  1. 导航到 [AWS 物联网 FleetWise 控制台](#)。
  2. 在服务主页上的“AWS 物联网入门”FleetWise 部分，选择“探索 Edge Agent”。
2. 创建或导入信号目录，其中包含用于创建车辆型号的信号。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 信号目录](#) 和 [导入信号目录 \(AWS CLI\)](#)。

## Note

- 如果您使用 AWS 物联网 FleetWise 控制台创建第一个车辆模型，则无需手动创建信号目录。当您创建第一个车辆模型时，AWS IoT FleetWise 会自动为您创建信号目录。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。

- AWS 物联网 FleetWise 目前支持每个 AWS 账户的信号目录 AWS 区域。

3. 使用信号目录中的信号创建车辆型号。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。

**Note**

- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，则可以上传 .dbc 文件以导入信号。 .dbc 是控制器局域网 (CAN 总线) 数据库支持的文件格式。创建车辆型号后，新的信号会自动添加到信号目录中。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。
- 如果您使用 CreateModelManifest API 操作创建车辆型号，则必须使用 UpdateModelManifest API 操作来激活车辆型号。有关更多信息，请参阅 [更新 AWS 物联网 FleetWise 车辆模型](#)。
- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，AWS IoT FleetWise 会自动为您激活车辆模型。

4. 创建解码器清单。解码器清单包含您在上一步中创建的车型号中指定的每个信号的解码信息。解码器清单与您创建的车型号关联。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

**Note**

- 如果您使用 CreateDecoderManifest API 操作创建解码器清单，则必须使用 UpdateDecoderManifest API 操作来激活解码器清单。有关更多信息，请参阅 [更新 AWS IoT FleetWise 解码器清单](#)。
- 如果您使用 AWS IoT FleetWise 控制台创建解码器清单，AWS IoT FleetWise 会自动为您激活解码器清单。

5. 根据车辆型号创建车辆。使用相同车辆型号创建的车辆会继承相同的一组信号。必须先使用 AWS IoT Core 配置车辆，然后才能将数据采集到云端。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆](#)。

6. ( 可选 ) 创建代表一组车辆的车队，然后将单个车辆与该车队相关联。这可以帮助您同时管理多辆车。有关更多信息，请参阅 [在物联网中 AWS 管理车队 FleetWise](#)。

7. ( 可选 ) 创建广告活动。活动会部署到车辆或车队。活动会向边缘代理软件发送有关如何选择、收集数据并将其传输到云端的指令。有关更多信息，请参阅 [通过活动收集 AWS 物联网 FleetWise 数据](#)。您可以创建广告系列、状态模板 ( 见下文 ) ，或者同时创建两者来收集数据。

**Note**

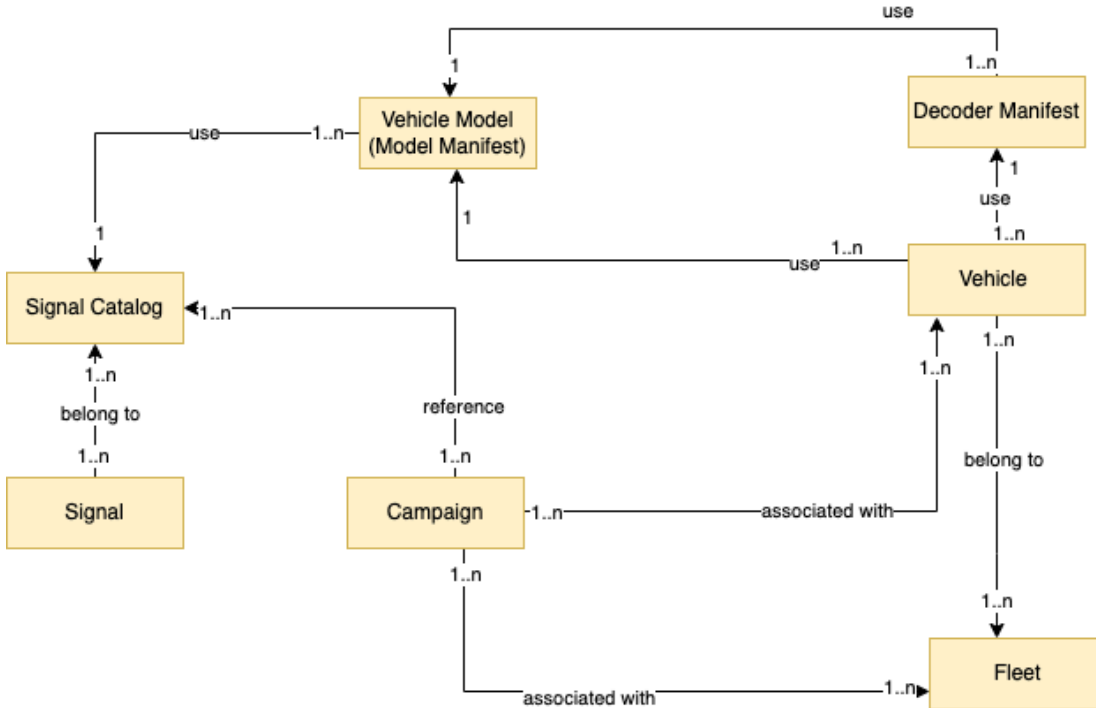
在 AWS 物联网 FleetWise 将其部署到车辆或车队之前，您必须使用 UpdateCampaign API 操作批准活动。有关更多信息，请参阅 [更新物 AWS 联网 FleetWise 活动](#)。

8. ( 可选 ) 创建状态模板。状态模板部署到车辆上。状态模板为车主提供了一种跟踪其车辆状态的机制。有关更多信息，请参阅 [监控车辆的最近已知状态](#)。

Edge Agent 软件 AWS IoT Core 使用您选择的 MQTT 主题将车辆数据传输到。要将活动数据发送到 AWS IoT FleetWise oT，它会使用保留的主题 `$aws/iotfleetwise/vehicles/vehicleName/signals`。对于上次已知状态，Edge Agent 使用保留的主题 `$aws/iotfleetwise/vehicles/vehicleName/last_known_states/data`。有关如何处理摄取数据的更多信息，请参阅 [可视化 AWS 物联网 FleetWise 车辆数据](#)。

# 对 AWS 物联网 FleetWise 车辆进行建模

AWS 物联网 FleetWise 提供了一个车辆建模框架，您可以使用该框架在云中构建车辆的虚拟表示。信号、信号目录、车辆型号和解码器清单是您用来对车辆进行建模的核心组件。



## 信号

信号是用于定义是否包含车辆数据及其元数据的基本结构。信号可以是属性、分支、传感器或执行器。例如，您可以创建一个传感器来接收车载温度值并存储其元数据，包括传感器名称、数据类型和单位。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 信号目录](#)。

## 信号目录

信号目录包含一系列信号。信号目录中的信号可用于对使用不同协议和数据格式的车辆进行建模。例如，有两辆汽车由不同的汽车制造商制造：一辆使用控制区域网络（CAN 总线）协议；另一辆使用车载诊断（OBD）协议。您可以在信号目录中定义一个传感器来接收车载温度值。该传感器可用于表示两辆车中的热电偶。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 信号目录](#)。

## 车辆型号（模型清单）

车辆型号是声明性结构，可用于标准化车辆形式并定义车辆中信号之间的关系。车辆型号要求同一类型的多辆车辆具有一致的信息。您可以添加信号来创建车辆型号。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆模型](#)。

## 解码器清单

解码器清单包含车辆型号中每个信号的解码信息。车辆中的传感器和执行器传输低级消息（二进制数据）。借助解码器清单，物 AWS 联网 FleetWise 能够将二进制数据转换为人类可读的值。每个解码器清单都与车辆型号相关联。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

您可以通过以下方式使用 AWS 物联网 FleetWise 控制台或 API 对车辆进行建模。

1. 创建或导入信号目录，其中包含用于创建车辆型号的信号。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 信号目录](#) 和 [导入信号目录 \(AWS CLI\)](#)。

### Note

- 如果您使用 AWS 物联网 FleetWise 控制台创建第一个车辆模型，则无需手动创建信号目录。当您创建第一个车辆模型时，AWS IoT FleetWise 会自动为您创建信号目录。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。
- AWS 物联网 FleetWise 目前支持每个 AWS 账户的信号目录 AWS 区域。

2. 使用信号目录中的信号创建车辆型号。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。

### Note

- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，则可以上传 .dbc 文件以导入信号。 .dbc 是控制器局域网 (CAN 总线) 数据库支持的文件格式。创建车辆型号后，新的信号会自动添加到信号目录中。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。
- 如果您使用 CreateModelManifest API 操作创建车辆型号，则必须使用 UpdateModelManifest API 操作来激活车辆型号。有关更多信息，请参阅 [更新物 AWS 联网 FleetWise 车辆模型](#)。
- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，AWS IoT FleetWise 会自动为您激活车辆模型。

3. 创建解码器清单。解码器清单包含您在上一步中创建的车辆型号中指定的每个信号的解码信息。解码器清单与您创建的车辆型号关联。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

**Note**

- 如果您使用 CreateDecoderManifest API 操作创建解码器清单，则必须使用 UpdateDecoderManifest API 操作来激活解码器清单。有关更多信息，请参阅 [更新 AWS IoT FleetWise 解码器清单](#)。
- 如果您使用 AWS IoT FleetWise 控制台创建解码器清单，AWS IoT FleetWise 会自动为您激活解码器清单。

CAN 总线数据库支持 .dbc 文件格式。您可以上传 .dbc 文件以导入信号和信号解码器。要获取 .dbc 文件示例，请执行以下操作。

获取 .dbc 文件

1. 下载 [EngineSignals.zip](#)。
2. 导航到下载了 EngineSignals.zip 文件的目录。
3. 解压缩该文件并将其作为 EngineSignals.dbc 保存在本地。

主题

- [管理 AWS 物联网 FleetWise 信号目录](#)
- [管理 AWS 物联网 FleetWise 车辆模型](#)
- [管理 AWS IoT FleetWise 解码器清单](#)

## 管理 AWS 物联网 FleetWise 信号目录

**Note**

您可以下载[演示脚本](#)，将 ROS 2 消息转换为与信号目录兼容的 VSS .json 文件。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。

信号目录是标准化信号的集合，可以重复用于创建车辆模型。AWS 物联网 FleetWise 支持[车辆信号规范 \(VSS\)](#)，您可以按照该规格来定义信号。信号可以是以下任何一种。

## 属性

属性表示通常不会变化的静态信息，例如制造商和生产日期。

## 分支

分支表示嵌套结构中的信号。分支演示信号层次结构。例如，Vehicle 分支有一个子分支 Powertrain。Powertrain 分支有一个子分支 combustionEngine。要找到 combustionEngine 分支，请使用 `Vehicle.Powertrain.combustionEngine` 表达式。

## 传感器

传感器数据报告车辆的当前状态，并随着车辆状态（例如液位、温度、振动或电压）的变化而随着时间的推移变化。

## 执行器

执行器数据报告车辆设备（例如电机、加热器和门锁）的状态。更改车辆设备的状态可以更新执行器数据。例如，您可以定义一个代表加热器的执行器。开启或关闭加热器时，执行器会收到新的数据。

## 自定义结构

自定义结构（也称为结构体）表示复杂或高阶数据结构。它便于对来自同一来源的数据进行逻辑绑定或分组。结构用于在原子操作中读取或写入数据，例如表示复杂数据类型或高阶形状。

在信号目录中，使用对结构数据类型的引用而不是基元数据类型来定义结构类型的信号。结构可用于所有类型的信号，包括传感器、属性、执行器和视觉系统数据类型。如果发送或接收结构类型的信号，AWS IoT FleetWise 预计所有包含的项目都具有有效值，因此所有项目都是必填项。例如，如果结构包含项目 `Vehicle.Camera.Image.height`、`Vehicle.Camera.Image.width` 和 `Vehicle.Camera.Image.data`，预计发送的信号包含所有这些项目的值。

### Note

视觉系统数据目前为预览版，可能会发生变化。

## 自定义属性

自定义属性代表复杂数据结构的成员。该属性的数据类型可以是基元结构，也可以是其他结构。

当使用结构和自定义属性表示高阶形状时，始终将预期的高阶形状定义为树结构并将其视为树结构。自定义属性用于定义所有叶节点，而结构用于定义所有非叶节点。

### Note

- 如果您使用 AWS 物联网 FleetWise 控制台创建第一个车辆模型，则无需手动创建信号目录。当您创建第一个车辆模型时，AWS IoT FleetWise 会自动为您创建信号目录。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。
- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，则可以上传 .dbc 文件以导入信号。 .dbc 是控制器局域网 (CAN 总线) 数据库支持的文件格式。创建车辆型号后，新的信号会自动添加到信号目录中。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。
- AWS 物联网 FleetWise 目前支持 AWS 账户 每个区域的信号目录。

AWS IoT FleetWise 提供了以下 API 操作，您可以使用这些操作来创建和管理信号目录。

- [CreateSignalCatalog](#)— 创建新的信号目录。
- [ImportSignalCatalog](#)— 通过上传 .json 文件导入信号以创建信号目录。信号必须按照 VSS 进行定义，并以 JSON 格式保存。
- [UpdateSignalCatalog](#)— 通过更新、删除或添加信号来更新现有信号目录。
- [DeleteSignalCatalog](#)— 删除现有的信号目录。
- [ListSignalCatalogs](#)— 检索所有信号目录摘要的分页列表。
- [ListSignalCatalogNodes](#)— 检索给定信号目录中所有信号 (节点) 摘要的分页列表。
- [GetSignalCatalog](#)— 检索有关信号目录的信息。

### 教程

- [配置 AWS 物联网 FleetWise 信号](#)
- [创建 AWS 物联网 FleetWise 信号目录](#)
- [导入 AWS 物联网 FleetWise 信号目录](#)
- [更新物 AWS 联网 FleetWise 信号目录](#)
- [删除物 AWS 联网 FleetWise 信号目录](#)
- [获取 AWS 物联网 FleetWise 信号目录信息](#)

## 配置 AWS 物联网 FleetWise 信号

本节演示了如何配置分支、属性、传感器和执行器。

## 主题

- [配置分支](#)
- [配置属性](#)
- [配置传感器或执行器](#)
- [配置复杂数据类型](#)

## 配置分支

要添加新连接，请指定以下信息：

- `fullyQualifiedName` – 分支的完全限定名称是分支路径加上分支名称。使用点 (.) 表示子分支。例如，`Vehicle.Chassis.SteeringWheel` 是 `SteeringWheel` 分支的完全限定名称。`Vehicle.Chassis.` 是指向这个分支的路径。

完全限定名称最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、冒号 (:) 和下划线 ( \_ )。

- ( 可选 ) `Description` – 分支的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) `deprecationMessage` - 正在移动或删除的节点或分支的弃用消息。

弃用消息最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) `comment` - 除描述之外的注释。注释可用于提供有关该分支的其他信息，例如分支的基本原理或对相关分支的引用。

注释最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

## 配置属性

要配置属性，请指定以下信息。

- `dataType`— 该属性的数据类型必须是以下类型之一：INT8、INT16、UINT8、UINT16、UINT32、\_ARRAY、INT64、UINT64、\_ARRAY、\_ARRAY、\_ARRAY、BOOLEAN\_ARRAY、FLOAT\_ARRAY、DOUBLEAN

INT8\_ARRAY、DOUBLEAN UINT8\_ARRAY、INT16 DOUBLEAN\_ARRAY、UINT16 DOUBLEAN INT32\_ARRAY、DOUBLEAN UINT32\_ARRAY、DOUBLEAN INT64\_ARRAY、DOUBLEAN UINT64\_ARRAY、DOUBLEAN\_ARRAY、DOUBLEAN\_ARRAY、在数据类型分支中定义。fullyQualifiedName

- fullyQualifiedName – 属性的完全限定名称是指向属性的路径加上属性的名称。使用点 (.) 表示子信号。例如，Vehicle.Chassis.SteeringWheel.Diameter 是 Diameter 属性的完全限定名称。Vehicle.Chassis.SteeringWheel. 是指向该属性的路径。

完全限定名称最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )。

- ( 可选 ) Description - 属性的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) unit - 属性的科学单位，例如 km 或摄氏度。
- ( 可选 ) min – 属性的最小值。
- ( 可选 ) max – 属性的最大值。
- ( 可选 ) defaultValue – 属性的默认值。
- ( 可选 ) assignedValue - 分配给属性的值。
- ( 可选 ) allowedValues - 该属性接受的值列表。
- ( 可选 ) deprecationMessage - 正在移动或删除的节点或分支的弃用消息。

弃用消息最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) comment - 除描述之外的注释。注释可用于提供有关属性的其他信息，例如属性的基本原理或对相关属性的引用。

注释最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

## 配置传感器或执行器

要配置传感器或执行器，请指定以下信息。

- dataType— 信号的数据类型必须是以下类型之一：INT8、 、 、 、 、 、 、 、 、 、 BOOLEAN、FLOAT、DOUBLE

UINT8、STRING INT16 UINT16 INT32、UNIX\_TIMESTAMP UINT32 INT64  
 UINT64、\_ARRAY、\_ARRAY、\_ARRAY、\_ARRAY、BOOLEAN\_ARRAY、FLOAT\_ARRAY、  
 INT8 DOUBLEAN\_ARRAY、UINT8 DOUBLEAN INT16 \_ARRAY、DOUBLEAN UINT16  
 \_ARRAY、DOUBLEAN INT32 \_ARRAY、UINT32 DOUBLEAN\_ARRA INT64 Y、DOUBLEAN  
 UINT64 \_ARRAY、DOUBLEAN\_ARRAY、DOUBLEAN\_ARRAY、DOUBLEAN\_在数据类型分支中  
 定义。fullyQualifiedName

- fullyQualifiedName – 信号的完全限定名称是指向信号的路径加上信号的名称。使用点 (.) 表示子信号。例如，Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState 是 HandsOffSteeringState 执行器的完全限定名称。Vehicle.Chassis.SteeringWheel.HandsOff. 是该执行器的路径。

完全限定名称最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )。

- ( 可选 ) Description - 信号的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) unit-信号的科单位，例如 km 或摄氏度。
- ( 可选 ) min - 信号的最小值。
- ( 可选 ) max - 信号的最大值。
- ( 可选 ) assignedValue - 分配给信号的值。
- ( 可选 ) allowedValues - 信号接受的值列表。
- ( 可选 ) deprecationMessage - 正在移动或删除的节点或分支的弃用消息。

弃用消息最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) comment - 除描述之外的注释。注释可用于提供有关传感器或执行器的更多信息，例如其原理或对相关传感器或执行器的引用。

注释最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

## 配置复杂数据类型

对视觉系统进行建模时会使用复杂数据类型。除分支外，这些数据类型还包括结构 ( 也称为结构体 ) 和属性。结构体是一种由多个值描述的信号，就像图像一样。属性代表结构的成员，例如原

始数据类型（例如 UINT8）或其他结构（例如时间戳）。例如，Vehicle.Cameras.Front 代表分支，Vehicle.Cameras.Front.Image 代表结构体，Vehicle.Cameras.Timestamp 代表属性。

以下复杂数据类型示例演示了如何将信号和数据类型导出到单个.json 文件。

### Example 复杂数据类型

```
{
  "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

您可以下载[演示脚本](#)，将 ROS 2 消息转换为与信号目录兼容的 VSS .json 文件。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。

视觉系统数据目前为预览版，可能会发生变化。

## 配置结构体

要配置自定义结构 ( 或结构体 ) ，请指定以下信息。

- `fullyQualifiedName` – 自定义结构的完全限定名称。例如，自定义结构的完全限定名称可以是 `ComplexDataTypes.VehicleDataTypes.SVMCamera`。

完全限定名称最多可包含 150 个字符。有效字符包括：`a-z`、`A-Z`、`0-9`、`:` ( 冒号 ) 和 `_` ( 下划线 ) 。

- ( 可选 ) `Description` - 信号的描述。

描述长度最多为 2048 个字符。有效字符包括：`a-z`、`A-Z`、`0-9`、`:` ( 冒号 ) 、 `_` ( 下划线 ) 和 `-` ( 连字符 ) 。

- ( 可选 ) `deprecationMessage` - 正在移动或删除的节点或分支的弃用消息。

弃用消息最多可包含 2048 个字符。有效字符包括：`a-z`、`A-Z`、`0-9`、`:` ( 冒号 ) 、 `_` ( 下划线 ) 和 `-` ( 连字符 ) 。

- ( 可选 ) `comment` - 除描述之外的注释。注释可用于提供有关传感器或执行器的更多信息，例如其原理或对相关传感器或执行器的引用。

注释最多可包含 2048 个字符。有效字符包括：`a-z`、`A-Z`、`0-9`、`:` ( 冒号 ) 、 `_` ( 下划线 ) 和 `-` ( 连字符 ) 。

## 配置属性

要配置自定义属性，请指定以下信息。

- `dataType`— 信号的数据类型必须是以下类型之一：`INT8`、`UINT8`、`BOOLEAN`、`DOUBLEAN`、`UINT16`、`INT16`、`_ARRAY`、`UINT32`、`INT32`、`UINT64`、`INT64`、`BOOLEAN_ARRAY`、`FLOAT_ARRAY`、`DOUBLEAN_ARRAY`、`INT8`、`DOUBLEAN`、`UINT8`、`_ARRAY`、`DOUBLEAN`、`INT16`、`_ARRAY`、`DOUBLEAN`、`UINT16`、`_ARRAY`、`INT32`、`DOUBLEAN_ARRAY`、`UINT32`、`DOUBLEAN_ARRAY`、`INT64`、`DOUBLEAN`、`UINT64`、`_ARRAY`、`DOUBLEAN_ARRAY`、`DOUBLEAN` 未知。
- `fullyQualifiedName` – 自定义属性的完全限定名称。例如，自定义属性的完全限定名称可以是 `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`。

完全限定名称最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )

- ( 可选 ) `Description` - 信号的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) `deprecationMessage` - 正在移动或删除的节点或分支的弃用消息。

弃用消息最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) `comment` - 除描述之外的注释。注释可用于提供有关传感器或执行器的更多信息，例如其原理或对相关传感器或执行器的引用。

注释最多可包含 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_ ( 下划线 ) 和 - ( 连字符 )。

- ( 可选 ) `dataEncoding` - 表示该属性是否为二进制数据。自定义属性的数据编码必须是下列编码之一：BINARY 或 TYPED。

- ( 可选 ) `structFullyQualifiedName` - 如果自定义属性的数据类型为 Struct 或，则为自定义属性的结构 ( 结构 ) 节点的完全限定名称。StructArray

完全限定名称最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )。

## 创建 AWS 物联网 FleetWise 信号目录

您可以使用 [CreateSignalCatalog](#) API 操作来创建信号目录。以下示例使用 AWS CLI。

要创建信号目录，请运行以下命令。

*signal-catalog-configuration* 替换为包含配置的.json 文件的名称。

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

- *signal-catalog-name* 替换为您正在创建的信号目录的名称。
- ( 可选 ) *description* 替换为描述以帮助您识别信号目录。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS 物联网 FleetWise 信号](#)。

```
{
  "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

您可以下载[演示脚本](#)，将 ROS 2 消息转换为与信号目录兼容的 VSS .json 文件。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。

视觉系统数据目前为预览版，可能会发生变化。

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 CreateSignalCatalog API 操作。

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"
      ]
    }
  ]
}
```

## 导入 AWS 物联网 FleetWise 信号目录

您可以使用 AWS IoT FleetWise 控制台或 API 导入信号目录。

主题

- [导入信号目录 \(控制台\)](#)
- [导入信号目录 \(AWS CLI\)](#)

### 导入信号目录 (控制台)

您可以使用 AWS IoT FleetWise 控制台导入信号目录。

**⚠ Important**

您最多可以有一个信号目录。如果您已经有信号目录，则无法在控制台中看到导入信号目录的选项。

## 导入信号目录

1. 打开[AWS 物联网 FleetWise控制台](#)。
2. 在导航窗格中，选择信号目录。
3. 在信号目录摘要页面上，选择导入信号目录。
4. 导入包含信号的文件。
  - 从 S3 存储桶上传文件
    - a. 选择 Import from S3 ( 从 S3 导入 )。
    - b. 选择浏览 S3。
    - c. 对于存储桶，请输入存储桶名称或对象，从列表中选择它，然后从列表中选择文件。选择选择文件按钮。

或者，对于 S3 URI，请输入 Amazon Simple Storage Service URI。有关更多信息，请参阅 Amazon S3 用户指南中的[存储桶访问方法](#)。

- 从您的计算机上传文件：
    - a. 选择从文件导入。
    - b. 上传[车辆信号规范 \(VSS\)](#) 格式的.json 文件。
5. 验证信号目录，然后选择导入文件。

## 导入信号目录 (AWS CLI)

您可以使用 [ImportSignalCatalog](#) API 操作上传有助于创建信号目录的 JSON 文件。您必须遵循[车辆信号规范 \(VSS\)](#) 才能将信号保存到 JSON 文件中。以下示例使用 AWS CLI。

要导入信号目录，请运行以下命令。

- `signal-catalog-name` 替换为您正在创建的信号目录的名称。
- ( 可选 ) 将描述替换为 `description`，以帮助您识别信号目录。

- `signal-catalog-configuration-vss` 替换为包含 VSS 中定义的信号的 JSON 字符串文件的名称。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS 物联网 FleetWise 信号](#)。

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

JSON 必须经过字符串化并通过 `vssJson` 字段传递。以下是 VSS 中定义的信号的示例。

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

以下示例显示在 JSON 字符串的 VSS 中定义的不同信号。

```
{
  "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

可以下载[演示脚本](#)，将 ROS 2 消息转换为与信号目录兼容的 VSS JSON 文件。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。  
视觉系统数据目前为预览版，可能会发生变化。

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 ImportSignalCatalog API 操作。

## 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"
      ]
    }
  ]
}
```

## 更新物 AWS 联网 FleetWise 信号目录

您可以使用 [UpdateSignalCatalog](#) API 操作来更新现有的信号目录。以下示例使用 AWS CLI。

要更新现有信号目录，请运行以下命令。

*signal-catalog-configuration* 替换为包含配置的 .json 文件的名称。

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

*signal-catalog-name* 替换为您要更新的信号目录的名称。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS 物联网 FleetWise 信号](#)。

**⚠ Important**

自定义结构是不可变的。如果您需要对现有的自定义结构（结构）重新排序或插入属性，请删除该结构并创建一个具有所需属性顺序的全新结构。

要删除自定义结构，请在 `nodesToRemove` 中添加该结构的完全限定名称。如果结构被任何信号引用，则无法删除。在请求更新信号目录之前，必须更新或删除任何引用该结构（其数据类型定义为目标结构）的信号。

```
{
  "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",
    "fullyQualified_name": "Vehicle.Chassis.SteeringWheel.Diameter",
    "max": 55
  }
}]
}

```

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateSignalCatalog API 操作。

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"
      ]
    }
  ]
}

```

## 验证信号目录更新

您可以使用 [ListSignalCatalogNodes](#) API 操作来验证信号目录是否已更新。以下示例使用 AWS CLI。

要检索给定信号目录中所有信号（节点）摘要的分页列表，请运行以下命令。

*signal-catalog-name* 替换为您正在检查的信号目录的名称。

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

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 `ListSignalCatalogNodes` API 操作。

JSON

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

## 删除物 AWS 联网 FleetWise 信号目录

您可以使用 [DeleteSignalCatalog](#) API 操作删除信号目录。以下示例使用 AWS CLI。

### Important

在删除信号目录之前，请确保它没有关联的车辆型号、解码器清单、车辆、车队或活动。有关说明，请参阅：

- [删除物 AWS 联网 FleetWise 车辆模型](#)
- [删除 AWS IoT FleetWise 解码器清单](#)
- [删除 AWS 物联网 FleetWise 车辆](#)
- [删除 AWS 物联网 FleetWise 舰队](#)
- [删除物 AWS 联网 FleetWise 活动](#)

要删除现有信号目录，请运行以下命令。`signal-catalog-name` 替换为要删除的信号目录的名称。

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

## 验证信号目录的删除

您可以使用 [ListSignalCatalogs](#) API 操作来验证信号目录是否已被删除。以下示例使用 AWS CLI。

要检索所有信号目录摘要的分页列表，请运行以下命令。

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

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 ListSignalCatalogs API 操作。

## JSON

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

## 获取 AWS 物联网 FleetWise 信号目录信息

您可以使用 [GetSignalCatalog](#) API 操作来检索信号目录信息。以下示例使用 AWS CLI。

要检索有关信号目录的信息，请运行以下命令。

*signal-catalog-name* 替换为要检索的信号目录的名称。

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

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 GetSignalCatalog API 操作。

JSON

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

#### Note

此操作具有最终一致性。换句话说，可能不会立即反映对信号目录进行的更改。

## 管理 AWS 物联网 FleetWise 车辆模型

您可以使用信号来创建有助于标准化车辆格式的车辆型号。车辆型号强制要求同一类型的多辆车辆提供一致的信息，以便您可以处理来自车队的的数据。使用相同车辆型号创建的车辆会继承相同的一组信号。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆](#)。

每个车辆型号都有一个状态字段，其中包含车辆型号的状态。状态可以是以下值之一：

- ACTIVE – 车辆型号处于活动状态。
- DRAFT – 车辆型号的配置已保存。

### Important

- 必须先拥有信号目录，然后才能使用 CreateModelManifest API 操作创建车辆模型。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 信号目录](#)。
- 如果您使用 AWS 物联网 FleetWise 控制台创建车辆模型，AWS IoT FleetWise 会自动为您激活车辆模型。
- 如果您使用 CreateModelManifest API 操作创建车辆型号，则车辆型号将保持 DRAFT 状态。
- 您无法通过处于 DRAFT 状态的车辆型号创建车辆。使用 UpdateModelManifest API 操作可将车辆型号更改为 ACTIVE 状态。
- 您无法编辑处于 ACTIVE 状态的车辆型号。

### 主题

- [创建 AWS 物联网 FleetWise 车辆模型](#)
- [更新物 AWS 联网 FleetWise 车辆模型](#)
- [删除物 AWS 联网 FleetWise 车辆模型](#)
- [获取 AWS 物联网 FleetWise 车辆型号信息](#)

## 创建 AWS 物联网 FleetWise 车辆模型

您可以使用 AWS 物联网 FleetWise 控制台或 API 来创建车辆模型。

### 主题

- [创建车辆型号 \(控制台\)](#)
- [创建车辆型号 \(AWS CLI\)](#)

### 创建车辆型号 (控制台)

在 AWS 物联网 FleetWise 控制台中，您可以通过以下方式创建车辆模型：

- [使用由提供的模板 AWS](#)
- [手动创建车辆型号](#)
- [复制车辆型号](#)

## 使用由提供的模板 AWS

AWS 物联网 FleetWise 提供了车载诊断 (OBD) II、J1979 模板，可自动为您创建信号目录、车辆模型和解码器清单。该模板还将 OBD 网络接口添加到解码器清单中。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

### 使用模板创建车辆型号

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 在车辆型号页面上，选择添加提供的模板。
4. 选择板载诊断 (OBD) II。
5. 输入 AWS 物联网 FleetWise 正在创建的 OBD 网络接口的名称。
6. 选择添加。

### 手动创建车辆型号

您可以从信号目录中添加信号，也可以通过上传一个或多个.dbc 文件来导入信号。.dbc 文件是控制器局域网 (CAN 总线) 数据库支持的文件格式。

#### Important

您无法使用 AWS 物联网 FleetWise 控制台创建带有视觉系统数据信号的车辆模型。而是使用 AWS CLI 来创建车辆模型。  
视觉系统数据目前为预览版，可能会发生变化。

### 手动创建车辆型号

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 在车辆型号页面上，选择创建车辆型号，然后执行以下操作。

#### 主题

- [步骤 1：配置车辆型号](#)
- [步骤 2：添加信号](#)

- [第 3 步：导入信号](#)
- [\( 可选 \) 步骤 4：添加属性](#)
- [第 5 步：审核并创建](#)

## 步骤 1：配置车辆型号

在一般信息部分，执行以下操作：

1. 输入车辆型号的名称。
2. ( 可选 ) 输入描述。
3. 选择下一步。

## 步骤 2：添加信号

### Note

- 如果这是您首次使用 AWS 物联网 FleetWise，则在您拥有信号目录之前，此步骤不可用。创建第一个车辆模型后，AWS 物联网 FleetWise 会自动创建信号目录，并将信号添加到第一个车辆模型中。
- 如果您熟悉 AWS 物联网 FleetWise，则可以通过从信号目录中选择信号或上传.dbc 文件来导入信号，从而向车辆模型添加信号。
- 您必须至少有一个信号才能创建车辆型号。

## 添加信号

1. 从要添加到车辆型号中的信号目录中选择一个或多个信号。您可以在右侧窗格中查看所选信号。

### Note

只有选定的信号才会添加到车辆型号中。

2. 选择下一步。

## 第 3 步：导入信号

### Note

- 如果这是您首次使用 AWS 物联网 FleetWise，则必须上传至少一个 .dbc 文件才能导入信号。
- 如果您熟悉 AWS 物联网 FleetWise，则可以通过从信号目录中选择信号或上传 .dbc 文件来导入信号，从而向车辆模型添加信号。
- 您必须至少有一个信号才能创建车辆型号。

### 导入信号

1. 选择选择文件。
2. 在此对话框中，选择包含信号的 .dbc 文件。可以上传多个 .dbc 文件。
3. AWS 物联网 FleetWise 解析您的 .dbc 文件以检索信号。

在信号部分，为每个信号指定以下元数据。

- Name - 信号的名称。

名称必须唯一。信号名称加路径最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )。

- 数据类型 — 信号的数据类型必须是以下类型之一：

INT8、 、 、 、 、 、 、 、 、 、 BOOLEAN、FLOAT、DOUBLE、STRING  
 UINT8、UNIX\_TIMESTAMP INT16 UINT16 INT32、\_ARRAY UINT32 INT64、\_ARRAY  
 UINT64、\_ARRAY、\_ARRAY、\_ARRAY、BOOLEAN\_ARRAY、FLOAT、DOUBLEAN\_ARRAY、  
 INT8 DOUBLEAN UINT8 \_ARRAY、STRING\_ARRAY、INT16 \_ARRAY、UINT16  
 \_ARRAY、\_ARRAY、INT32 \_ARRAY、\_ARRAY、UINT32 \_ARRAY、\_ARRAY、BOOLEAN  
 INT64 UINT64 \_ARRAY、FLOAT\_ARRAY、

- Signal type - 信号的类型，可以是传感器或执行器。
- ( 可选 ) Unit - 信号的科学单位，例如 km 或摄氏度。
- ( 可选 ) Path - 信号的路径。类似于 JSONPath，使用点 (.) 来指代子信号。例如 **Vehicle.Engine.Light**。

信号名称加路径最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )。

- ( 可选 ) Min – 信号的最小值。
- ( 可选 ) Max – 信号的最大值。
- ( 可选 ) Description – 信号的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

#### 4. 选择下一步。

( 可选 ) 步骤 4：添加属性

您最多可以添加 100 个属性，包括信号目录中的现有属性。

#### 添加属性

##### 1. 在添加属性中，为每个属性指定以下元数据。

- Name – 属性的名称

名称必须唯一。信号名称和路径最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )

- 数据类型 — 属性的数据类型必须是以下类型之一：

INT8、INT16、UNIX\_TIMESTAMP、UINT16、INT32、\_ARRAY、UINT32、INT64、\_ARRAY、UINT64、\_ARRAY、\_ARRAY、\_ARRAY、BOOLEAN\_ARRAY、FLOAT\_ARRAY、DOUBLEAN、INT8\_ARRAY、DOUBLEAN、UINT8\_ARRAY、STRIN、INT16、G\_ARRAY、\_ARRAY、UINT16\_ARRAY、INT32\_ARRAY、\_ARRAY、UINT32\_ARRAY、\_ARRAY、INT64、BOOLEAN\_ARRAY、FLOAT、UINT64\_ARRAY

- ( 可选 ) Unit - 属性的科学单位，例如 km 或摄氏度。
- ( 可选 ) Path - 信号的路径。类似于 JSONPath，使用点 (.) 来指代子信号。例如 **Vehicle.Engine.Light**。

信号名称加路径最多可包含 150 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 ) 和 \_ ( 下划线 )

- ( 可选 ) Min – 属性的最小值。
- ( 可选 ) Max – 属性的最大值。
- ( 可选 ) Description - 属性的描述。

描述长度最多为 2048 个字符。有效字符包括：a-z、A-Z、0-9、:( 冒号 )、\_( 下划线 ) 和 - ( 连字符 )。

2. 选择下一步。

## 第 5 步：审核并创建

验证车辆型号的配置，然后选择创建。

### 复制车辆型号

AWS 物联网 FleetWise 可以复制现有车辆模型的配置来创建新模型。所选车辆型号中指定的信号将复制到新的车辆型号中。

### 复制车辆型号

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 从车辆型号列表选择一个模型，然后选择复制模型。

要配置车辆型号，请按照[手动创建车辆型号](#)教程操作。

AWS 物联网可能需要几分钟 FleetWise 才能处理您创建车辆模型的请求。成功创建车辆型号后，在车辆型号页面上，状态列将显示为活动。当车辆型号变为活动状态时，您无法对其进行编辑。

## 创建车辆型号 (AWS CLI)

您可以使用 [CreateModelManifest](#) API 操作来创建车辆模型（模型清单）。下面的示例使用了 AWS CLI。

### Important

必须先拥有信号目录，然后才能使用 `CreateModelManifest` API 操作创建车辆模型。有关如何创建信号目录的更多信息，请参阅 [创建 AWS 物联网 FleetWise 信号目录](#)。

要创建车辆型号，请运行以下命令。

`vehicle-model-configuration` 替换为包含配置的 .json 文件的名称。

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

- *vehicle-model-name* 替换为您正在创建的车辆模型的名称。
- *signal-catalog-ARN* 替换为信号目录的亚马逊资源名称 (ARN)。
- ( 可选 ) *description* 替换为描述以帮助您识别车辆型号。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS 物联网 FleetWise 信号](#)。

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

如果您[启用了使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 CreateModelManifest API 操作。

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"  
      ]  
    }  
  ]  
}
```

## 更新物 AWS 联网 FleetWise 车辆模型

您可以使用 [UpdateModelManifestAPI](#) 操作来更新现有的车辆模型（模型清单）。下面的示例使用了 AWS CLI。

要更新现有车辆型号，请运行以下命令。

*update-vehicle-model-configuration* 替换为包含配置的 .json 文件的名称。

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

- *vehicle-model-name* 替换为您要更新的车辆型号的名称。
- （可选）要激活车辆模型，请 *vehicle-model-status* 替换为 ACTIVE。

### Important

激活车辆型号后，将无法更改此车辆型号。

- （可选）*description* 替换为更新的描述以帮助您识别车辆型号。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateModelManifest API 操作。

### 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"
        ]
    }
]
}

```

## 验证车辆型号更新

您可以使用 [ListModelManifestNodes](#) API 操作来验证车辆模型是否已更新。以下示例使用 AWS CLI。

要检索给定车辆型号中所有信号（节点）摘要的分页列表，请运行以下命令。

*vehicle-model-name* 替换为您正在检查的车辆型号的名称。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 `ListModelManifestNodes` API 操作。

## JSON

```

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

```

## 删除物 AWS 联网 FleetWise 车辆模型

您可以使用 AWS IoT FleetWise 控制台或 API 删除车辆模型。

### Important

必须先删除与车辆型号相关的车辆和解码器清单。有关更多信息，请参阅[删除 AWS 物联网 FleetWise 车辆](#)和[删除 AWS IoT FleetWise 解码器清单](#)：

### 删除车辆型号 ( 控制台 )

要删除车辆模型，请使用 AWS 物联网 FleetWise 控制台。

#### 删除车辆型号

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 在车辆型号页面上，选择目标车辆型号。
4. 选择删除。
5. 在删除 **vehicle-model-name?** 中，输入要删除的车辆型号的名称，然后选择确认。

### 删除车辆型号 (AWS CLI)

您可以使用 [DeleteModelManifest](#) API 操作删除现有的车辆模型 ( 车型清单 )。下面的示例使用了 AWS CLI。

要删除车辆型号，请运行以下命令。

*model-manifest-name* 替换为您要删除的车辆型号的名称。

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

#### 验证删除车辆型号

您可以使用 [ListModelManifests](#) API 操作来验证车辆模型是否已删除。以下示例使用 AWS CLI。

要检索所有车辆型号摘要的分页列表，请运行以下命令。

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

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 ListModelManifests API 操作。

JSON

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

## 获取 AWS 物联网 FleetWise 车辆型号信息

您可以使用 [GetModelManifest](#) API 操作来检索有关车辆模型的信息。以下示例使用 AWS CLI。

要检索有关车辆型号的信息，请运行以下命令。

*vehicle-model* 替换为要检索的车辆型号的名称。

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

### Note

此操作具有[最终一致性](#)。换句话说，可能不会立即反映对车辆型号进行的更改。

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 GetModelManifest API 操作。

## JSON

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

## 管理 AWS IoT FleetWise 解码器清单

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

解码器清单包含解码信息，物 AWS 联网 FleetWise 使用这些信息将车辆数据（二进制数据）转换为人类可读的值，并为数据分析做好准备。网络接口和信号解码器是您用来配置解码器清单的核心组件。

### 网络接口

包含有关车载网络使用的协议的信息。AWS 物联网 FleetWise 支持以下协议。

#### 控制器局域网 (CAN 总线)

一种协议，用于定义电子控制单元之间如何传输数据 (ECUs)。ECUs 可以是发动机控制单元、安全气囊或音频系统。

#### 板载诊断 (OBD) II

一种进一步开发的协议，用于定义相互之间 ECUs 如何传输自我诊断数据。它提供了许多标准的诊断故障代码 (DTCs)，可帮助识别您的车辆出了什么问题。

## 车辆中间件

车辆中间件定义为一种网络接口。车辆中间件的示例包括机器人操作系统 (ROS 2) 和可扩展的面向服务的 IP 中间件 (SOME/IP)。

### Note

AWS 物联网 FleetWise 支持用于视觉系统数据的 ROS 2 中间件。

## 自定义接口

您也可以使用自己的接口来解码边缘的信号。这可以节省您的时间，因为您无需在云中创建解码规则。

## 信号解码器

提供特定信号的详细解码信息。车辆模型中指定的每个信号都必须与信号解码器配对。如果解码器清单包含 CAN 网络接口，则它必须包含 CAN 解码器信号。如果解码器清单包含 OBD 网络接口，则必须包含 OBD 信号解码器。

如果解码器清单还包含车辆中间件接口，则它必须包含消息信号解码器。或者，如果解码器清单包含自定义解码接口，则它还必须包含自定义解码信号。

每个解码器清单都必须与车辆型号相关联。AWS 物联网 FleetWise 使用关联的解码器清单来解码基于车辆模型创建的车辆的数据。

每个解码器清单都有一个状态字段，其中包含解码器清单的状态。状态可以是以下值之一：

- ACTIVE – 解码器清单处于活动状态。
- DRAFT – 未保存解码器清单的配置。
- VALIDATING – 正在验证解码器清单资格。这仅适用于包含至少一个视觉系统数据信号的解码器清单。
- INVALID – 解码器清单验证失败，尚无法激活。这仅适用于包含至少一个视觉系统数据信号的解码器清单。您可以使用 ListDecoderManifests 和 GetDecoderManifest APIs 来检查验证失败的原因。

### ⚠ Important

- 如果您使用 AWS IoT FleetWise 控制台创建解码器清单，AWS IoT FleetWise 会自动为您激活解码器清单。
- 如果您使用 CreateDecoderManifest API 操作创建解码器清单，则解码器清单将保持 DRAFT 状态。
- 您无法通过与 DRAFT 解码器清单关联的车辆型号创建车辆。使用 UpdateDecoderManifest API 操作可将解码器清单更改为 ACTIVE 状态。
- 您无法编辑处于 ACTIVE 状态的解码器清单。

### 主题

- [配置 AWS IoT FleetWise 网络接口和解码器信号](#)
- [创建 AWS IoT FleetWise 解码器清单](#)
- [更新 AWS IoT FleetWise 解码器清单](#)
- [删除 AWS IoT FleetWise 解码器清单](#)
- [获取 AWS 物联网 FleetWise 解码器清单信息](#)

## 配置 AWS IoT FleetWise 网络接口和解码器信号

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

每个解码器清单都至少有一个网络接口和信号解码器，并与相关车辆模型中指定的信号配对。

如果解码器清单包含 CAN 网络接口，则必须包含 CAN 信号解码器。如果解码器清单包含 OBD 网络接口，则必须包含 OBD 信号解码器。

### 主题

- [配置网络接口](#)
- [配置信号解码器](#)

## 配置网络接口

要配置 CAN 网络接口，请指定以下信息。

- name – CAN 接口的名称。

接口名称必须是唯一的，可以包含 1-100 个字符。

- ( 可选 ) protocolName - 协议的名称。

有效值：CAN-FD 和 CAN。

- ( 可选 ) protocolVersion— AWS 物联网 FleetWise 目前支持 CAN-FD 和 CAN 2.0b。

有效值：1.0 和 2.0b。

要配置 OBD 网络接口，请指定以下信息。

- name – OBD 接口的名称。

接口名称必须是唯一的，可以包含 1-100 个字符。

- requestMessageId – 请求数据的消息 ID。

- ( 可选 ) dtcRequestIntervalSeconds— 在几秒钟内向车辆请求诊断故障代码 (DTCs) 的频率。例如，如果指定值为 120，则 Edge Agent 软件每 2 分钟收集 DTCs 一次存储的内容。

- ( 可选 ) hasTransmissionEcu – 车辆是否有变速器控制模块 (TCM)。

有效值：true 和 false。

- ( 可选 ) obdStandard— AWS 物联网 FleetWise 支持的 OBD 标准。AWS 物联网 FleetWise 目前支持全球统一机载诊断 (WWH-OBD) -4 标准。ISO15765

- ( 可选 ) pidRequestIntervalSeconds— 向车辆申请 OBD II PIDs 的频率。例如，如果指定值为 120，则 Edge Agent 软件每 2 分钟收集 PIDs 一次 OBD II。

- ( 可选 ) useExtendedIds-是否在消息 IDs 中使用扩展。

有效值：true 和 false。

要配置中间件网络接口，请指定以下信息。

- name – 车辆中间件接口名称。

接口名称必须是唯一的，可以包含 1-100 个字符。

- `protocolName` – 协议名称。

有效值：R0S\_2

要配置自定义解码接口，请指定以下信息。

- `name`— 用于解码边缘信号的解码器的名称。

解码器接口名称可以包含 1-100 个字符。

## 配置信号解码器

要配置 CAN 信号解码器，请指定以下信息。

- `factor` – 用于解码消息的乘数。
- `isBigEndian` – 消息的字节顺序是否为大端序。如果是大端序，则序列中最重要值首先存储在最低的存储地址上。
- `isSigned` – 消息是否已签名。如果已签名，则消息可以同时表示正数和负数。
- `length`— 消息的长度（以位为单位）。
- `messageId` – 消息的 ID。
- `offset` – 用于计算信号值的偏移量。再加上因子，计算结果为  $value = raw\_value * factor + offset$ 。
- `startBit` – 表示消息第一位的位置。
- ( 可选 ) `name` – 信号的名称。
- ( 可选 ) `signalValueType`-信号的值类型。整数是默认值类型。

要配置 OBD 信号解码器，请指定以下信息。

- `byteLength` – 消息的总字节长度。
- `offset` – 用于计算信号值的偏移量。再加上缩放，计算结果为  $value = raw\_value * scaling + offset$ 。
- `pid` – 用于从车辆请求此信号的相关消息的诊断代码。
- `pidResponseLength` – 所请求消息的长度。
- `scaling` – 用于解码消息的乘数。
- `serviceMode` – 消息中的操作模式（诊断服务）。

- `startByte` – 指示消息的起始。
- ( 可选 ) `bitMaskLength` - 消息中被屏蔽的位数。
- ( 可选 ) `bitRightShift` - 向右移动的位置数。
- ( 可选 ) `isSigned`-消息是否已签名。如果已签名，则消息可以同时表示正数和负数。默认情况下，消息未签名 (`false`)。
- ( 可选 ) `signalValueType`-信号的值类型。整数是默认值类型。

要配置消息信号解码器，请指定以下信息。

- `topicName` – 消息信号的主题名称。它对应于 ROS 2 中的主题。有关结构化消息对象的更多信息，请参阅[StructuredMessage](#)。
- `structuredMessage` – 消息信号的结构化消息。它可以用 `a primitiveMessageDefinition`、“`structuredMessageList`定义”或 `structuredMessageDefinition` 递归方式定义。

要配置自定义解码信号，请指定以下信息。

- ( 可选 ) `id`— 您使用解码器接口自己解码的信号的 ID。信号 ID 可以包含 1-150 个字符。如果未指定，`id`则`fullyQualifiedNames`默认为信号的。

## 创建 | AWS IoT FleetWise 解码器清单

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 AWS IoT FleetWise 控制台或 API 为您的车辆模型创建解码器清单。

### 主题

- [创建解码器清单 \(控制台\)](#)
- [创建解码器清单 \(AWS CLI\)](#)

## 创建解码器清单（控制台）

您可以使用 AWS IoT FleetWise 控制台创建与您的车辆型号关联的解码器清单。

### Important

您无法使用物 AWS 联 FleetWise网控制台在解码器清单中配置视觉系统数据信号。可改用 AWS CLI。视觉系统数据目前为预览版，可能会发生变化。

### 创建解码器清单

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 选择目标车辆型号。
4. 在“车辆型号摘要”页面上，选择创建解码器清单，然后执行以下操作。

### 主题

- [步骤 1：配置解码器清单](#)
- [第 2 步：映射 CAN 接口](#)
- [第 3 步：审核并创建](#)

### 步骤 1：配置解码器清单

在一般信息部分，执行以下操作：

1. 为解码器清单输入唯一名称。
2. （可选）输入描述。
3. 选择下一步。

### 添加网络接口

每个解码器清单必须具有至少一个网络接口。您可以向解码器清单中添加多个网络接口。

## 添加网络接口

1. 上传网络接口文件。您可以上传适用于 CAN 协议的 .dbc 文件，或者为 ROS 2 或自定义接口上传 .json 文件。
2. 输入您的网络接口的名称。如果您上传了自定义接口，则名称已提供。

## 映射缺失的信号

如果车辆模型中有信号在上传的网络接口中缺少配对信号解码器，则可以创建一个默认的自定义解码器来映射缺失的信号。这是可选的，因为您可以在下一步中手动映射信号。

## 创建默认的自定义解码器

1. 选择“为缺失的信号创建默认的自定义解码器”。
2. 选择下一步。

## 第 2 步：映射 CAN 接口

您可以使用 CAN 信号解码器映射 CAN 信号。如果您选中“为缺失的信号创建默认自定义解码器”复选框，则任何缺少解码器信号的信号都会自动映射到默认的自定义信号解码器。

## 要映射 CAN 信号

1. 在 CAN 信号映射中，选择信号解码器。
2. 选择下一步。

### Note

如果您添加了 ROS 2 或自定义接口，则可以在创建解码器清单之前验证映射。

## 第 3 步：审核并创建

验证解码器清单的配置，然后选择创建。

## 创建解码器清单 (AWS CLI)

您可以使用 [CreateDecoderManifest](#) API 操作来创建解码器清单。下面的示例使用了 AWS CLI。

**⚠ Important**

在创建解码器清单之前，您必须拥有车辆型号。每个解码器清单都必须与车辆型号相关联。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 车辆模型](#)。

要创建解码器清单，请运行以下命令。

*decoder-manifest-configuration* 替换为包含配置的 .json 文件的名称。

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

- *decoder-manifest-name* 替换为您正在创建的解码器清单的名称。
- *vehicle-model-ARN* 替换为车辆型号的 Amazon 资源名称 (ARN)。
- ( 可选 ) *description* 替换为描述以帮助您识别解码器清单。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅 [配置 AWS IoT FleetWise 网络接口和解码器信号](#)。

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [
    {
      "canInterface": {
        "name": "myNetworkInterface",
        "protocolName": "CAN",
        "protocolVersion": "2.0b"
      },
      "interfaceId": "Qq1acaenBy0B3sSM39SYm",
      "type": "CAN_INTERFACE"
    }
  ],
  "signalDecoders": [
    {
      "canSignal": {
        "name": "Engine_Idle_Time",
        "factor": 1,

```

```

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

```

- *decoder-manifest-name* 替换为您正在创建的解码器清单的名称。
- *vehicle-model-ARN* 替换为车辆型号的 Amazon 资源名称 (ARN)。
- ( 可选 ) *description* 替换为描述以帮助您识别解码器清单。

结构 ( 结构体 ) 中属性节点的顺序必须与信号目录和车辆模型 ( 模型清单 ) 中定义的顺序保持一致。有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS IoT FleetWise 网络接口和解码器信号](#)。

```

{
  "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
  },
  "fullyQualified_name": "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
  },
  "fullyQualified_name": "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* 替换为您正在创建的解码器清单的名称。

- *vehicle-model-ARN* 替换为车辆型号的 Amazon 资源名称 (ARN)。
- ( 可选 ) *description* 替换为描述以帮助您识别解码器清单。

有关如何配置分支、属性、传感器和执行器的更多信息，请参阅[配置 AWS IoT FleetWise 网络接口和解码器信号](#)。

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [
    {
      "interfaceId": "myCustomInterfaceId",
      "type": "CUSTOM_DECODING_INTERFACE",
      "customDecodingInterface": {
        "name": "myCustomInterface"
      }
    }
  ],
  "signalDecoders": [
    {
      "customDecodingSignal": {
        "fullyQualifiedNames": "Vehicle.actuator1",
        "interfaceId": "myCustomInterfaceId",
        "type": "CUSTOM_DECODING_SIGNAL",
        "customDecodingSignal": {
          "id": "Vehicle.actuator1"
        }
      }
    },
    {
      "customDecodingSignal": {
        "fullyQualifiedNames": "Vehicle.actuator2",
        "interfaceId": "myCustomInterfaceId",
        "type": "CUSTOM_DECODING_SIGNAL",
        "customDecodingSignal": {
          "id": "Vehicle.actuator2"
        }
      }
    }
  ]
}
```

**Note**

您可以下载[演示脚本](#)来创建包含视觉系统信号的解码器清单。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。

视觉系统数据目前为预览版，可能会发生变化。

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 CreateDecoderManifest API 操作。

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"
      ]
    }
  ]
}
```

## 更新 AWS IoT FleetWise 解码器清单

**Important**

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅[AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [UpdateDecoderManifest](#) API 操作来更新解码器清单。您可以添加、移除和更新网络接口和信号解码器。您还可以更改解码器清单的状态。下面的示例使用了 AWS CLI。

要更新解码器清单，请运行以下命令。

*decoder-manifest-name* 替换为您要更新的解码器清单的名称。

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

如果信号没有指定的解码规则，则可以创建默认的分码规则。信号被添加到自定义解码接口中，CustomDecodingSignal\$id 设置为信号的完全限定名称。要使用默认解码规则更新解码器清单，请运行以下命令。

*decoder-manifest-name* 替换为您要更新的解码器清单的名称。

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

#### Important

激活解码器清单后，您无法对其进行编辑。

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateDecoderManifest API 操作。

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"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

## 验证解码器清单更新

您可以使用 [ListDecoderManifestSignals](#) API 操作来验证解码器清单中的解码器信号是否已更新。以下示例使用 AWS CLI。

要检索给定解码器清单中所有解码器信号（节点）摘要的分页列表，请运行以下命令。

*decoder-manifest-name* 替换为您正在检查的解码器清单的名称。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 `ListDecoderManifestSignals` API 操作。

## JSON

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

您可以使用 [ListDecoderManifestNetworkInterfaces](#) API 操作来验证解码器清单中的网络接口是否已更新。下面的示例使用了 AWS CLI。

要检索给定解码器清单中所有网络接口摘要的分页列表，请运行以下命令。

`decoder-manifest-name` 替换为您正在检查的解码器清单的名称。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 `ListDecoderManifestNetworkInterfaces` API 操作。

JSON

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

## 删除 AWS IoT FleetWise 解码器清单

您可以使用 AWS IoT FleetWise 控制台或 API 删除解码器清单。

### Important

必须先删除与解码器清单关联的车辆。有关更多信息，请参阅 [删除 AWS 物联网 FleetWise 车辆](#)。

主题

- [删除解码器清单 \(控制台\)](#)
- [删除解码器清单 \(AWS CLI\)](#)

## 删除解码器清单 ( 控制台 )

您可以使用 AWS IoT FleetWise 控制台删除解码器清单。

### 删除解码器清单

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆型号。
3. 选择目标车辆型号。
4. 在“车辆型号摘要”页面上，选择解码器清单选项卡。
5. 选择目标解码器清单，然后选择删除。
6. 在删除 **decoder-manifest-name?** 中，输入要删除的解码器清单的名称，然后选择确认。

## 删除解码器清单 (AWS CLI)

您可以使用 [DeleteDecoderManifest](#) API 操作删除解码器清单。以下示例使用 AWS CLI。

### Important

在删除解码器清单之前，请先删除关联的车辆。有关更多信息，请参阅 [删除 AWS 物联网 FleetWise 车辆](#)。

要删除解码器清单，请运行以下命令。

*decoder-manifest-name* 替换为您要删除的解码器清单的名称。

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

### 验证已删除解码器清单

您可以使用 [ListDecoderManifests](#) API 操作来验证解码器清单是否已被删除。以下示例使用 AWS CLI。

要检索所有解码器清单摘要的分页列表，请运行以下命令。

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

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 ListDecoderManifests API 操作。

JSON

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

## 获取 AWS 物联网 FleetWise 解码器清单信息

您可以使用 [GetDecoderManifest](#) API 操作来验证解码器清单中的网络接口和信号解码器是否已更新。以下示例使用 AWS CLI。

要检索有关解码器清单的信息，请运行以下命令。

*decoder-manifest* 替换为要检索的解码器清单的名称。

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

### Note

此操作具有**最终一致性**。换句话说，可能不会立即反映对解码器清单进行的更改。

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 GetDecoderManifest API 操作。

## JSON

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

# 管理 AWS 物联网 FleetWise 车辆

车辆是车辆型号的实例。车辆必须根据车辆型号进行创建，并与解码器清单相关联。车辆将一个或多个数据流上传到云端。例如，车辆可以将里程、发动机温度和加热器状态数据发送到云端。每个车辆事件都包含以下信息：

## vehicleName

标识车辆的 ID。

请勿在车辆名称中添加个人信息 (PII) 或其他机密或敏感信息。其他 AWS 服务 (包括亚马逊) 均可访问车辆名称 CloudWatch。车辆名称不适合用于私有或敏感数据。

## modelManifestARN

车辆型号 (模型清单) 的 Amazon 资源名称 (ARN)。每个车辆都是根据车辆型号创建的。使用相同车辆型号创建的车辆由从该车辆型号继承的一组相同信号组成。这些信号在信号目录中定义和标准化。

## decoderManifestArn

解码器清单的 ARN。解码器清单提供解码信息，物 AWS 联网 FleetWise 可以使用这些信息将原始信号数据 (二进制数据) 转换为人类可读的值。解码器清单必须与车辆模型相关联。AWS 物联网 FleetWise 使用相同的解码器清单来解码基于相同车辆模型创建的车辆的原始数据。

## attributes

属性是包含静态信息的键值对。车辆可以包含继承自车辆型号的属性。您可以添加其他属性来区分单个车辆和使用相同车辆型号创建的其他车辆。例如，如果您有一辆黑色汽车，则可以为属性指定以下值：{"color": "black"}。

### Important

必须先关联的车辆型号中定义属性，然后才能将其添加到单个车辆中。

有关车辆型号、解码器清单和属性的更多信息，请参阅[对 AWS 物联网 FleetWise 车辆进行建模](#)。

AWS 物联网 FleetWise 提供了以下 API 操作，您可以使用这些操作来创建和管理车辆。

- [CreateVehicle](#)— 创建一辆新车。
- [BatchCreateVehicle](#)— 创建一辆或多辆新车。

- [UpdateVehicle](#)— 更新现有车辆。
- [BatchUpdateVehicle](#)— 更新一辆或多辆现有车辆。
- [DeleteVehicle](#)— 删除现有车辆。
- [ListVehicles](#)— 检索所有车辆摘要的分页列表。
- [GetVehicle](#)— 检索有关车辆的信息。

## 教程

- [配置 AWS 物联网 FleetWise 车辆](#)
- [AWS 物联网中的保留话题 FleetWise](#)
- [创建物 AWS 联网 FleetWise 汽车](#)
- [创建多 FleetWise 辆 AWS 物联网车辆](#)
- [更新 AWS 物联网 FleetWise 车辆](#)
- [更新多 FleetWise 辆 AWS 物联网车辆](#)
- [删除 AWS 物联网 FleetWise 车辆](#)
- [获取 AWS 物联网 FleetWise 车辆信息](#)

## 配置 AWS 物联网 FleetWise 车辆

车辆中运行的 AWS 物联网边缘代理 FleetWise 软件可收集数据并将其传输到云端。AWS 物联网与 FleetWise 集成 AWS IoT Core，支持边缘代理软件和云之间通过 MQTT 进行安全通信。每辆车对应一个 AWS IoT 事物。您可以使用现有 AWS IoT 事物来创建车辆，也可以将 AWS 物联网设置 FleetWise 为自动为您的车辆创建 AWS IoT 事物。有关更多信息，请参阅 [创建物 AWS 联网 FleetWise 汽车](#)。

AWS IoT Core 支持[身份验证](#)和[授权](#)，有助于安全地控制对 AWS 物联网 FleetWise 资源的访问。车辆可以使用 X.509 证书进行身份验证（登录）以使用物 AWS 联网，FleetWise 并使用 AWS IoT Core 策略获得授权（拥有权限）以执行指定操作。

## 对车辆进行身份验证

您可以创建 AWS IoT Core 策略来验证您的车辆。

对您的车辆进行身份验证

- 要创建 AWS IoT Core 策略，请运行以下命令。

- `policy-name` 替换为要创建的策略的名称。
- `file-name` 替换为包含 AWS IoT Core 策略的 JSON 文件的名称。

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

在使用策略示例之前，执行以下操作：

- `us-east-1` 替换为您在其中创建 AWS IoT FleetWise 资源的 AWS 区域。
- `111122223333` 替换为您的 AWS 账户 ID。

此示例包括 AWS IoT 保留的主题 FleetWise。您必须将这些主题添加到策略中。有关更多信息，请参阅 [AWS 物联网中的保留话题 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"
    ]
  }
]
}
```

## 授权车辆

您可以创建 X.509 证书来为您的车辆授权。

授权您的车辆

### Important

建议您为每辆车创建新证书。

1. 要创建 RSA 密钥对并颁发 X.509 证书，请运行以下命令。

- *cert* 替换为保存 CertificatePem 命令输出内容的文件的名称。
- *public-key* 替换为保存 KeyPair 命令输出内容的文件名。PublicKey。
- *private-key* 替换为保存 KeyPair 命令输出内容的文件名。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. 从输出中复制证书的 Amazon 资源名称 (ARN)。

3. 要将策略附加到证书，请运行以下命令。

- *policy-name* 替换为您创建的 AWS IoT Core 策略的名称。
- *certificate-arn* 替换为您复制的证书的 ARN。

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

4. 要将证书附加到物品，请运行以下命令。

- *thing-name* 用你的 AWS IoT 东西的名字或你的车辆的ID代替。
- *certificate-arn* 替换为您复制的证书的 ARN。

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

## AWS 物联网中的保留话题 FleetWise

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

AWS IoT FleetWise 保留使用以下主题。如果保留的主题允许，您可以订阅或发布该主题。但是，您不能创建以美元符号 (\$) 开头的新主题。如果您对保留的主题使用不受支持的发布或订阅操作，则可能会导致连接终止。

Topic	允许的客户端操作	说明
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> / checkins	发布	边缘代理软件向该主题发布状态信息。  车辆状态信息以协议缓冲区 (Protobuf) 格式交换。有关更多信息，请参阅适用于 <a href="#">AWS IoT 的 Edge Agent FleetWise 软件开发人员指南</a> 。
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> / signals	发布	边缘代理软件会向该主题发布信号。  信号信息以协议缓冲区 (Protobuf) 格式交换。有关更多信息，请参阅适用于 <a href="#">AWS IoT 的 Edge Agent FleetWise 软件开发人员指南</a> 。
\$aws/iotfleetwise/	Subscribe	AWS IoT 为此主题 FleetWise 发布了数

Topic	允许的客户端操作	说明
<code>vehicles/ <i>vehicleName</i> /collection_schemes</code>		据收集方案。车辆会使用这些数据收集方案。
<code>\$aws/iotfleetwise/ vehicles/ <i>vehicleName</i> /decoder_manifests</code>	Subscribe	AWS 物联网 FleetWise 发布了该主题的解码器清单。车辆会使用这些解码器清单。
<code>\$aws/iotfleetwise/ vehicles/ <i>vehicleName</i> /command/request</code>	Subscribe	AWS IoT 向该主题 FleetWise 发布执行命令的请求。然后，车辆会消耗这些命令请求。
<code>\$aws/iotfleetwise/ vehicles/ <i>vehicleName</i> /command/response</code>	发布	Edge Agent 软件会发布车辆对此主题的命令响应。  命令响应以协议缓冲区 (Protobuf) 格式交换。有关更多信息，请参阅适用于 <a href="#">AWS IoT 的 Edge Agent FleetWise 软件开发人员指南</a> 。
<code>\$aws/iotfleetwise/ vehicles/ <i>vehicleName</i> /command/notification</code>	Subscribe	AWS IoT FleetWise 会发布此主题的命令状态更新。通知以 JSON 格式发送。

Topic	允许的客户端操作	说明
\$aws/iotfleetwise/vehicles/ <i>\$vehicle_name</i> /last_known_states/config	Subscribe	AWS IoT 向该主题 FleetWise 发布状态模板配置。车辆使用这些状态模板配置。
\$aws/iotfleetwise/vehicles/ <i>\$vehicle_name</i> /last_known_states/data	发布	Edge Agent 软件将从信号中收集的数据发布到本主题。
\$aws/iotfleetwise/vehicles/ <i>\$vehicle_name</i> /last_known_state/ <i>\$state_template_name</i> /data	Subscribe	<p>AWS IoT FleetWise 会发布从本主题中指定的 <i>\$state_template_name</i> 配置的信号中收集的数据。更新可以是部分的。例如，如果状态模板关联包含多个采用 on-change 更新策略的信号，则给定消息中仅包含已更改的信号。</p> <p>信号信息以协议缓冲区 (Protobuf) 格式交换。有关更多信息，请参阅适用于 <a href="#">AWS IoT 的 Edge Agent FleetWise 软件开发人员指南</a>。</p>

# 创建物 AWS 联网 FleetWise 汽车

## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 AWS 物联网 FleetWise 控制台或 API 来创建车辆。

## ⚠ Important

在开始之前，请检查以下事项：

- 您必须拥有车辆型号，并且该车辆型号的状态必须为 ACTIVE。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆模型](#)。
- 您的车辆型号必须与解码器清单相关联，并且解码器清单的状态必须为 ACTIVE。有关更多信息，请参阅 [管理 AWS IoT FleetWise 解码器清单](#)。

## 主题

- [创建车辆 \(控制台\)](#)
- [创建车辆 \(AWS CLI\)](#)

## 创建车辆 (控制台)

您可以使用 AWS IoT FleetWise 控制台创建车辆。

### 创建车辆

1. 打开 [AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆。
3. 在车辆摘要页面上，选择创建车辆，然后执行以下步骤。

## 主题

- [步骤 1：定义车辆属性](#)

- [步骤 2：配置车辆证书](#)
- [步骤 3：将策略附加到证书](#)
- [步骤 4：审核并创建](#)

## 步骤 1：定义车辆属性

在此步骤中，您将为车辆命名并将其与模型清单和解码器清单相关联。

1. 为车辆输入唯一名称。

### Important

车辆对应于一 AWS IoT 件事物。如果已存在具有该名称的物品，请选择将车辆与 IoT 物品相关联以使用车辆更新该物品。或者，选择不同的车辆名称，AWS 物联网 FleetWise 将自动为车辆创建新事物。

2. 从列表中选择车辆型号（模型清单）。
3. 从列表中选择解码器清单。解码器清单与车辆型号关联。
4. （可选）要关联车辆属性，请选择添加属性。如果您跳过此步骤，则必须在创建车辆后添加属性，然后才能将其部署到活动中。
5. （可选）要将标签与车辆相关联，请选择添加新标签。您还可以在创建车辆后添加标签。
6. 选择下一步。

## 步骤 2：配置车辆证书

要将您的车辆当作 AWS IoT 物品使用，您必须配置带有附加政策的车辆证书。如果您跳过此步骤，则必须在创建车辆后配置证书，然后才能将其部署到活动中。

1. 选择自动生成新证书(推荐)。
2. 选择下一步。

## 步骤 3：将策略附加到证书

将策略附加到您在上一步中配置的证书。

1. 在策略中，请输入现有的策略名称。要创建新策略，请选择创建策略。

## 2. 选择下一步。

### 步骤 4：审核并创建

验证车辆的配置，然后选择创建车辆。

#### Important

创建车辆后，您必须下载证书和密钥。您将使用证书和私钥在 Edge Agent for AWS IoT FleetWise 软件中连接车辆。

## 创建车辆 (AWS CLI)

创建车辆时，必须使用与解码器清单关联的车辆型号。您可以使用 [CreateVehicle](#) API 操作来创建车辆。下面的示例使用了 AWS CLI。

要创建车辆，请运行以下命令。

*file-name* 替换为包含车辆配置的 .json 文件的名称。

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

### Example— 车辆配置

- ( 可选 ) `associationBehavior` 可以是以下值之一：
  - `CreateIotThing`— 创建车辆后，AWS 物联网 FleetWise 会自动为您的车辆创建带有您的车辆 ID 名称 AWS IoT 的事物。
  - `ValidateIotThingExists` – 使用现有 AWS IoT 物品创建车辆。

要创建 AWS IoT 事物，请运行以下命令。*thing-name* 用要创建的事物的名称替换。

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

如果未指定，AWS IoT FleetWise 会自动为您的车辆创建 AWS IoT 物品。

**⚠ Important**

确保在车辆创建后配置该 AWS IoT 物品。有关更多信息，请参阅 [配置 AWS 物联网 FleetWise 车辆](#)。

- *vehicle-name* 替换为以下内容之一。
- 您的 AWS IoT 事物的名称 ( 如果配置 `associationBehavior` 为 `ValidateIotThingExists` )。
- 要创建的车辆的 ID ( 如果 `associationBehavior` 配置为 `CreateIotThing` )。

车辆 ID 可以包含 1-100 个字符。有效字符：a—z、A—Z、0—9、短划线 (-)、下划线 (\_) 和冒号 (:)。

- *model-manifest-ARN* 替换为您的车辆型号 ( 车型清单 ) 的 ARN。
- 替换为 *decoder-manifest-ARN* 与指定车辆型号关联的解码器清单的 ARN。
- ( 可选 ) 您可以添加其他属性，将此车辆与使用相同车辆型号创建的其他车辆区分开来。例如，如果您有一辆电动汽车，则可以为属性指定以下值：`{"fuelType": "electric"}`。

**⚠ Important**

必须先关联的车辆型号中定义属性，然后才能将其添加到单个车辆中。

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

**Example— 将状态模板与车辆相关联**

您可以使用 `stateTemplates` 字段将 [状态模板](#) 与车辆相关联，以允许在云端收集车辆的状态更新。

在此示例中，*stateTemplateUpdateStrategy* 可以是以下之一：

- `periodic` : 允许您指定 Edge Agent 软件向云端发送信号更新的固定速率 ( 即使信号值在两次更新之间没有变化 , Edge Agent 软件也会发送更新 ) 。
- `onChange`: 只要信号发生变化 , Edge Agent 软件就会发送信号更新。

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

*create-vehicle.json*文件包含的位置 ( 例如 ) :

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

如果您[启用了使用客户托管 AWS KMS 密钥的加密](#) , 请包括以下政策声明 , 以便您的角色可以调用 CreateVehicle API 操作。

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"
    ]
  }
]
}

```

## 创建多 FleetWise 辆 AWS 物联网车辆

您可以使用 [BatchCreateVehicle](#) API 操作同时创建多辆车辆。下面的示例使用了 AWS CLI。

要创建多辆车，请运行以下命令。

*file-name* 替换为包含多辆车配置的 .json 文件的名称。

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

### Example— 车辆配置

```

{
  "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"
      }
    }
  ]
}

```

```
    }  
  }  
]  
}
```

可以为每个批处理操作创建最多 10 辆车。有关配置文件的更多信息，请参阅[创建物 AWS 联网 FleetWise 汽车](#)。

如果您[启用了使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 BatchCreateVehicle API 操作。

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"  
      ]  
    }  
  ]  
}
```

## 更新 AWS 物联网 FleetWise 车辆

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅[AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [UpdateVehicle](#) API 操作来更新现有车辆。下面的示例使用了 AWS CLI。

要更新车辆，请运行以下命令。

*file-name* 替换为包含您的车辆配置的 .json 文件的名称。

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

### Example— 车辆配置

- *vehicle-name* 替换为您要更新的车辆的 ID。
- ( 可选 ) *model-manifest-ARN* 替换为用于替换正在使用的车辆型号的车辆型号 ( 车型清单 ) 的 ARN。
- ( 可选 ) 替换为 *decoder-manifest-ARN* 与您指定的新车辆型号关联的解码器清单的 ARN。
- ( 可选 ) *attribute-update-mode* 替换为车辆属性。
- Merge – 通过使用新值更新现有属性，并在不存在新属性时添加新属性，将新属性合并到现有属性中。

例如，如果车辆具有以下属性：`{"color": "black", "fuelType": "electric"}`，并且您使用以下属性更新了车辆：`{"color": "", "fuelType": "gasoline", "model": "x"}`，则更新的车辆具有以下属性：`{"fuelType": "gasoline", "model": "x"}`。

- Overwrite – 用新属性替换现有属性。

例如，如果车辆具有以下属性：`{"color": "black", "fuelType": "electric"}`，并且您使用 `{"model": "x"}` 属性更新了车辆，则更新的车辆具有 `{"model": "x"}` 属性。

如果输入中存在属性，则这是必需的。

- ( 可选 ) 要添加新属性或使用新值更新现有值，请配置 `attributes`。例如，如果您有一辆电动汽车，则可以为属性指定以下值：`{"fuelType": "electric"}`。

要删除属性，请将 `attributeUpdateMode` 配置为 Merge。

#### Important

必须先关联的车辆型号中定义属性，然后才能将其添加到单个车辆中。

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

```
    }  
  }  
}
```

### Example— 添加或移除与车辆关联的状态模板

您可以使用以下字段关联其他状态模板或从车辆中移除现有关联：

- stateTemplatesToAdd
- stateTemplatesToRemove

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

*update-vehicle.json* 文件包含的位置（例如）：

```
{  
  "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"]  
}
```

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateVehicle API 操作。

### 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"
        ]
    }
}
}
```

## 更新多 FleetWise 辆 AWS 物联网车辆

您可以使用 [BatchUpdateVehicle](#) API 操作同时更新多辆现有车辆。下面的示例使用了 AWS CLI。

要更新多辆车，请运行以下命令。

*file-name* 替换为包含多辆车配置的.json 文件的名称。

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

### Example— 车辆配置

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

```
    }  
  }  
]  
}
```

每次批量操作最多可以更新 10 辆车。有关每辆车的配置的更多信息，请参阅[更新 AWS 物联网 FleetWise 车辆](#)。

如果您[启用了使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 BatchUpdateVehicle API 操作。

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"  
      ]  
    }  
  ]  
}
```

## 删除 AWS 物联网 FleetWise 车辆

您可以使用 AWS 物联网 FleetWise 控制台或 API 删除车辆。

### Important

车辆被删除后，AWS 物联网 FleetWise 会自动将车辆从关联的车队和活动中移除。有关更多信息，请参阅[在物联网中 AWS 管理车队 FleetWise](#)和[通过活动收集 AWS 物联网 FleetWise 数据](#)。但是，车辆仍然作为事物存在或仍与其中的事物相关联 AWS IoT Core。有关删除物品的说明，请参阅《AWS IoT Core 开发人员指南》中的[删除物品](#)。

## 删除车辆 (控制台)

您可以使用 AWS 物联网 FleetWise 控制台删除车辆。

### 删除车辆

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆。
3. 在车辆页面上，选择要删除的车辆旁边的按钮。
4. 选择删除。
5. 在删除 **vehicle-name** 中，输入车辆的名称，然后选择删除。

## 删除车辆 (AWS CLI)

您可以使用 [DeleteVehicle](#) API 操作删除车辆。以下示例使用 AWS CLI。

要删除车辆，请运行以下命令。

*vehicle-name* 替换为您要删除的车辆的 ID。

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

### 验证车辆是否被删除

您可以使用 [ListVehicles](#) API 操作来验证车辆是否已被删除。下面的示例使用了 AWS CLI。

要检索所有车辆摘要的分页列表，请运行以下命令。

```
aws iotfleetwise list-vehicles
```

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 ListVehicles API 操作。

### JSON

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

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

## 获取 AWS 物联网 FleetWise 车辆信息

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [GetVehicle](#) API 操作来检索车辆信息。下面的示例使用了 AWS CLI。

要检索车辆的元数据，请运行以下命令。

*vehicle-name* 替换为您要检索的车辆的 ID。

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

### Note

此操作具有 [最终一致性](#)。换句话说，可能不会立即反映对车辆进行的更改。

您可以使用 [GetVehicleStatus](#) API 操作来检索与车辆关联的资源的状态。下面的示例使用了 AWS CLI。

要检索与车辆关联的资源的状态，请运行以下命令。

- 替换为 *vehicle-name* 与资源关联的车辆的 ID。

- *type* 替换为要检索其状态的资源类型。type 的有效值为 CAMPAIGN、STATE\_TEMPLATE 和 DECODER。

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

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 GetVehicle 或 GetVehicleStatus API 操作。

## JSON

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

# 在物联网中 AWS 管理车队 FleetWise

车队表示一组车辆。没有关联车辆的车队是一个空实体。在使用车队同时管理多辆车之前，必须将车辆与该车队相关联。一辆车可以归属于多个车队。您可以通过部署活动来控制从车队中收集哪些数据以及何时收集数据。有关更多信息，请参阅 [通过活动收集 AWS 物联网 FleetWise 数据](#)。

车队包含以下信息。

`fleetId`

车队的 ID。

( 可选 ) `description`

可帮助您找到车队的描述。

`signalCatalogArn`

信号目录的 Amazon 资源名称 ( ARN ) 。

AWS IoT FleetWise 提供了以下 API 操作，您可以使用这些操作来创建和管理舰队。

- [CreateFleet](#)— 创建一组包含相同信号组的车辆。
- [AssociateVehicleFleet](#)— 将车辆与车队关联起来。
- [DisassociateVehicleFleet](#)— 断开车辆与车队的关联。
- [UpdateFleet](#)— 更新现有舰队的描述。
- [DeleteFleet](#)— 删除现有舰队。
- [ListFleets](#)— 检索所有舰队摘要的分页列表。
- [ListFleetsForVehicle](#)— 检索车辆所属的所有车队 IDs 的分页列表。
- [ListVehiclesInFleet](#)— 检索车队中所有车辆的分页摘要列表。
- [GetFleet](#)— 检索有关舰队的信息。

主题

- [创建 AWS 物联网 FleetWise 舰队](#)
- [将 AWS 物联网 FleetWise 车辆与车队关联](#)
- [解除 AWS 物联网 FleetWise 车辆与车队的关联](#)

- [更新 AWS 物联网 FleetWise 舰队](#)
- [删除 AWS 物联网 FleetWise 舰队](#)
- [获取 AWS 物联网 FleetWise 车队信息](#)

## 创建 AWS 物联网 FleetWise 舰队

您可以使用 [CreateFleet](#) API 操作来创建车队。以下示例使用 AWS CLI。

### Important

在创建车队之前，您必须拥有信号目录。有关更多信息，请参阅 [创建 AWS 物联网 FleetWise 信号目录](#)。

要创建车队，请运行以下命令。

- *fleet-id* 替换为您正在创建的舰队的 ID。

车队 ID 必须唯一，且包含 1-100 个字符。有效字符：字母 (A-Z 和 a-z)、数字 (0-9)、冒号 (:)、短划线 (-) 和下划线 (\_)。

- ( 可选 ) *description* 用描述替换。

描述可包含 1-2048 个字符。

- *signal-catalog-arn* 替换为信号目录的 ARN。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 CreateFleet API 操作。

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"
    ]
  }
]
```

## 将 AWS 物联网 FleetWise 车辆与车队关联

您可以使用 [AssociateVehicleFleet](#) API 操作将车辆与车队相关联。以下示例使用 AWS CLI。

### Important

- 拥有车辆和车队后，才能将车辆与车队关联起来。有关更多信息，请参阅 [管理 AWS 物联网 FleetWise 车辆](#)。
- 如果您将车辆与活动所针对的车队相关联，AWS IoT FleetWise 会自动将活动部署到该车辆。

要将车辆与车队关联，请运行以下命令。

- *fleet-id* 替换为舰队的 ID。
- *vehicle-name* 替换为车辆的 ID。

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

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 AssociateVehicleFleet API 操作。

## 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"
      ]
    }
  ]
}
```

## 解除 AWS 物联网 FleetWise 车辆与车队的关联

您可以使用 [DisassociateVehicleFleet](#) API 操作取消车辆与车队的关联。以下示例使用 AWS CLI。

要解除车辆与车队的关联，请运行以下命令。

- *fleet-id* 替换为舰队的 ID。
- *vehicle-name* 替换为车辆的 ID。

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

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 `DisassociateVehicleFleet` API 操作。

## 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"
  ]
}
```

## 更新 AWS 物联网 FleetWise 舰队

您可以使用 [UpdateFleet](#) API 操作来更新队列的描述。下面的示例使用了 AWS CLI。

要更新车队，请运行以下命令。

- *fleet-id* 替换为您要更新的舰队的 ID。
- *description* 替换为新的描述。

描述可包含 1-2048 个字符。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateFleet API 操作。

### 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"
    ]
  }
]
```

## 删除 AWS 物联网 FleetWise 舰队

您可以使用 [DeleteFleet](#) API 操作删除舰队。以下示例使用 AWS CLI。

### Important

在删除车队之前，请确保它没有关联的车辆。有关如何解除车辆与车队关联的说明，请参阅[解除 AWS 物联网 FleetWise 车辆与车队的关联](#)。

要删除车队，请运行以下命令。

*fleet-id* 替换为您要删除的舰队的 ID。

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

## 验证舰队已删除

您可以使用 [ListFleets](#) API 操作来验证队列是否已被删除。下面的示例使用了 AWS CLI。

要检索所有车队的分页摘要列表，请运行以下命令。

```
aws iotfleetwise list-fleets
```

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 ListFleets API 操作。

JSON

```
{
```

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

## 获取 AWS 物联网 FleetWise 车队信息

您可以使用 [ListFleetsForVehicle](#) API 操作来检索车辆所属的所有车队 IDs 的分页列表。下面的示例使用了 AWS CLI。

要检索该车辆所属 IDs 的所有车队的分页列表，请运行以下命令。

*vehicle-name* 替换为车辆的 ID。

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

如果您启用了使用客户托管 [AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 [ListFleetsForVehicle](#) API 操作。

### JSON

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

```

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

```

您可以使用 [ListVehiclesInFleet](#) API 操作来检索车队中所有车辆的分页摘要列表。下面的示例使用了 AWS CLI。

要检索车队中所有车辆的分页摘要列表，请运行以下命令。

*fleet-id* 替换为舰队的 ID。

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

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 ListVehiclesInFleet API 操作。

JSON

```

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

```

您可以使用 [GetFleet](#) API 操作来检索舰队信息。下面的示例使用了 AWS CLI。

要检索车队的元数据，请运行以下命令。

`fleet-id` 替换为舰队的 ID。

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

#### Note

此操作具有[最终一致性](#)。换句话说，可能不会立即反映对车队进行的更改。

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 GetFleet API 操作。

JSON

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

## 通过活动收集 AWS 物联网 FleetWise 数据

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

活动是指数据收集规则的编排。活动向 AWS 物联网边缘代理 FleetWise 软件提供了有关如何选择、收集数据并将其传输到云端的说明。

您要在云中创建活动。在您或您的团队批准活动后，AWS 物联网会将该活动 FleetWise 设置为准备部署，并将在下次办理车辆登机手续时部署。您可以选择将活动部署到一辆车还是部署到车队。在将正在运行的活动部署到车辆之前，边缘代理软件不会开始收集数据。

### ⚠ Important

只有在具备以下条件后，活动才会生效。

- 您的车辆中正在运行边缘代理软件。有关如何开发、安装和使用边缘代理软件的更多信息，请执行以下操作。
  1. 打开 [AWS 物联网 FleetWise 控制台](#)。
  2. 在服务主页的 AWS 物联网入门 FleetWise 部分，选择探索 Edge Agent。
- 您已设置 AWS IoT Core 好配置您的车辆。有关更多信息，请参阅 [配置 AWS 物联网 FleetWise 车辆](#)。

### 📘 Note

您还可以 [监控车辆的最近已知状态](#)（而不是舰队）使用状态模板近乎实时地使用状态模板，这些模板允许您使用“On Change”或“定期”更新策略流式传输遥测数据。该功能还提供“按需”功能，用于激活或停用先前部署的模板或一次性请求当前的车辆状态（获取）。对上次已知状态的访问目前处于封锁状态。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

每个活动包含以下信息。

### signalCatalogArn

与活动关联的信号目录的 Amazon 资源名称 (ARN)。

( 可选 ) tags

标签是可用于管理活动的元数据。您可以将同一标签分配给不同服务的资源，以指示这些资源是否相关。

### TargetArn

要部署活动的车辆或车队的 ARN。

name

有助于识别活动的唯一名称。

### collectionScheme

数据收集方案为 Edge Agent 软件提供了有关要收集哪些数据或何时收集数据的指示。AWS 物联网 FleetWise 目前支持基于条件的收集方案和基于时间的收集方案。

- **conditionBasedCollectionScheme**— 基于条件的收集方案使用逻辑表达式来识别要收集的数据。满足条件时，边缘代理软件会收集数据。
  - **expression**— 用于识别要收集哪些数据的逻辑表达式。例如，如果指定 `$variable.`myVehicle.InVehicleTemperature` > 50.0` 表达式，则边缘代理软件会收集大于 50.0 的温度值。有关如何编写表达式的说明，请参阅[AWS 物联网 FleetWise 活动的逻辑表达式](#)。
  - ( 可选 ) **conditionLanguageVersion**-条件表达式语言的版本。
  - ( 可选 ) **minimumTriggerIntervalMs**-两个数据收集事件之间的最短持续时间，以毫秒为单位。如果信号经常变动，则您可能要以较低速率收集数据。
  - ( 可选 ) **triggerMode**— 可以是以下值之一：
    - **RISING\_EDGE**— Edge Agent 软件仅在首次满足条件时才收集数据。例如 `$variable.`myVehicle.AirBagDeployed` == true`。
    - **ALWAYS** - 满足条件时，边缘代理软件会收集数据。
- **timeBasedCollectionScheme**— 定义基于时间的收集方案时，请指定以毫秒为单位的时间段。边缘代理软件使用该时间段来决定收集数据的频率。例如，如果时间段为 120,000 毫秒，则边缘代理软件会每隔两分钟收集一次数据。
  - **periodMs**— 决定数据收集频率的时间段（以毫秒为单位）。

## ( 可选 ) compression

为了节省无线带宽并减少网络流量，您可以指定 [SNAPPY](#) 来压缩车辆中的数据。

默认情况下 (OFF)，边缘代理软件不压缩数据。

## dataDestinationConfigs

选择广告活动将传输车辆数据的单个目的地。您可以将数据发送到 [MQTT 主题](#)，也可以将其存储在 Amazon S3 或 Amazon Timestream 中。

MQTT ( 消息队列遥测传输 ) 是一种轻量级且被广泛采用的消息传递协议。您可以将数据发布到 MQTT 主题，以使用规则建立自己的事件驱动架构。AWS IoT 对 MQTT 的支持基于 [MQTT v3.1.1 规范](#) 和 [MQTT v5.0 规范](#)，但有一些区别。有关更多信息，请参阅 [MQTT 的区别](#)。

S3 可以是一种经济实惠的数据存储机制，可提供持久的数据管理功能和下游数据服务。您可以使用 S3 获取与驾驶行为或分析长期维护相关的数据。

Timestream 是一种数据持久化机制，可以帮助您近乎实时地识别趋势和模式。您可以使用 Timestream 获取时间序列数据，执行车速或制动历史趋势分析等操作。

### Note

Amazon Timestream 不在亚太地区 ( 孟买 ) 区域推出。

## ( 可选 ) dataExtraDimensions

您可以添加一个或多个属性，为信号提供其他信息。

## ( 可选 ) dataPartitions

创建数据分区以临时存储车辆上的信号数据。您可以配置何时以及如何将数据转发到云端。

- 通过定义最大 FleetWise 存储大小、最短存活时间和存储位置，指定 AWS IoT 如何将数据存储在车辆或车队上。
- 竞选活动 spoolingMode 必须是 TO\_DISK。
- 上传配置包括定义条件语言和逻辑表达式的版本。

## ( 可选 ) description

添加描述以帮助确定广告活动的目的。

### ( 可选 ) diagnosticsMode

将诊断模式配置为后SEND\_ACTIVE\_DTCS，活动会发送存储的标准诊断故障代码 (DTCs)，以帮助识别您的车辆出了什么问题。例如，P0097 表示发动机控制模块 (ECM) 已确定进气温度传感器 2 (IAT2) 输入低于正常传感器范围。

默认情况下 (OFF)，边缘代理软件不发送诊断代码。

### ( 可选 ) expiryTime

定义广告活动的到期日期。活动到期后，边缘代理软件将停止收集此活动中指定的数据。如果在车辆上部署了多个活动，则边缘代理软件会使用其他活动来收集数据。

默认值：253402243200 ( 9999 年 12 月 31 日，世界标准时间 00:00:00 )

### ( 可选 ) postTriggerCollectionDuration

您可以定义触发后的收集持续时间，以便边缘代理软件在调用方案后的指定时间段内继续收集数据。例如，如果使用以下表达式调用基于条件的收集方案：`$variable.`myVehicle.Engine.RPM` > 7000.0`，则边缘代理软件将继续收集引擎的每分钟转数 (RPM) 值。即使 RPM 只高于 7000 一次，也可能表明存在机械问题。在这种情况下，您可能希望边缘代理软件继续收集数据以帮助监控情况。

默认值：0

### ( 可选 ) priority

指定一个整数来表示广告系列的优先级。数值越小，活动的优先级越高。如果在车辆上部署多个活动，则优先级较高的活动将先启动。

默认值：0

### ( 可选 ) signalsToCollect

调用数据收集方案时从中收集数据的信号列表。

- name— 调用数据收集方案时从中收集数据的信号的名称。
- dataPartitionId— 要在信号中使用的数据分区的 ID。该身份证必须与中 IDs 提供的其中一个相匹配dataPartitions。如果您将信号作为条件上传到数据分区，则必须将相同的信号包含在数据分区中signalsToCollect。
- ( 可选 ) maxSampleCount— 调用数据收集方案时，Edge Agent 软件收集并传输到云端的最大数据样本数。
- ( 可选 ) minimumSamplingIntervalMs— 两个数据样本收集事件之间的最短持续时间，以毫秒为单位。如果信号经常变动，则您可能要使用此参数以较低速率收集数据。

有效范围：0-4294967295

( 可选 ) `spoolingMode`

如果 `spoolingMode` 配置为 `T0_DISK`，那么在车辆未连接到云端时，边缘代理软件会临时将数据存储在本机。重新建立连接后，存储在本机的数据将自动传输到云端。

默认值：OFF

( 可选 ) `startTime`

已批准的活动将在开始时间激活。

默认值：0

活动状态可以是以下值之一：

- `CREATING`— AWS IoT FleetWise 正在处理您创建活动的请求。
- `WAITING_FOR_APPROVAL` - 活动在创建后即进入 `WAITING_FOR_APPROVAL` 状态。要批准该活动，请使用 `UpdateCampaign` API 操作。活动获得批准后，AWS IoT FleetWise 会自动将活动部署到目标车辆或车队。有关更多信息，请参阅 [更新物 AWS 联网 FleetWise 活动](#)。
- `RUNNING` - 活动已激活。
- `SUSPENDED` - 活动已暂停。要批准该活动，请使用 `UpdateCampaign` API 操作。

AWS IoT FleetWise 提供了以下 API 操作，您可以使用这些操作来创建和管理活动。

- [CreateCampaign](#)— 创建新的广告系列。
- [UpdateCampaign](#)— 更新现有广告系列。创建活动后，您必须使用此 API 操作来批准该活动。
- [DeleteCampaign](#)— 删除现有广告系列。
- [ListCampaigns](#)— 检索所有广告系列的分页摘要列表。
- [GetCampaign](#)— 检索有关广告系列的信息。

教程

- [创建物 AWS 联网 FleetWise 活动](#)
- [更新物 AWS 联网 FleetWise 活动](#)
- [删除物 AWS 联网 FleetWise 活动](#)

- [获取 AWS 物联网 FleetWise 活动信息](#)
- [存储和转发广告系列数据](#)
- [使用 AWS IoT 收集诊断故障代码数据 FleetWise](#)
- [可视化 AWS 物联网 FleetWise 车辆数据](#)

## 创建物 AWS 联网 FleetWise 活动

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 AWS 物联网 FleetWise 控制台或 API 创建活动来收集车辆数据。

### Important

要使活动发挥作用，必须满足以下条件：

- 您的车辆中正在运行边缘代理软件。有关如何开发、安装和使用边缘代理软件的更多信息，请执行以下操作：
  1. 打开[AWS 物联网 FleetWise 控制台](#)。
  2. 在服务主页的 AWS 物联网入门 FleetWise 部分，选择探索 Edge Agent。
- 您已设置 AWS IoT Core 好配置您的车辆。有关更多信息，请参阅 [配置 AWS 物联网 FleetWise 车辆](#)。

### 主题

- [创建活动 \(控制台\)](#)
- [创建活动 \(AWS CLI\)](#)
- [AWS 物联网 FleetWise 活动的逻辑表达式](#)

## 创建活动 (控制台)

使用 AWS 物联网 FleetWise 控制台创建活动以选择、收集车辆数据并将其传输到云端。

## 创建市场活动

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择活动。
3. 在活动页面上，选择创建活动系列，然后完成以下主题中的步骤。

### 主题

- [步骤 1：配置活动](#)
- [第 2 步：指定存储和上传条件](#)
- [步骤 3：配置数据目标](#)
- [第 4 步：添加车辆](#)
- [第 5 步：审核并创建](#)
- [第 6 步：部署活动](#)

#### Important

- 在创建活动之前，您必须拥有信号目录和车辆。有关更多信息，请参阅[管理 AWS 物联网 FleetWise 信号目录](#)和[管理 AWS 物联网 FleetWise 车辆](#)。
- 创建活动后，必须批准该活动。有关更多信息，请参阅[更新物 AWS 联网 FleetWise 活动](#)。

## 步骤 1：配置活动

在一般信息部分，执行以下操作：

1. 输入活动的名称。
2. (可选) 输入描述。

配置活动的数据收集方案。数据收集方案向边缘代理软件提供有关要收集哪些数据，以及何时收集数据的指令。在 AWS 物联网 FleetWise 控制台中，您可以通过以下方式配置数据收集方案：


- 手动定义数据收集方案。
- 上传文件以自动定义数据收集方案。

在配置选项中，选择以下选项之一：

- 要手动指定数据收集方案的类型并定义用于自定义该方案的选项，请选择定义数据收集方案。

手动指定数据收集方案的类型，并定义用于自定义该方案的选项。

1. 在数据收集方案详细信息部分，选择您希望此活动使用的数据收集方案的类型。要使用逻辑表达式识别要收集的车辆数据，请选择基于条件。要使用特定的时间段来决定收集车辆数据的频率，请选择基于时间。
2. 定义活动收集数据的持续时间。

 Note

默认情况下，已批准的活动会立即激活，并且没有设定的结束时间。为避免额外费用，您必须指定时间范围。

3. 如果您指定了基于条件的数据收集方案，则必须定义逻辑表达式来识别要收集的数据。AWS 物联网 FleetWise 使用逻辑表达式来识别要为基于条件的方案收集哪些数据。该表达式必须将信号的完全限定名称指定为变量、比较运算符和比较值。

例如，如果您指定 `$variable.`myVehicle.InVehicleTemperature` > 50.0` 表达式，AWS IoT FleetWise 会收集大于 50.0 的温度值。有关如何编写表达式的说明，请参阅 [AWS 物联网 FleetWise 活动的逻辑表达式](#)。


输入用于识别要收集哪些数据的逻辑表达式。

4. (可选) 指定条件表达式的语言版本。默认值是 1。
5. (可选) 指定最小触发间隔，即两个数据收集事件之间的最小持续时间。例如，如果信号经常变动，则您可能要以较低速率收集数据。
6. 指定边缘代理软件收集数据的触发模式条件。默认情况下，AWS 物联网边缘代理 FleetWise 软件始终在满足条件时收集数据。或者，它只在第一次满足条件时收集数据，即首次触发时。
7. 如果您指定了基于时间的数据收集方案，则必须指定一个介于 10000 - 60000 毫秒之间的时间段。边缘代理软件使用该时间段来决定收集数据的频率。
8. (可选) 编辑方案的高级方案选项。
  - a. 为了节省无线带宽并减少网络流量，您可以指定 SNAPPY 来压缩数据。
  - b. (可选) 要定义数据收集事件发生后继续收集数据的时间（以毫秒为单位），您可以指定触发后收集持续时间。

- c. (可选) 要指明活动的优先级, 请指定活动优先级。优先级数字较小的活动将首先部署, 并被认为具有更高的优先级。
  - d. 在车辆未连接到云端时, 边缘代理软件会临时将数据存储在本地。重新建立连接后, 存储在本地的数据将自动传输到云端。指定是否希望边缘代理软件在连接中断期间在本地存储数据。
  - e. (可选) 要为信号提供更多信息, 请将最多五个属性添加为额外数据维度。
- 要上传文件以定义数据收集方案, 请选择从本地设备上传.json 文件。AWS IoT FleetWise 会自动定义您可以在文件中定义哪些选项。您可以查看和更新所选的选项。

上传包含数据收集方案详细信息的 .json 文件。

1. 要导入有关数据收集方案的信息, 请选择选择文件。有关所需文件格式的更多信息, 请参阅 [CreateCampaignAPI](#) 文档。

 Note

AWS 物联网 FleetWise 目前支持.json 文件格式扩展名。

2. AWS IoT FleetWise 会根据您的文件中的信息自动定义数据收集方案。查看 AWS IoT 为您 FleetWise 选择的选项。如果需要, 您可以更新选项。

## 第 2 步：指定存储和上传条件

要选择当车辆未连接到云端时, Edge Agent 软件是否将数据临时存储在本地, 请指定后台处理模式。

- 在数据后台处理模式下, 选择以下选项之一：
  - 未存储 — 当车辆离线时, Edge Agent 软件会收集数据, 但不会在本地临时存储数据。当车辆重新连接时, Edge Agent 软件会将数据传输到云端。
  - 存储到磁盘 — 当车辆离线时, Edge Agent 软件会在本地收集和临时存储数据。收集的数据临时存储在 Edge Agent 配置文件“持久性”部分定义的位置。当车辆重新连接时, Edge Agent 会将数据传输到云端。
  - 使用分区存储到磁盘 — 车辆始终将边缘上的数据临时存储在您指定的数据分区中。您可以选择何时将存储的数据转发到云端。
    1. (可选) 输入分区 ID 以指定一组特定的数据。
    2. 输入文件夹名称作为存储数据的位置。存储位置的绝对路径是{persistence\_path} / {vehicle\_name} / {campaign\_name} / {storage\_location}。

3. 输入存储在分区中的数据的最大存储大小。当分区达到最大大小时，较新的数据会覆盖较旧的数据。
4. 输入此分区中的数据在磁盘上保存的最短时间。
5. ( 可选 ) 输入分区的上传条件。

## 指定信号

在活动期间，您可以指定要从中收集数据的信号。

### 指定要从中收集数据的信号

1. 选择信号名称。
2. ( 可选 ) 在“最大样本数”中，输入 Edge Agent 软件在活动期间收集并传输到云端的最大数据样本数。
3. ( 可选 ) 在最小采样间隔中，输入两个数据样本收集事件之间的最小持续时间 ( 以毫秒为单位 )。如果信号经常变动，则您可能要使用此参数以较低速率收集数据。
4. 要添加其他信号，请选择添加更多信号。您最多可以添加 999 个信号。
5. 选择下一步。

## 步骤 3：配置数据目标

### Note

如果活动包含视觉系统数据信号，则只能将车辆数据存储于 Amazon S3 中。你不能将其存储在 Timestream 中，也不能将其发送到 MQTT 主题。

视觉系统数据目前为预览版，可能会发生变化。

Amazon Timestream 不在亚太地区 ( 孟买 ) 区域推出。

选择您要发送或存储活动收集的数据的目的地。您可以将车辆数据发送到 MQTT 主题，也可以将其存储在 Amazon S3 或 Amazon Timestream 中。

在目标设置中，执行以下操作：

- 从下拉列表中选择 Amazon S3、Amazon Timestream 或 MQTT 主题。

## Amazon S3

### Important

只有当 AWS IoT FleetWise 有权写入 S3 存储桶时，您才能将数据传输到 S3。有关授予访问权限的更多信息，请参阅[使用 AWS IoT 控制访问权限 FleetWise](#)。

要在 S3 存储桶中存储车辆数据，请选择 Amazon S3。S3 是一种对象存储服务，可将数据以对象形式存储在存储桶中。有关更多信息，请参阅《[亚马逊简单存储服务用户指南](#)》中的[创建、配置和使用 Amazon S3 存储桶](#)。

S3 优化了数据存储成本，并提供了其他车辆数据使用机制，例如数据湖、集中式数据存储、数据处理管道和分析。您可以使用 S3 存储数据以进行批处理和分析。例如，您可以为机器学习 (ML) 模型创建硬刹车事件报告。传入的车辆数据在交付前会缓冲 10 分钟。

在 S3 目标设置中，执行以下操作：

1. 对于 S3 存储桶，请选择 AWS IoT FleetWise 有权访问的存储桶。
2. (可选) 输入可用于整理 S3 存储桶中存储的数据的自定义前缀。
3. 选择输出格式，即保存在 S3 存储桶中的格式文件。
4. 选择是否要将存储在 S3 存储桶中的数据压缩为 .gzip 文件。建议压缩数据，因为这样可以最大限度降低存储成本。
5. 您在 S3 目标设置中选择的选项会更改示例 S3 对象 URI。这是将哪些文件保存在 S3 中的示例。

## Amazon Timestream

### Important

只有当 AWS 物联网 FleetWise 有权将数据写入 Timestream 时，您才能将数据传输到表。有关授予访问权限的更多信息，请参阅[使用 AWS IoT 控制访问权限 FleetWise](#)。

Amazon Timestream 不在亚太地区 (孟买) 区域推出。

要将车辆数据存储在 Timestream 表中，请选择 Amazon Timestream。您可以使用 Timestream 查询车辆数据，以便识别趋势和模式。例如，您可以使用 Timestream 来创建车辆燃油水平警报。传入的车辆数据将近乎实时地传输到 Timestream。有关更多信息，请参阅[什么是亚马逊 Time stream?](#) 在《[亚马逊 Timestream 开发者指南](#)》中。

在 Timestream 表设置中，执行以下操作：

1. 对于 Timestream 数据库名称，请从下拉列表中选择您的 Timestream 数据库的名称。
2. 对于 Timestream 表名称，请从下拉列表中选择您的 Timestream 表的名称。

在 Timestream 的服务访问权限中，执行以下操作：

- 从下拉列表选择 IAM 角色。

## MQTT 主题

### Important

只有当 IoT FleetWise 拥有主题权限时，您才能将数据路由到 MQTT AWS T AWS IoT 主题。有关授予访问权限的更多信息，请参阅[使用 AWS IoT 控制访问权限 FleetWise](#)。

要向 MQTT 主题发送车辆数据，请选择 MQTT 主题。

由 MQTT 消息发送的车辆数据以近乎实时的方式传送，允许您使用规则采取行动，或将数据路由到其他目的地。有关使用 MQTT 的更多信息，请参阅《AWS IoT Core 开发人员指南》AWS IoT 中的[设备通信协议](#)和[规则](#)。

1. 在 MQTT 主题下，输入主题名称。
  2. 在 MQTT 的服务访问权限主题下，选择是否要让 AWS IoT 为您 FleetWise 创建和使用新的服务角色。如果要使用现有的服务角色，请在选择角色下的下拉列表中选择该角色。
- 选择下一步。

## 第 4 步：添加车辆

要选择要将活动部署到哪些车辆，请在车辆列表中选择它们。通过搜索您在创建车辆时添加的属性及其值或按车辆名称筛选车辆。

在筛选车辆中，执行以下操作：

1. 在搜索框中，查找属性或车辆名称，然后从列表中进行选择。

**Note**

每个属性只能使用一次。

2. 输入要部署活动的属性值或车辆名称。例如，如果属性的完全限定名称为 `fuelType`，则输入 `gasoline` 作为其值。
3. 要搜索其他车辆属性，请重复上述步骤。您最多可以搜索五个车辆属性和数量不限的车辆名称。
4. 符合您搜索条件的车辆列在车辆名称下。选择您想要将活动部署到的车辆。

**Note**

搜索结果中最多显示 100 辆车。选择全选，将所有车辆添加到活动中。

5. 选择下一步。

## 第 5 步：审核并创建

验证活动的配置，然后选择创建活动。

**Note**

创建活动后，您或您的团队必须将活动部署到车辆。

## 第 6 步：部署活动

创建活动后，您或您的团队必须将活动部署到车辆。

### 部署活动

1. 在活动摘要页面上，选择部署。
2. 查看并确认您要开始部署并从与活动关联的车辆收集数据。
3. 选择部署。

如果您想暂停从与活动关联的车辆收集数据，请在活动摘要页面上选择暂停。要继续从与活动关联的车辆收集数据，请选择继续。

## 创建活动 (AWS CLI)

您可以使用 [CreateCampaign](#) API 操作来创建广告活动。下面的示例使用了 AWS CLI。

创建活动，从车辆收集的数据可以发送到 MQTT 主题或存储在亚马逊 S3 (S3) 或 Amazon Timestream 中。选择 Timestream 可获得快速、可扩展且无需服务器的时间序列数据库，例如存储需要近乎实时处理的数据。选择 S3 作为具有业界领先的可扩展性、数据可用性、安全性和性能的对象存储。选择 MQTT 以近乎实时的方式交付数据，并使用[规则](#)执行您定义的操作或将数据路由到其他目的地。AWS IoT

### Important

只有当 AWS 物联网 FleetWise 有权代表你发送 MQTT 消息或向 S3 或 Timestream 写入数据时，你才能将车辆数据传输到 MQTT 主题、Amazon S3 或 Amazon Timestream。有关授予访问权限的更多信息，请参阅[使用 AWS IoT 控制访问权限 FleetWise](#)。  
Amazon Timestream 不在亚太地区（孟买）区域推出。

## 创建活动

### Important

- 在创建活动之前，您必须拥有信号目录和车辆或车队。有关更多信息，请参阅[管理 AWS 物联网 FleetWise 信号目录](#)、[管理 AWS 物联网 FleetWise 车辆](#) 和 [在物联网中 AWS 管理车队 FleetWise](#)。
- 创建活动后，您必须使用 UpdateCampaign API 操作来批准该活动。有关更多信息，请参阅[更新物 AWS 物联网 FleetWise 活动](#)。

要创建活动，请运行以下命令。

*file-name* 替换为包含活动配置的 .json 文件的名称。

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

- campaign-name* 替换为你正在创建的广告活动的名称。

- *signal-catalog-arn* 替换为信号目录的亚马逊资源名称 (ARN)。
- *target-arn* 替换为您创建的车队或车辆的 ARN。
- *bucket-arn* 替换为 S3 存储桶的 ARN。

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

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

**Note**

Amazon Timestream 不在亚太地区 ( 孟买 ) 区域推出。

- *campaign-name* 替换为你正在创建的广告活动的名称。
- *signal-catalog-arn* 替换为信号目录的 ARN。
- *target-arn* 替换为您创建的车队或车辆的 ARN。
- *role-arn* 替换为授予 Io AWS T 向 Timestream 表传送数据的 FleetWise 权限的任务执行角色的 ARN。
- *table-arn* 替换为时间流表的 ARN。

```
{
  "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* 替换为你正在创建的广告活动的名称。
- *signal-catalog-arn* 替换为信号目录的亚马逊资源名称 (ARN)。
- *target-arn* 替换为您创建的车队或车辆的 ARN。
- *topic-arn* 替换为您指定为包含车辆数据的消息目标的 [MQTT 主题](#) 的 ARN。
- *role-arn* 替换为任务执行角色的 ARN，该角色授予 Io AWS T 针对您指定的 MQTT 主题发送、接收消息和对消息采取操作的 FleetWise 权限。

```

{
  "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"
    }
  }
]
}
```

如果您[启用了使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 CreateCampaign API 操作。

## 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"
  ]
}
]
}

```

## AWS 物联网 FleetWise 活动的逻辑表达式

AWS 物联网 FleetWise 使用逻辑表达式来识别在活动中要收集哪些数据。有关表达式的更多信息，请参阅《AWS IoT Events 开发人员指南》中的[表达式](#)。

表达式变量的结构应符合所收集数据类型的规则。对于遥测系统数据，表达式变量应为信号的完全限定名称。对于视觉系统数据，表达式会将信号的完全限定名称与从信号的数据类型到其某个属性的路径相结合。

例如，如果信号目录包含以下节点：

```

{
  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"
}

```

如果节点符合 ROS 2 定义：

```

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

```

```
}
```

以下是所有可能的事件表达式变量：

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

## 更新物 AWS 联网 FleetWise 活动

您可以使用 [UpdateCampaign](#) API 操作来更新现有广告系列。以下命令使用 AWS CLI。

- *campaign-name* 替换为你要更新的广告系列的名称。
- 将 *action* 替换为以下各项之一：
  - APPROVE— 批准活动以允许 AWS 物联网将其部署 FleetWise 到车辆或车架上。
  - SUSPEND - 暂停活动。从车辆中删除活动，所有处于暂停状态的活动中的所有车辆都将停止发送数据。
  - RESUME - 重新激活处于 SUSPEND 状态的活动。该活动将在下次办理登机手续时重新部署到所有车辆上，车辆将恢复发送数据。
  - UPDATE— 通过定义属性并将其与广告活动关联来更新广告系列。
- *description* 替换为新的描述。

描述最多可以包含 2,048 个字符。

- *data-extra-dimensions* 替换为指定的车辆属性，以丰富活动期间收集的数据。例如，您可以将车辆品牌和型号添加到广告活动中，AWS 物联网 FleetWise 会将数据与这些属性关联为 Amazon Timestream 中的维度。然后，您可以根据车辆品牌和型号查询数据。

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

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 UpdateCampaign API 操作。

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"  
      ]  
    }  
  ]  
}
```

## 删除物 AWS 联网 FleetWise 活动

您可以使用 AWS IoT FleetWise 控制台或 API 删除活动。

### 创建活动（控制台）

要删除活动，请使用 AWS IoT FleetWise 控制台。

## 删除活动

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择活动。
3. 在活动页面上，选择目标活动。
4. 选择删除。
5. 在删除 **campaign-name?** 中，输入要删除的活动的名称，然后选择确认。

## 删除活动 (AWS CLI)

您可以使用 [DeleteCampaign](#) API 操作删除广告活动。以下示例使用 AWS CLI。

要创建活动，请运行以下命令。

*campaign-name* 替换为您要删除的车辆名称。

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

### 已删除的数据分区不可恢复

删除活动会移除设备上的所有数据，分区中的数据不会上传到云端。

## 确认广告活动已删除

您可以使用 [ListCampaigns](#) API 操作来验证广告活动是否已被删除。下面的示例使用了 AWS CLI。

要检索所有活动的分页摘要列表，请运行以下命令。

```
aws iotfleetwise list-campaigns
```

## 获取 AWS 物联网 FleetWise 活动信息

您可以使用 [GetCampaign](#) API 操作来检索车辆信息。下面的示例使用了 AWS CLI。

要检索活动的元数据，请运行以下命令。

*campaign-name* 替换为您要检索的广告系列的名称。

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

### Note

此操作具有最终一致性。换句话说，可能不会立即反映对活动进行的更改。

如果您启用了使用客户托管 AWS KMS 密钥的加密，请包括以下政策声明，以便您的角色可以调用 GetCampaign API 操作。

### JSON

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

## 存储和转发广告系列数据

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

在战役中使用数据分区将车辆和车队的信号数据临时存储在 Edge 上。通过为数据分区配置上传和存储选项，您可以优化将数据转发到指定数据目的地（例如 Amazon S3 存储桶）的理想条件。例如，您可

以将数据分区配置为在车辆连接到 Wi-Fi 之前将数据存储存储在车辆上。然后，车辆连接后，活动会触发该特定分区中的数据发送到云端。或者，您可以使用 AWS IoT Jobs 收集数据。

## 主题

- [创建数据分区](#)
- [上传广告活动数据](#)
- [使用 AWS IoT 任务上传数据](#)

## 创建数据分区

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

活动中的数据分区会临时存储信号数据。您可以配置何时以及如何将数据转发到云端。

数据分区的工作原理是首先使用活动指定一组特定的数据。dataPartitionId然后，您可以进一步定义分区存储选项，例如最大大小、使数据分区保持活动状态的最短时间（在磁盘上）以及在 Edge 上存储数据的位置。您可以使用确定车辆上的存储位置storageLocation。存储位置决定了活动存储文件夹下数据分区的文件夹名称。战役存储文件夹位于 Edge 配置文件中定义的持久性路径下以车辆名称命名的文件夹下。这是存储位置的绝对路径：`{persistence_path} / {vehicle_name} / {campaign_name} / {storage_location}`。

假脱机模式设置为TO\_DISK指定应将分区后的数据保存到车辆上的磁盘中。数据分区的数据存储以FIFO（先进先出）为基础运行。如果您删除某个活动，则还会删除关联数据分区中的数据。如果您没有为连接 on/off 用例指定数据分区，则当没有连接时，AWS 物联网 FleetWise 仍将数据存储存储在车辆上的环形缓冲区中。恢复连接后，物 AWS 联网会将数据 FleetWise 上传到云端。此行为可在适用于 AWS 物联网的边缘代理 FleetWise 软件中进行配置。

### Important

如果您的数据分区超过设置的最大存储限制，则当分区达到最大大小时，较新的数据会覆盖较旧的数据。Edge 上丢失的数据无法恢复。存储大小由您的 Edge 存储限制决定。当数据上传到云端时，可以在最短的实时时间过去后将其删除。适当设置最短存活时间，以避免意外删除。

上传选项决定变量表达式和条件语言。如果指定了上传选项，则还必须指定存储选项。您也可以请求将数据分区中的信号上传到云端。有关更多信息，请参阅 [上传广告活动数据](#)。

定义数据分区条件后，`signalsToCollect`可以帮助指定要在数据分区中考虑哪些信号。您可以 IDs 为数据分区指定，也可以将设置`dataPartitionIddefault`为使用已建立的默认数据分区。没有指定的信号`dataPartitionId`将与默认信号相关联`dataPartition`。

## 创建数据分区

使用以下示例，创建具有数据分区存储条件的活动。此示例活动配置为在亚马逊 Timestream 中存储车辆数据。

1. `campaign-name`替换为你正在创建的广告活动的名称。
2. ( 可选 ) 提供说明。
3. `role-arn`替换为任务执行角色的 Amazon 资源名称 (ARN)，该角色授予物 AWS 联网向 Timestream 表传输数据的 FleetWise 权限。
4. `table-arn`替换为时间流表的 ARN。
5. `signal-catalog-arn`替换为信号目录的 ARN。
6. `data-partition-id`同时替换 `dataPartitions` ID 和要关联的 ID `signalsToCollect`。首先，替换要在信号中使用的数据分区的 ID。对于`signalsToCollect`，ID 必须与中 IDs 提供的其中一个相匹配`dataPartitions`。

### Note

使用`default`作为 ID，为广告活动建立默认数据分区。

7. `target-arn`替换为您创建的车队或车辆的 ARN。

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

满足所有指定条件后，分区后的数据将转发到云端，从而可以收集和存储新的分区信号。

接下来，您将调用 UpdateCampaign API 将其部署到适用于 AWS 物联网 FleetWise 软件的 Edge Agent。有关更多信息，请参阅 [上传广告活动数据](#)。

## 上传广告活动数据

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

在 Edge 上上传广告系列数据的方法有两种：

- 符合您的上传条件的广告活动将在获得批准后自动将数据上传到云端。要批准广告活动，请使用 updateCampaign API 操作。
- 通过 AWS IoT Jobs，即使不满足指定条件，您也可以强制上传数据。有关更多信息，请参阅 [使用 AWS IoT 任务上传数据](#)。

### 使用 UpdateCampaign API 操作上传广告活动数据

创建广告活动后，广告活动状态将显示为，WAITING\_FOR\_APPROVAL直到您将其更改action为APPROVED。

- 使用以下示例action通过调用 [UpdateCampaign](#)API 操作来更新广告系列。

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

## 使用 AWS IoT 任务上传数据

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

借 AWS IoT 助 Jobs，您可以将活动配置为在需要时将存储的车辆数据上传到云端。

为竞选活动创建工作文档

- 使用以下示例为竞选活动创建工作文档。作业文档是一个.json 文件，其中包含有关执行任务所需的车辆或车队的信息。有关创建作业文档的更多信息，请参阅[使用AWS IoT 开发人员指南 AWS CLI中的创建和管理作业](#)。

要请求只有一辆车上传数据，请将任务目标设置为与车辆关联 AWS IoT 的事物。要请求多辆车（在同一个广告系列中）上传数据，请创建一个包含与车辆对应的所有事物的事物组，然后将任务目标设置为该事物组。

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

- a. CampaignArn替换为同一地区和账户中的活动的 Amazon 资源名称 (ARN)。广告活动 ARN 为必填项。
- b. ( 可选 ) endTime替换为车辆上收集的数据的时间戳，采用 ISO 8601 UTC 格式 ( 不含毫秒 )。例如 2024-03-05T23:00:00Z。时间戳是排他性的，它决定了要上传的最后一个数据点。如果省略endTime，Edge Agent 软件将继续上传，直到所有广告系列存储的数据都上传完毕。上传所有数据后，它会将[任务执行状态](#)更新为SUCCEEDED。作业的[状态](#)更新为COMPLETED。

## 使用托管作业模板创建作业

1. CollectCampaignData从托管模板列表中选择 IoT-iO TFleet Wise-。有关更多信息，请参阅《AWS IoT 开发人员指南》中的 [AWS 通过托管模板创建作业](#)。
2. 托管模板具有CampaignArn和endTime参数。
  - a. CampaignArn替换为同一地区和账户中的活动的 Amazon 资源名称 (ARN)。广告活动 ARN 为必填项。
  - b. ( 可选 ) endTime替换为车辆上收集的数据的时间戳，采用 ISO 8601 UTC 格式 ( 不含毫秒 )。例如 2024-03-05T23:00:00Z。时间戳是排他性的，它决定了要上传的最后一个数据点。如果省略endTime，Edge Agent 软件将继续上传，直到所有广告系列存储的数据都上传完毕。上传所有数据后，它会将[任务执行状态](#)更新为SUCCEEDED。作业的[状态](#)更新为COMPLETED。

有关疑难解答的相关主题，请参阅[存储和转发问题](#)。

有关 AWS IoT 作业的更多信息，请参阅《AWS IoT 开发人员指南》中的[作业](#)。

## 使用 AWS IoT 收集诊断故障代码数据 FleetWise

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

当车辆检测到错误时，它会生成诊断故障代码 (DTC)，并记录受影响的传感器或执行器的快照。DTCs 帮助您近乎实时地了解错误，了解导致错误的原因并采取纠正措施。AWS 物联网 FleetWise 支持通过数据收集活动收集 DTCs，包括相应的 DTC 快照和扩展数据。本主题介绍了便于 DTC 数据收集的概念、工作流程和关键字，并通过示例进行了说明。

以下显示了使用 DTC 的关键概念。

### 自定义函数

自定义函数是指能够调用和执行自己在 Edge Agent 上预定义的函数，从而扩展了[自定义解码](#)概念。这些功能与 AWS 物联网 FleetWise 代理配合使用。适用于 AWS 物联网的 Edge Agent FleetWise 软件提供了用于计算信号统计数据的内置功能，例如最小值、最大值和平均值。自定义

函数允许您为特定用例创建量身定制的逻辑，从而扩展了此功能。对于诊断故障代码 (DTC) 数据收集，开发人员可以利用自定义函数来实现高级数据检索机制，例如通过统一诊断服务 (UDS) 或其他诊断接口直接从车辆边缘获取 DTC 代码、快照和扩展数据。

有关更多信息，请参阅《边缘代理开发人员 [指南](#)》中的 [自定义函数指南](#) 和 [DTC 数据收集参考实现](#)。

## 信号获取

在数据收集活动中，通常会从设备连续收集信号，并在 Edge Agent 软件上进行缓冲。然后，信号会定期上传或存储在基于时间的广告系列中，或者在基于条件的广告系列中由特定条件触发。但是，由于担心设备流量拥塞，无法从设备收集和持续缓冲 DTC 信号。为了解决这个问题，AWS 物联网 FleetWise 提供了信号提取功能，可确保不连续地从设备获取目标信号。

信号获取支持周期性和条件驱动的动作。您可以使用自定义函数为不应连续从设备收集的每个信号定义获取驱动的方法、条件和精确操作。对于由信号获取机制管理的信号，触发类型和本地存储或云端上传的条件仍受两者控制 `CollectionScheme`，两者均受支持，这 `timeBasedCollectionScheme` 与 `conditionBasedCollectionScheme` 常规信号相同。

以下主题向您展示了如何创建和使用 DTCs。

## 主题

- [诊断故障代码关键字](#)
- [为诊断故障代码创建数据收集活动](#)
- [诊断故障代码用例](#)

## 诊断故障代码关键字

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

## signalsToFetch创建广告系列的参数

使用signalsToFetch语法配置如何在 Edge 上获取信号信息。标准信号获取是通过建模来控制的，这些规则是在解码器清单中明确定义的，或者通过边缘优先建模进行自定义定义。有了要获取的信号，您可以定义在活动期间何时以及如何获取数据。

要获取的信号允许收集 DTC 信息。例如，您可以创建一个名为字符串类型的信号，该信号可以DTC\_Info包含每个发动机控制单元 (ECU) 的 DTC 信息。或者，您可以筛选特定的 ECU。

- SignalFetchInformation结构和参数定义。

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

- fullyQualified\_name: 您要使用自定义提取的信号的完全限定名称 (FQDN)。
- signalFetchConfig: 定义了如何获取上述定义信号的规则。它支持基于时间和基于条件的获取。
- conditionLanguageVersion: 用于解析配置中表达式的条件语言版本。
- actions : 在 Edge 上评估的所有操作表达式的列表。Edge 将获得定义信号的值。

### Important

操作只能使用custom\_function。

## 广告系列表达式关键词

以下表达式采用车辆支持的信号的完全限定名称，如果该信号在边缘的信号缓冲区中没有任何数据，则返回 true。另一方面，它会返回 false。

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

## Example使用情况

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

We want to make sure DTC\_Info signal is being generated on edge.

此表达式采用以下输入：

函数名称：字符串

Edge 支持的自定义函数的名称

参数：可变参数 **Expression**

的参数functionName。这可以是任何表达式列表。

参数支持文字类型：字符串、整数、布尔值或双精度。

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

## Example使用情况

```
{
  "fullyQualifiedNames":["Vehicle.ECU1.DTC_INFO"],
  "signalFetchConfig":{
    "timeBased":{
      "executionFrequencyMs":2000
    }
  },
  "actions":"custom_function("DTC_QUERY", -1, 2, -1)"
}
```

## 为诊断故障代码创建数据收集活动

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

本主题介绍如何为诊断故障代码 (DTC) 创建数据收集活动。

1. 在 Edge 上定义自定义信号。您需要将 Edge 上的 DTC 信号的解码规则定义为自定义解码信号。有关更多信息，请参阅 [教程：使用自定义解码接口配置与网络无关的数据收集](#)。
2. 在 Edge 上定义自定义函数。您需要定义一个自定义函数，用于在编译时在 Edge 上收集 DTC 信号。

有关更多信息，请参阅《边缘代理开发人员指南》中的[自定义函数指南](#)和[DTC 数据收集参考实现](#)。

**Note**

示例自定义函数 DTC\_QUERY 如[演示脚本所示](#)。

3. 创建信号目录，将 DTC 信号建模为字符串类型。

```
[
  {
    "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. 创建并激活添加了 DTC 信号的车辆模型。

5. 创建并激活添加了 DTC 信号的解码器清单。DTC 信号应具有 CUSTOM\_DECODING\_INTERFACE 网络接口类型的 CUSTOM\_DECODING\_SIGNAL 信号解码器类型。

#### Example 信号解码器

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

#### Example 网络接口

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

#### Note

控制器局域网 (CAN) 信号不支持字符串数据类型。

6. 配置和制造车辆。车辆必须使用在前面步骤中激活的车辆模型 (车型清单) 和解码器清单。
7. 创建并批准该活动。您需要通过定义 DTC 信号 (可选择使用遥测信号) 来创建战役, 然后将其部署到车辆上。
8. 访问定义的目的地中的数据。DTC 数据将 DTCCodeDTCSnapshot、和 DTCExtendedDatastrings 作为原始字符串包含在活动中定义的数据目标中。

## 诊断故障代码用例

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下用例假设该DTC\_QUERY函数是在[演示脚本](#)中定义的。

### 定期获取

按配置的时间间隔获取 DTC 集合。

以下示例是一个为所有人定期获取信号的Vehicle.DTC\_INFO广告系列，所有人 DTCs 都有状态掩码。ECUs收集数据是有条件的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"
      }
  }
]
}

```

## 条件驱动地获取

满足条件时获取 DTC 集合。例如，当 CAN 信号为时 `Vehicle.Ignition == 1`，获取并上传 DTC 数据。

以下示例广告系列具有条件驱动的信号提取，`Vehicle.ECU1.DTC_INFO`用于检查 DTC (") 是否处于待处理状态，ECU-1 的 `RecordNumber 1 AAA123` 为 `RecordNumber 1`。此活动采用基于时间的数据收集和上传。

```

{
  "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"
    }
  }
]
}
```

## 按需获取

为舰队获取特定 DTC。

对于按需使用案例，您可以使用定期提取中定义的不同活动。通过使用 AWS 物联网 FleetWise 控制台，在部署活动后不久暂停活动或运行以下 CLI 命令来实现按需效果。

- *command-name* 替换为命令名。

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

然后，在 DTC 数据到来后暂停活动。

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

您可以再次恢复活动以获取 DTC 数据。

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

## 可视化 AWS 物联网 FleetWise 车辆数据

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

AWS 物联网边缘代理 FleetWise 软件将选定的车辆数据发送到 MQTT 主题，或者将其传输到 Amazon Timestream 或亚马逊简单存储服务 (Amazon S3) Service。数据到达数据目的地后，您可以使用其他 AWS 服务对其进行处理、重新路由、可视化和共享。

### Note

Amazon Timestream 不在亚太地区 (孟买) 区域推出。

## 处理发送到 MQTT 主题的车辆数据

由 MQTT 消息发送的车辆数据以近乎实时的方式传送，允许您使用规则采取行动，或将数据路由到其他目的地。有关使用 MQTT 的更多信息，请参阅《AWS IoT Core 开发人员指南》AWS IoT 中的 [设备通信协议和规则](#)。

在 MQTT 消息中发送的默认数据架构包含以下字段。

字段名称	数据类型	说明
eventId	varchar	数据收集事件的 ID。
vehicleName	varchar	从中收集数据的车辆的 ID。
name	varchar	边缘代理软件用来收集数据的活动的名称。
time	timestamp	数据点的时间戳。
measure_name	varchar	信号的名称。
measure_value::bigint	bigint	整数类型的信号值。
measure_value::double	double	双精度类型的信号值。
measure_value::boolean	布尔值	布尔值类型的信号值。
measure_value::varchar	varchar	varchar 类型的信号值。

## 在 Timestream 中处理车辆数据

Timestream 是一个完全托管的时间序列数据库，每天可以存储和分析数万亿的时间序列数据点。您的数据存储在客户管理的 Timestream 表中。您可以使用 Timestream 查询车辆数据，以便深入了解您的车辆。有关更多信息，请参阅[什么是 Amazon Timestream ?](#)

传输到 Timestream 的数据的默认架构包含以下字段。

字段名称	数据类型	说明
eventId	varchar	数据收集事件的 ID。
vehicleName	varchar	从中收集数据的车辆的 ID。
name	varchar	边缘代理软件用来收集数据的活动的名称。
time	timestamp	数据点的时间戳。
measure_name	varchar	信号的名称。
measure_value::bigint	bigint	整数类型的信号值。
measure_value::double	double	双精度类型的信号值。
measure_value::boolean	布尔值	布尔值类型的信号值。
measure_value::varchar	varchar	varchar 类型的信号值。

## 可视化存储在 Timestream 中的车辆数据

将您的车辆数据传输到 Timestream 后，您可以使用以下 AWS 服务来可视化、监控、分析和共享您的数据。

- 使用 [Grafana 或 Amazon Managed Grafana](#) 可视化和监控控制面板中的数据。您可以使用单个 Grafana 控制面板可视化来自多个 AWS 来源（例如 Amazon CloudWatch 和 Timestream）以及其他数据源的数据。
- 使用 [Quick](#) 在仪表板中分析和可视化数据。

## 在 Amazon S3 中处理车辆数据

Amazon S3 是一种对象存储服务，可存储和保护任意数量的数据。您可以将 S3 用于各种用例，例如数据湖、备份和恢复、存档、企业应用程序、AWS IoT 设备和大数据分析。您的数据作为存储桶中的对象存储在 S3 中。有关更多信息，请参阅[什么是 Amazon S3？](#)

传输到 Amazon S3 的默认数据架构包含以下字段。

字段名称	数据类型	说明
eventId	varchar	数据收集事件的 ID。
vehicleName	varchar	从中收集数据的车辆的 ID。
name	varchar	边缘代理软件用来收集数据的活动的名称。
time	timestamp	数据点的时间戳。
measure_name	varchar	信号的名称。
measure_value_BIGINT	bigint	整数类型的信号值。
measure_value_DOUBLE	double	双精度类型的信号值。
measure_value_BOOLEAN	布尔值	布尔值类型的信号值。
measure_value_STRUCT	struct	Struct 类型的信号值。
measure_value_VARCHAR	varchar	varchar 类型的信号值。

## 亚马逊 S3 对象格式

AWS 物联网将车辆数据 FleetWise 传输到 S3，然后将其另存为对象。您可以使用唯一标识这些数据的对象 URI 来查找活动数据。S3 对象 URI 格式取决于收集的数据是非结构化数据还是经过处理的数据。

### 非结构化数据

非结构化数据以非预定义的方式存储在 S3 中。它可以是各种格式，例如图像或视频。

传递给 AWS 物联网 FleetWise 的车辆消息以及来自 Amazon Ion 文件的信号数据会被解码并作为对象传输到 S3。S3 对象代表每个信号，并采用二进制编码形式。

非结构化数据 S3 对象 URI 采用以下格式：

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

### 处理的数据

处理的数据存储在 S3 中，并经过验证、丰富和转换消息这些处理步骤。对象列表和速度是处理的数据示例。

传输到 S3 的数据存储为对象，这些对象表示缓冲了大约 10 分钟的记录。默认情况下，在将对象写入 S3 之前，AWS IoT FleetWise 会以该格式添加 UTC 时间 `year=YYYY/month=MM/date=DD/hour=HH` 前缀。此前缀在存储桶中创建一个逻辑层级结构，其中，每个正斜杠 (/) 均在该层级结构中创建一个层级。处理的数据还包含指向非结构化数据的 S3 对象 URI。

处理的数据 S3 对象 URI 采用以下格式：

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

### 原始数据

原始数据，也称为主要数据，是从 Amazon Ion 文件中收集的数据。您可以使用原始数据来解决任何问题或找出错误的根本原因。

原始数据 S3 对象 URI 采用以下格式：

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

## 分析存储在 Amazon S3 中的车辆数据

将车辆数据传输到 S3 后，您可以使用以下 AWS 服务来监控、分析和共享您的数据。

使用 Amazon SageMaker 提取和分析数据，用于下游标签和机器学习 (ML) 工作流程。

有关更多信息，请参阅 Amazon SageMaker 开发者指南中的以下主题：

- [处理数据](#)
- [训练机器学习模型](#)
- [标注图像](#)

使用 Amazon Athena 对您的数据进行分类 AWS Glue 爬网程序 和分析。默认情况下，写入 S3 的对象具有 Apache Hive 样式的时间分区，其数据路径包含以等号连接的键值对。

有关更多信息，请参阅《Amazon Athena 用户指南》中的以下主题：

- [在 Athena 中对数据进行分区](#)
- [使用连接 AWS Glue 到 Amazon S3 中的数据源](#)
- [将 Athena 搭配使用时的最佳实践 AWS Glue](#)

使用 Quick 直接读取 Athena 表或 S3 存储桶，实现数据可视化。

### Tip

如果您直接从 S3 读取，请确认您的车辆数据为 JSON 格式，因为 Quick 不支持 Apache Parquet 格式。

有关更多信息，请参阅《Amazon QuickSight 用户指南》中的以下主题：

- [支持的数据源](#)
- [创建数据源](#)

# 命令

## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

本文档介绍如何使用 [AWS 物联网的命令功能 FleetWise](#)。有关在中使用命令功能的信息 AWS IoT Device Management，请参阅 [命令](#)。

您全权负责以安全且符合适用法律的方式部署命令。有关您责任的更多信息，请参阅 [AWS IoT 服务的AWS 服务条款](#)。

使用命令功能从云端对车辆执行命令。命令一次只能针对一台设备，可用于低延迟、高吞吐量的应用程序，例如取回设备端日志，或启动设备状态更改。

该命令是由管理的资源 AWS IoT Device Management。它包含可重复使用的配置，这些配置在向车辆发送命令执行时适用。您可以为特定用例预定义一组命令，也可以使用它们为重复使用案例创建可重复使用的配置。例如，您可以配置可供应用程序用来锁定车辆门或远程更改温度的命令。

使用 AWS IoT 命令功能，您可以：

- 创建命令资源并重复使用该配置向目标设备发送多个命令，然后在设备上执行这些命令。
- 控制您希望在设备上执行每条命令的粒度。例如，您可以将车辆配置为一 AWS IoT 件东西，然后发送命令来锁定或解锁车辆的车门。
- 在目标设备上同时运行多个命令，无需等待前一个命令完成。
- 选择启用命令事件通知，并在设备运行命令时和命令完成后从设备检索状态和结果信息。

以下主题向您介绍如何创建、发送、接收和管理命令。

## 主题

- [命令概念](#)
- [车辆和指挥系统](#)
- [创建和管理命令](#)
- [启动和监控命令执行](#)
- [示例：使用命令控制车辆转向模式 \(AWS CLI\)](#)

- [命令使用场景](#)

## 命令概念

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

命令是从云端发送到目标设备的指令。目标设备可以是车辆，它必须在事物注册表中注册为AWS IoT 事物。该命令可以包含定义车辆执行器需要执行的动作的参数。然后，车辆解析命令及其参数，并对其进行处理以采取相应的行动。然后，它会向云应用程序提供命令执行的状态。

有关详细的工作流程，请参阅[车辆和指挥系统](#)。

### 主题

- [命令关键概念](#)
- [命令执行状态](#)

## 命令关键概念

下文显示了使用命令功能的一些关键概念以及它如何与上次已知状态 (LKS) 状态模板配合使用。

### 命令

Command 是一个实体，你可以用它向实体车辆发送指令，让它执行诸如开启发动机或更改车窗位置之类的操作。您可以为特定用例预定义一组命令，也可以使用它们为重复使用案例创建可重复使用的配置。例如，您可以配置可供应用程序用来锁定车辆门或远程更改温度的命令。

### 命名空间

使用命令功能时，必须为命令指定命名空间。在 AWS IoT 中创建命令时 FleetWise，必须选择AWS-IoT-FleetWise作为命名空间。使用此命名空间时，必须提供用于在车辆上运行命令的参数。如果要改为在中创建命令，则必须 AWS IoT Device Management 改用AWS-IoT命名空间。有关更多信息，请参阅AWS IoT Device Management 开发者指南中的[命令](#)。

## 命令状态

您创建的命令将处于可用状态，这意味着它可用于在车辆上开始执行命令。如果命令过时，您可以弃用该命令。对于处于已弃用状态的命令，现有的命令执行将运行直至完成。您无法更新命令或运行任何新的执行。要发送新的执行，必须恢复命令以使其可用。

如果不再需要某个命令，也可以将其删除。当您将命令标记为删除时，如果该命令已被弃用的持续时间超过最大超时时间 24 小时，则该命令将被立即删除。如果命令未被弃用，或已被弃用的持续时间短于最大超时时间，该命令将处于待删除状态。该命令将在 24 小时后自动从您的账户中删除。

## 参数

在创建命令时，您可以选择性地指定您希望目标车辆在运行命令时执行的参数。您创建的命令是可重复使用的配置，它可用于向您的车辆发送多个命令执行并同时执行它们。或者，您也可以仅在运行时指定参数，然后选择执行一次性操作，即创建命令并将其发送到您的车辆。

## 目标车辆

当你想运行命令时，必须指定将接收命令并执行特定操作的目标车辆。目标车辆必须已经注册为物品 AWS IoT。将命令发送到车辆后，它将根据您指定的参数和值开始执行命令的实例。

## 执行器

要运行命令时，必须指定车辆上将接收该命令的执行器及其决定要执行的操作的值。您可以选择为执行器配置默认值，以避免发送不准确的命令。例如，您可以对门锁执行器使用默认值，这样该命令就不会意外解锁门。LockDoor有关执行器的一般信息，请参见[重要概念](#)。

## 数据类型支持

用于命令功能的执行器支持以下数据类型。

### Note

远程信息处理数据、命令或上次已知状态 (LKS) 不支持数组。您只能将阵列数据类型用于视觉系统数据。

- 浮点类型。支持以下类型：
  - 浮点数 ( 32 位 )
  - 双精度 ( 64 位 )
- 整数 ( 有符号和无符号 )。支持以下整数类型。

- int8 和 uint8
- int16 和 uint16
- int32 和 uint32
- 长。支持以下长类型。
  - 做多 (int64)
  - 无符号长整型 (uint64)
- 字符串
- 布尔值

## 命令执行

命令执行是在目标设备上运行的命令的实例。车辆使用您在创建命令时或开始执行命令时指定的参数执行命令。然后，车辆执行指定的操作并返回执行状态。

### Note

对于给定的车辆，您可以同时运行多个命令。有关您可以为每辆车运行的最大并发执行次数的信息，请参阅[AWS IoT Device Management 命令配额](#)。

## 上次已知状态 (LKS) 状态模板

状态模板为车主提供了一种跟踪其车辆状态的机制。要近乎实时地监控车辆的最新已知状态 (LKS)，您可以创建状态模板并将其与您的车辆相关联。

使用命令功能，您可以执行可用于状态数据收集和处理的“按需”操作。例如，您可以一次性请求当前的车辆状态（获取），或者激活或停用之前部署的 LKS 状态模板以开始或停止报告车辆数据。有关演示如何在状态模板中使用命令的示例，请参阅[命令使用场景](#)。

## 命令执行状态

开始执行命令后，您的车辆可以发布执行状态，并提供状态原因作为有关执行的附加信息。以下各节描述了各种命令执行状态和状态代码。

### 主题

- [命令执行状态原因代码和描述](#)
- [命令执行状态和状态码](#)

- [命令执行超时状态](#)

## 命令执行状态原因代码和描述

要报告命令执行状态的更新，您的车辆可以使用AWS IoT Core 开发者指南中描述的[命令保留主题](#)，使用UpdateCommandExecution该 API 将更新的状态信息发布到云端。在报告状态信息时，您的设备可以提供更多背景信息，说明使用对象执行的每个命令的状态，以及StatusReason对象中reasonDescription包含的字段reasonCode和字段。

## 命令执行状态和状态码

下表显示了各种命令执行状态代码以及命令执行可以过渡到的允许状态。它还显示命令执行是否“终端”（也就是说，不会有进一步的状态更新），变更是由车辆还是云端发起的，以及不同的预定义状态代码以及它们如何映射到云端报告的状态。

- 有关如何 AWS IoT FleetWise 使用预定义的状态代码和statusReason对象的信息，请参阅适用于 AWS IoT 的 Edge Agent FleetWise 软件文档中的[命令状态](#)。
- 有关终端和非终端执行以及状态间转换的更多信息，请参阅AWS IoT Core 开发者指南中的[命令执行状态](#)。

## 命令执行状态和来源

命令执行状态	说明	由设备/云端发起？	是否为终端执行？	允许的状态转换	预定义的状态码
CREATED	当开始执行命令 (API) 的 StartCommandExecution API 请求成功时，命令执行状态将更改为CREATED。	云	否	<ul style="list-style-type: none"> <li>• IN_PROGRESS</li> <li>• SUCCEEDED</li> <li>• FAILED</li> <li>• REJECTED</li> <li>• TIMED_OUT</li> </ul>	无
IN_PROGRESS	当车辆开始执行命令时，它可以向响	设备	否	<ul style="list-style-type: none"> <li>• 进行中</li> <li>• SUCCEEDED</li> </ul>	COMMAND_STATUS_COMMAND

命令执行状态	说明	由设备/云端发起？	是否为终端执行？	允许的状态转换	预定义的状态码
	应主题发布一条消息，将状态更新为IN_PROGRESS。			<ul style="list-style-type: none"> <li>FAILED</li> <li>REJECTED</li> <li>超时</li> </ul>	MAND_IN_PROGRESS
SUCCEEDED	当车辆成功处理命令并完成执行后，它可以向响应主题发布一条消息以更新状态SUCCEEDED。	设备	是	不适用	COMMAND_STATUS_SUCCEEDED
FAILED	当车辆未能执行命令时，它可以向响应主题发布一条消息，将状态更新为FAILED。	设备	是	不适用	COMMAND_EXECUTION_FAILED
REJECTED	如果车辆未能接受命令，它可以向响应主题发布一条消息，将状态更新为REJECTED。	设备	是	不适用	无

命令执行状态	说明	由设备/云端发起？	是否为终端执行？	允许的状态转换	预定义的状态码
TIMED_OUT	<p>TIMED_OUT</p> <p>由于以下任何原因，命令执行状态可能会更改为。</p> <ul style="list-style-type: none"> <li>未收到命令执行的结果，云端会自动报告TIMED_OUT状态。</li> <li>车辆报告说，它在尝试执行命令时出现了超时。在这种情况下，命令执行变为终止。</li> </ul> <p>有关此状态的更多信息，请参阅<a href="#">命令执行超时状态</a>。</p>	设备和云	否	<ul style="list-style-type: none"> <li>SUCCEEDED</li> <li>FAILED</li> <li>REJECTED</li> <li>超时</li> </ul>	COMMAND_STATUS_EXECUTION_TIMEOUT

## 命令执行超时状态

云端和设备都可以报告命令执行超时。在命令发送到设备后，计时器启动。如果在指定的持续时间内没有收到来自设备的响应，云端会报告TIMED\_OUT状态。在这种情况下，TIMED\_OUT状态下的命令执行为非终止。

设备可以将此状态改写为终端状态SUCCEEDED，例如FAILED、或REJECTED。它还可以报告运行命令时出现超时。在这种情况下，命令执行状态保持为，TIMED\_OUT但StatusReason对象的字段会根据设备报告的信息进行更新。TIMED\_OUT状态下的命令执行现在变为终止。

有关更多信息，请参阅AWS IoT Core 开发者指南中的[命令执行超时注意事项](#)。

## 车辆和指挥系统

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您全权负责以安全且符合适用法律的方式部署命令。

要使用命令功能：

1. 首先，创建一个命令资源。（可选）指定包含执行命令所需信息的参数。
2. 指定将接收命令并执行指定操作的目标车辆。
3. 现在，您可以在目标设备上运行该命令，检查命令执行详细信息以检索状态，并使用 CloudWatch 日志进一步解决任何问题。

以下各节向您展示了车辆和命令之间的工作流程。

主题

- [工作流程概述](#)
- [车辆工作流程](#)
- [命令工作流程](#)
- [\( 可选 \) 命令通知](#)

## 工作流程概述

以下步骤概述了车辆和命令之间的命令工作流程。当您使用任何命令 HTTP API 操作时，请求使用 Sigv4 凭证进行签名。

**Note**

除了 StartCommandExecution API 操作之外，所有通过 HTTP 协议执行的操作都使用控制平面端点。

## 1. 建立 MQTT 连接并订阅命令主题

要为命令工作流程做准备，设备必须与 `iot:Data-ATS` 终端节点建立 MQTT 连接，并订阅上述命令请求主题。或者，您的设备也可以订阅已接受和拒绝的命令响应主题。

## 2. 创建车辆模型和命令资源

现在，您可以使用和 `CreateCommand` 控制平面 API 操作创建载具 `CreateVehicle` 和命令资源。命令资源包含在车辆上执行命令时要应用的配置。

## 3. 在目标设备上启动命令执行

使用带有账户特定 `iot:Jobs` 端点 `StartCommandExecution` 的数据平面 API 在车辆上开始执行命令。API 会向命令请求主题发布一条 `protobuf` 编码的有效负载消息。

## 4. 更新命令执行的结果

车辆处理收到的命令和有效载荷，然后使用 `UpdateCommandExecution` API 将命令执行结果发布到响应主题。如果您的车辆订阅了命令已接受和拒绝的响应主题，它将收到一条消息，指示云服务接受还是拒绝了响应。

## 5. ( 可选 ) 检索命令执行结果

要检索命令执行的结果，您可以使用 `GetCommandExecution` 控制平面 API 操作。在您的车辆向响应主题发布命令执行结果后，此 API 将返回更新的信息。

## 6. ( 可选 ) 订阅和管理命令事件

要接收命令执行状态更新的通知，您可以订阅命令事件主题。然后，您可以使用 `CreateTopicRule` 控制平面 API 将命令事件数据路由到其他应用程序，例如 AWS Lambda 函数或 Amazon SQS，并在此基础上构建应用程序。

# 车辆工作流程

以下步骤详细描述了使用命令功能时的车辆工作流程。

**Note**

本节中描述的操作使用 MQTT 协议。

## 1. 建立 MQTT 连接

要让您的车辆做好使用命令功能的准备，它必须先连接到 AWS IoT Core 消息代理。必须允许您的车辆执行 `iot:Connect` 操作才能连接到消息代理 AWS IoT Core 并与消息代理建立 MQTT 连接。要查找您的数据平面终端节点 AWS 账户，请使用 `DescribeEndpoint` API 或 `describe-endpoint` CLI 命令，如下所示。

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

运行此命令将返回账户特定的数据面板端点，如下所示。

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

## 2. 订阅命令请求主题

建立连接后，您的设备即可订阅 AWS IoT 命令 MQTT 请求主题。当您创建命令并在目标设备上开始执行命令时，消息代理将向请求主题发布一条 protobuf 编码的有效载荷消息。然后，您的设备可以接收有效载荷消息并处理命令。在此示例中，`<DeviceID>` 替换为目标车辆的唯一标识符。此 ID 可以是您车辆的唯一标识符或事物名称

**Note**

发送到设备的有效载荷消息必须使用 protobuf 格式。

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

## 3. ( 可选 ) 订阅命令响应主题

或者，您可以订阅这些命令响应主题，以接收一条消息，指示云服务是接受还是拒绝了来自设备的响应。

**Note**

您的车辆可以选择订阅/accepted和/rejected回复主题。即使您的车辆没有明确订阅这些主题，它们也会自动收到这些回复消息。

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

#### 4. 更新命令执行的结果

然后，目标车辆会处理该命令。然后，它使用 UpdateCommandExecution API 将执行结果发布到以下 MQTT 响应主题。

**Note**

对于给定的车辆和命令执行，<DeviceID>必须与设备订阅的请求主题中的相应字段相匹配。

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

该 UpdateCommandExecution API 是基于 MQTT 的数据平面 API 操作，已通过 TLS 进行身份验证。

- 如果云服务成功处理了命令执行结果，则会向 MQTT 接受的主题发布一条消息。接受的主题使用以下格式。

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

- 如果云服务未能处理命令执行结果，则会向 MQTT 被拒绝的主题发布响应。被拒绝的主题使用以下格式。

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

有关此 API 的更多信息和示例，请参阅[更新命令执行结果](#)。

## 命令工作流程

以下步骤详细描述了命令的工作流程。

### Note

本节中描述的操作使用 HTTP 协议。

### 1. 注册您的车辆

现在，您已经准备好使用命令功能的车辆，您可以通过注册车辆然后创建将发送给车辆的命令来准备应用程序。要注册车辆，请使用[CreateVehicle](#)控制平面 API 操作创建车辆模型（模型清单）的实例。有关更多信息和示例，请参阅[创建车辆](#)。

### 2. 创建命令

使用 [CreateCommand](#) HTTP 控制平面 API 操作对适用于您目标车辆的命令进行建模。指定执行命令时要使用的任何参数和默认值，并确保它使用 AWS-IoT-FleetWise 命名空间。有关使用此 API 的更多信息和示例，请参阅[创建命令资源](#)。

### 3. 开始执行命令

现在，您可以使用[StartCommandExecution](#)数据平面 API 操作在车辆上执行您创建的命令。AWS IoT Device Management 获取命令和命令参数，并验证传入的请求。然后，它使用所需参数调 AWS IoT FleetWise 用 API 来生成车辆特定的有效载荷。然后，AWS IoT Device Management 通过 MQTT 将有效负载发送到您的设备订阅的命令请求主题。有关使用此 API 的更多信息和示例，请参阅[发送命令 \(AWS CLI\)](#)。

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

### Note

如果从云端发送命令时设备处于离线状态，并且正在使用 MQTT 持久会话，则该命令将在消息代理处等待。如果设备在超时持续时间之前恢复联机，并且已订阅命令请求主题，

则设备可以处理命令并将结果发布到响应主题。如果设备在超时持续时间之前没有恢复联机，则命令执行将超时，有效载荷消息将过期。

#### 4. 检索命令执行情况

在设备上执行命令后，使用 [GetCommandExecution](#) 控制平面 API 操作来检索和监控命令执行的结果。您还可以使用 API 获取有关执行数据的其他信息，例如上次更新时间、执行完成时间以及指定的参数。

##### Note

要检索最新的状态信息，设备必须已将命令执行结果发布到响应主题。

有关使用此 API 的更多信息和示例，请参阅 [获取命令执行](#)。

## ( 可选 ) 命令通知

您可以订阅命令事件，以便在命令执行状态更改时接收通知。以下步骤显示如何订阅命令事件，然后处理它们。

#### 1. 创建主题规则

您可以订阅命令事件主题，并在命令执行状态更改时接收通知。您还可以创建主题规则，将车辆处理的数据路由到其他应用程序，例如 AWS Lambda 函数。您可以使用控制台或 AWS IoT [CreateTopicRule](#) AWS IoT Core 控制平面 API 操作创建主题规则。有关更多信息，请参阅 [创建和 AWS IoT 规则](#)。

在此示例中，将 *<CommandID>* 替换为您要接收通知的命令的标识符，并将 *<CommandExecutionStatus>* 替换为命令执行的状态。

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

##### Note

要接收所有命令和命令执行状态的通知，您可以使用通配符并订阅以下主题。

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

## 2. 接收和处理命令事件

如果您在上一步中创建了订阅命令事件的主题规则，则可以管理收到的命令推送通知。您也可以选择使用您创建的主题规则在其上构建应用程序，例如使用 Amazon SQS、Amazon SNS 或 Step Functions。AWS Lambda

以下代码显示了您将收到的命令事件通知的示例有效载荷。

```
{
  "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
}
```

## 创建和管理命令

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以配置可重复使用的远程操作或向您的设备发送一次性即时指令。使用此功能时，您可以指定您的设备可以近乎实时地执行的指令。命令允许您为目标车辆配置可重复使用的远程操作。创建命令后，您可以开始执行针对特定车辆的命令。

本主题介绍如何使用 AWS IoT Core API 或创建和管理命令资源 AWS CLI。它向您展示了如何对命令资源执行以下操作。

## 主题

- [创建命令资源](#)
- [检索关于命令的信息](#)
- [在你的账户中列出命令](#)
- [更新或弃用命令资源](#)
- [删除命令资源](#)

## 创建命令资源

您可以使用 [CreateCommand](#) AWS IoT Core 控制平面 API 操作或 AWS 物联网 FleetWise 控制台来创建命令。

### 创建命令 (控制台)

您可以使用 AWS IoT FleetWise 控制台创建命令。

### 创建命令

1. 打开 [AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格上，选择命令。
3. 选择创建命令。
4. 指定唯一的命令 ID 以帮助您识别要在车辆上运行的命令。
5. (可选) 指定可选的显示名称和描述。
6. (可选) 选择执行器和默认参数值。参数指定目标车辆在收到命令后可以执行的操作。如果您不添加参数，则需要在运行命令时提供这些参数。
7. 选择一个 IAM 角色来授予为命令生成有效负载的权限。请参阅 [控制访问权限](#)。
8. 选择创建命令。

### 创建命令 (AWS CLI)

以下示例说明如何使用参数创建命令。

## 创建命令时的注意事项

当您在以下位置创建命令时 AWS IoT FleetWise :

- 您必须指定授 `roleArn` 予在车辆上创建和运行命令的权限。有关更多信息以及示例策略 ( 包括何时启用 KMS 密钥 ) , 请参阅 [授 AWS IoT Device Management 予使用以下命令生成有效负载的权限 AWS IoT FleetWise](#)。
- 必须指定 `AWS-IoT-FleetWise` 为命名空间。
- 您可以跳过该 `mandatory-parameters` 字段, 改为在运行时指定它们。或者, 您可以创建带有参数的命令, 也可以选择为它们指定默认值。如果您指定了默认值, 则在运行时, 您可以使用这些值或通过指定自己的值来覆盖它们。有关这些其他示例, 请参见 [命令使用场景](#)。
- 您最多可以为该字段指定三个名称/值对。 `mandatory-parameters` 但是, 在车辆上执行命令时, 只接受一个名称/值对, 并且该 `name` 字段必须使用带有前缀的完全限定名称。 `$actuatorPath`。
- `command-id` 替换为命令的唯一标识符。您可以使用 UUID、字母数字字符、“-”和“\_”。
- `role-arn` 替换为授予您创建和运行命令权限的 IAM 角色, 例如 `"arn:aws:iam:accountId:role/FwCommandExecutionRole"`。
- ( 可选 ) `display-name` 替换为用户友好的命令 `description` 名称以及对命令的有意义的描述。
- 将 `mandatory-parameters` 对象 `namevalue` 的 `and` 替换为正在创建的命令所需的信息。该 `name` 字段是信号目录中定义的完全限定名称, 前缀 `$actuatorPath`。为前缀。例如, `name` 可以是也可以是 `$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode` 表示转向模式状态的布尔值, 例如 `{"B": false}`。 `value`

```
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  
    }  
  ]'
```

CreateCommandAPI 操作返回的响应包含命令的 ID 和 ARN ( 亚马逊资源名称 ) 。

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

## 检索关于命令的信息

您可以使用[GetCommand](#) AWS IoT Core 控制平面 API 操作来检索有关命令资源的信息。

要获取有关命令资源的信息，请运行以下命令。*command-id*替换为创建命令时使用的标识符。

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

GetCommandAPI 操作返回的响应包含以下信息。

- 命令的 ID 和 ARN ( 亚马逊资源名称 )。
- 命令的创建和上次更新的日期和时间。
- 指示是否可以在车辆上运行的命令状态。
- 您在创建命令时指定的任何参数。

```
{
  "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
}
```

## 在你的账户中列出命令

您可以使用 [ListCommands](#) AWS IoT Core 控制平面 API 操作列出您在账户中创建的所有命令。

要列出您账户中的命令，请运行以下命令。默认情况下，API 会返回为两个命名空间创建的命令。要筛选列表以仅显示为其创建的命令 AWS IoT FleetWise，请运行以下命令。

### Note

您也可以按升序或降序对列表进行排序，或者筛选列表以仅显示具有特定命令参数名称的命令。

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

ListCommandsAPI 操作返回的响应包含以下信息。

- 命令的 ID 和 ARN ( 亚马逊资源名称 ) 。
- 命令的创建和上次更新的日期和时间。
- 指示命令是否可在车辆上运行的命令状态。

## 更新或弃用命令资源

您可以使用 [UpdateCommand](#) AWS IoT Core 控制平面 API 操作或 [AWS IoT FleetWise 控制台](#) 来更新命令资源。您可以更新命令的显示名称和描述。如果当前未使用某个命令，也可以将其弃用。

### Note

在执行命令时，您无法修改命名空间信息或要使用的参数。

### 更新命令 ( 控制台 )

#### 更新命令

要从控制台更新命令，请转到 [AWS 物联网 FleetWise 控制台的命令](#) 页面并执行以下步骤。

1. 选择要更新的命令，然后选择“编辑”。
2. 编辑命令详细信息，然后选择“保存更改”。

## 弃用命令

要弃用控制台中的命令，请转到 AWS 物联网 FleetWise 控制台的[命令](#)页面并执行以下步骤。

1. 选择要弃用的命令，然后选择“弃用”。
2. 确认弃用，然后选择“弃用”。

## 更新命令 (AWS CLI)

### 更新命令

要更新命令资源，请运行以下命令。*command-id*替换为要更新的命令的标识符，并提供更新的*display-name*和*description*。

```
aws iot update-command \  
  --command-id command-id \  
  --display-name display-name \  
  --description description
```

UpdateCommandAPI 操作返回以下响应。

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

### 弃用命令

如果您打算不再在设备上继续使用某个命令，或者该命令已过时，则可以弃用该命令。以下示例显示了如何弃用命令。

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

UpdateCommandAPI 操作返回的响应包含命令的 ID 和 ARN (亚马逊资源名称)。

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

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

命令被弃用后，现有的命令执行将继续在车辆上运行，直到它们变为终端。要运行任何新的命令执行，必须使用 UpdateCommand API 恢复命令以使其可用。有关弃用和恢复命令的更多信息及其注意事项，请参阅《开发人员指南》AWS IoT Core 中的[弃用命令资源](#)。

## 删除命令资源

您可以使用 [DeleteCommand](#) AWS IoT Core 控制平面 API 操作或 AWS 物联网 FleetWise 控制台来删除命令资源。

### Note

删除是永久性操作，无法撤消。该命令将从您的账户中永久删除。

### 删除命令 (控制台)

要从控制台删除 [命令](#)，请转到 [AWS 物联网 FleetWise 控制台的命令](#) 页面并执行以下步骤。

1. 选择要删除的命令，然后选择“删除”。
2. 确认要删除该命令，然后选择“删除”。

### 删除命令 (AWS CLI)

要删除命令资源，请运行以下命令。*command-id* 替换为要删除的命令的标识符。以下示例说明如何删除命令资源。

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

如果删除请求成功：

- 如果该命令的弃用时间超过了最大超时时间 24 小时，则该命令将被立即删除，并且您将看到 HTTP statusCode 为 204。
- 如果该命令未被弃用，或者已被弃用的时间短于最大超时时间，则该命令将处于 pending deletion 状态，您将看到 HTTP 为 202 statusCode。该命令将在最长 24 小时超时后自动从您的账户中删除。

# 启动和监控命令执行

## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

创建命令资源后，您可以开始在目标车辆上执行命令。一旦车辆开始执行命令，它就可以开始更新命令执行的结果，并将状态更新和结果信息发布到 MQTT 保留主题中。然后，您可以检索命令执行的状态并监控您账户中执行的状态。

本主题介绍如何使用 AWS CLI 或 AWS 物联网 FleetWise 控制台向车辆发送命令。它还向您展示了如何监视和更新命令执行的状态。

## 主题

- [更新命令执行结果](#)
- [获取命令执行](#)
- [列出您账户中的命令执行情况](#)
- [删除命令执行](#)

## 发送命令（控制台）

要从控制台发送命令，请转到 AWS 物联网 FleetWise 控制台的 [车辆](#) 页面并执行以下步骤。

1. 选择要向其发送命令的车辆。
2. 选择 Run command（运行命令）。
3. 选择命令 ID。
4. 指定命令执行超时，然后选择运行命令。

## 发送命令 (AWS CLI)

您可以使用 [StartCommandExecution](#) AWS IoT 数据平面 API 操作向车辆发送命令。然后，车辆将命令转发给汽车中间件服务（例如 SOME/IP（基于 IP 的可扩展面向服务的中间件），或者将其发布到车辆网络（例如控制器局域网 (CAN) 设备接口）上。下面的示例使用了 AWS CLI。

## 主题

- [发送命令时的注意事项](#)
- [获取账户特定的数据面板端点](#)
- [发送命令示例](#)

## 发送命令时的注意事项

当您在以下位置开始执行命令时 AWS IoT FleetWise :

- 你必须为车辆准备 AWS IoT 一件东西。有关更多信息，请参阅 [配置 AWS 物联网 FleetWise 车辆](#)。
- 您必须已经创建了以命名空间AWS-IoT-FleetWise为命名空间的命令role-Arn，并提供了授予您在 AWS IoT 中创建和运行命令的权限 FleetWise。有关更多信息，请参阅 [创建命令资源](#)。
- 如果您选择使用在创建命令时为参数指定的任何默认值，则可以跳过该parameters字段。如果在创建时mandatory-parameters未指定，或者您想通过为参数指定自己的值来覆盖任何默认值，则必须指定该parameters字段。有关这些其他示例，请参见[命令使用场景](#)。
- 您最多可以为该字段指定三个名称/值对。mandatory-parameters但是，在车辆上执行命令时，只接受一个名称/值对，并且该name字段必须使用带有前缀的完全限定名称。\$actuatorPath。

## 获取账户特定的数据面板端点

在运行 API 命令之前，您必须获取该终端节点的账户特定终端节点 URL。iot:Jobs例如，如果您运行此命令：

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

它将返回账户特定的端点 URL，如下面的示例响应所示。

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

## 发送命令示例

要向车辆发送命令，请运行以下命令。

- 将要执行的命令替换*command-arn*为 ARN。您可以从 create-command CLI 命令的响应中获取此信息。
- *target-arn*替换为要为其执行命令的目标设备或 AWS IoT 事物的 ARN。

**Note**

您可以指定 AWS IoT 事物 ( 物AWS 联网 FleetWise 车辆 ) 的目标 ARN。目前不支持事物组和队列。

- `endpoint-url` 替换为您在中获得的账户专用端点[获取账户特定的数据面板端点](#)，例如，前缀为 `https://`。 `https://123456789012abcd.jobs.iot.ap-south-1.amazonaws.com`
- 将 `name` 和 `value` 替换为使用 `create-command CLI` 创建命令时指定的 `mandatory-parameters` 字段。

该 `name` 字段是信号目录中定义的完全限定名称，前缀 `$actuatorPath.` 为前缀。例如，`name` 可以是也可以

是 `$actuatorPath.Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringMode` 表示转向模式状态的布尔值，例如 `{"B": false}`。 `value`

- ( 可选 ) 您也可以指定其他参数 `executionTimeoutSeconds`。此可选字段指定设备必须响应执行结果的时间 ( 以秒为单位 )。您可以将超时的最大值配置为 24 小时。

创建命令执行后，计时器启动。在计时器到期之前，如果命令执行状态未更改为使其终止的状态，例如 `SUCCEEDED` 或 `FAILED`，则状态会自动更改为 `TIMED_OUT`。

**Note**

设备还可以报告 `TIMED_OUT` 状态，或者将此状态改写为诸如 `SUCCEEDED`、或 `FAILED`、之类的状态 `REJECTED`，命令执行将变为终止。有关更多信息，请参阅 [命令执行超时状态](#)。

```
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
    }
  ]'
```

StartCommandExecutionAPI 操作返回命令执行 ID。您可以使用此 ID 查询命令执行状态、详细信息和命令执行历史记录。

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

运行该命令后，您的设备将收到包含以下信息的通知。该issued\_timestamp\_ms字段对应于调用 StartCommandExecution API 的时间。对timeout\_ms应于在调用 StartCommandExecution API 时使用executionTimeoutSeconds参数配置的超时值。

```
timeout_ms: 9000000
issued_timestamp_ms: 1723847831317
```

## 更新命令执行结果

要更新命令执行状态，您的设备必须已建立 MQTT 连接并订阅了以下命令请求主题。

在此示例中，*<device-id>*替换为目标设备的唯一标识符（可以是VehicleId或事物的名称），并*<execution-id>*替换为命令执行的标识符。

### Note

- 有效载荷必须使用 protobuf 格式。
- 您的设备可以选择订阅/accepted和/rejected回复主题。即使您的设备没有明确订阅这些回复消息，它们也会收到这些回复消息。

```
// 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
```

您的设备可以向命令响应主题发布消息。处理完命令后，它会向该主题发送一个 protobuf 编码的响应。该*<DeviceID>*字段必须与请求主题中的相应字段匹配。

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

在您的设备发布对此主题响应后，您可以使用 `GetCommandExecution` API 检索更新的状态信息。命令执行的状态可以是此处列出的任何状态。

- IN\_PROGRESS
- SUCCEEDED
- FAILED
- REJECTED
- TIMED\_OUT

请注意，任何状态 `SUCCEEDED`、`FAILED`、和 `REJECTED` 的命令执行都是终止的，并且状态由设备报告。当命令执行处于终止状态时，这意味着不会对其状态或相关字段进行进一步的更新。设备或云端可能会报告 `TIMED_OUT` 状态。如果云端报告了状态，则设备稍后可能会更新状态原因字段。

例如，下面显示了设备发布的 MQTT 消息示例。

#### Note

对于命令执行状态，如果您的设备使用 `statusReason` 对象发布状态信息，则必须确保：

- `reasonCode` 使用该模式 `[A-Z0-9_-]+`，其长度不超过 64 个字符。
- `reasonDescription` 的长度不超过 1,024 个字符。它可以使用除控制字符（如换行符）之外的任何字符。

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

有关展示如何使用 AWS IoT Core MQTT 测试客户端订阅主题和查看命令执行消息的示例，请参阅 [AWS IoT Core 开发者指南中的使用 MQTT 测试客户端查看命令更新](#)。

## 获取命令执行

您可以使用 [GetCommandExecution](#) AWS IoT 控制平面 API 操作来检索有关命令执行的信息。您必须已经使用 `StartCommandExecution` API 操作执行了此命令。

要检索已执行命令的元数据，请运行以下命令。

- `execution-id` 替换为命令的 ID。您可以从 `start-command-execution` CLI 命令的响应中获取此信息。
- `target-arn` 替换为要执行命令的目标车辆或 AWS IoT 事物的 ARN。

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

`GetCommandExecutionAPI` 操作返回一个响应，其中包含有关命令执行的 ARN、执行状态以及命令开始执行的时间和完成时间的信息。以下代码显示 API 请求的示例响应。

为了提供有关每个命令执行状态的更多上下文，命令功能提供了一个 `statusReason` 对象。该对象包含两个字段，`reasonCode` 和 `reasonDescription`。使用这些字段，您的设备可以提供有关命令执行状态的更多信息。此信息将覆盖云端报告的所有默认 `reasonCode` 信息。`reasonDescription`

要报告此信息，您的设备可以将更新的状态信息发布到云端。然后，当您使用 `GetCommandExecution` API 检索命令执行状态时，您将看到最新的状态代码。

### Note

执行响应中的 `completedAt` 字段对应于设备向云端报告终端状态的时间。对于 `TIMED_OUT` 状态，只有当设备报告超时时，才会设置此字段。当 `TIMED_OUT` 状态由云端设置时，`TIMED_OUT` 状态不会更新。有关超时行为的更多信息，请参阅 [命令执行超时状态](#)。

```
{  
  "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 }
  }'
```

## 列出您账户中的命令执行情况

使用[ListCommandExecutions](#) AWS IoT Core 控制平面 HTTP API 操作列出您账户中的所有命令执行情况。该示例使用 AWS CLI。

### 主题

- [列出命令执行时的注意事项](#)
- [列出命令执行示例](#)

## 列出命令执行时的注意事项

以下是使用 ListCommandExecutions API 时的一些注意事项。

- 您必须至少指定targetArn或，commandArn具体取决于您是要列出特定命令还是目标车辆的执行情况。API 请求不能为空，也不能在同一个请求中包含两个字段。
- 您只能提供startedTimeFilter或completedTimeFilter信息。API 请求不能为空，也不能在同一个请求中包含两个字段。您可以使用对象的before和after字段列出在特定时间范围内创建或完成的命令执行。
- before和after字段都不能大于当前时间。默认情况下，如果您未指定任何值，则该before字段为当前时间，after字段为当前时间-6 个月。也就是说，根据您使用的筛选条件，API 将列出过去六个月内创建或完成的所有执行。
- 您可以使用sort-order参数来指定是否要按升序列出执行次数。默认情况下，如果您未指定此字段，则执行将按降序列出。
- 在列出命令 ARN 的命令执行时，您无法根据命令执行的状态筛选命令执行。

## 列出命令执行示例

以下示例显示如何列出您 AWS 账户中的命令执行。

运行命令时，您必须指定是否筛选列表，以仅显示使用 `targetArn` 为特定设备创建的命令执行，或使用 `commandArn` 为特定命令指定的执行。

在此示例中：

- `<target-arn>` 与您要执行命令的目标设备的 Amazon 资源编号 (ARN)，例如 `arn:aws:iot:us-east-1:123456789012:thing/b8e4157c98f332cffb37627f`。
- `<target-arn>` 与您要执行命令的目标设备的 Amazon 资源编号 (ARN)，例如 `arn:aws:iot:us-east-1:123456789012:thing/b8e4157c98f332cffb37627f`。
- `<after>` 与您希望列出在此时间之后创建的执行的时间，例如 `2024-11-01T03:00`。

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

运行此命令会生成一个响应，其中包含您创建的命令执行列表，以及执行开始和完成的时间。它还提供状态信息，以及包含有关状态附加信息的 `statusReason` 对象。

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

## 删除命令执行

如果您不再想使用某个命令执行，可以将其从您的账户中永久删除。

### Note

只有当命令执行进入终端状态（例如 SUCCEEDED、FAILED 或 REJECTED）时，才能将其删除。

以下示例说明如何使用该命令删除 `delete-command-execution` AWS CLI 命令执行。<code><code>*<execution-id>*</code> 替换为要删除的命令执行的标识符。

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

如果 API 请求成功，则该命令执行会生成状态码 200。您可以使用 `GetCommandExecution` API 来验证该命令执行是否已不在您的账户中。

## 示例：使用命令控制车辆转向模式 (AWS CLI)

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下示例向您展示了如何使用命令功能 AWS CLI。此示例使用 AWS IoT FleetWise 车辆作为目标设备来展示如何发送命令来远程控制转向模式。

### 主题

- [车辆转向模式示例概述](#)
- [先决条件](#)

- [用于使用远程命令的 IAM 政策](#)
- [运行 AWS IoT 命令 \(AWS CLI\)](#)
- [清理](#)

## 车辆转向模式示例概述

在这个例子中，你将：

1. 使用来更改车辆转向模式的操作创建命令资源。create-command AWS CLI
2. 使用检索有关命令的信息，例如命令的创建时间或上次更新的时间get-command AWS CLI。
3. 使用以转向模式为必填参数start-command-execution AWS CLI 的向车辆发送命令，然后将在设备上执行该命令。
4. 使用获取命令执行的结果get-command-execution AWS CLI。您可以检查执行何时完成，并检索其他详细信息，例如执行结果和完成执行命令所花费的时间。
5. 通过删除您不想再使用的命令和命令执行来执行清理活动。

## 先决条件

在运行此示例之前：

- 将您的 AWS IoT FleetWise 车辆作为 AWS IoT 注册表中的 AWS IoT 一件事进行配置。您还必须向您的事物添加证书并激活它，并将策略附加到您的事物上。然后，您的设备可以连接到云端并执行命令。有关更多信息，请参阅[预配车辆](#)。
- 创建一个 IAM 用户和一个 IAM 策略，该策略授予您使用命令执行 API 操作的权限，如所示[用于使用远程命令的 IAM 政策](#)。

## 用于使用远程命令的 IAM 政策

下表显示了一个示例 IAM 策略，该策略授予对命令功能的所有控制平面和数据平面 API 操作的访问权限。该应用程序的用户将有权执行所有远程命令 API 操作，如表所示。

### API 操作

API 操作	控制/数据平面	协议	说明	资源
CreateCommand	控制面板	HTTP	创建命令资源	• 命令

API 操作	控制/数据平面	协议	说明	资源
GetCommand	控制面板	HTTP	检索命令的相关信息	• 命令
UpdateCommand	控制面板	HTTP	更新有关命令的信息或将其弃用	• 命令
ListCommands	控制面板	HTTP	列出您账户中的命令	• 命令
DeleteCommand	控制面板	HTTP	删除命令	• 命令
StartCommandExecution	数据面板	HTTP	开始执行命令	• 命令 • thing
UpdateCommandExecution	数据面板	MQTT	更新命令执行	• 命令 • thing
GetCommandExecution	控制面板	HTTP	检索有关命令执行的信息	• 命令 • thing
ListCommandExecutions	控制面板	HTTP	列出您账户中的命令执行情况	• 命令 • thing
DeleteCommandExecution	控制面板	HTTP	删除命令执行	• 命令 • thing

在此示例中：

- *us-east-1* 和你的 AWS 区域，比如 *ap-south-1*。
- 将 *111122223333* 替换成您的 AWS 账户编号，例如 *57EXAMPLE833*。
- *command-id* *command-id1*、*command-id2* 使用您的唯一命令标识符，例如 *LockDoor* 或 *TurnOffAC*。
- *thing-name* 用你的 AWS IoT 事物名称，比如 *my\_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"
      ]
    }
  ]
}
```

## 运行 AWS IoT 命令 (AWS CLI)

下文显示了如何使用 AWS CLI 来执行命令操作和更改车辆转向模式。

### 1. 为转向模式操作创建命令资源

使用 CLI 创建要发送到设备的 `create-command` 命令。在此示例中，指定：

- `command-id` 用作 *TurnOffSteeringMode*
- `role-arn` "arn:aws:iam:accountId:role/FwCommandExecutionRole" `s role-arn` 必须提供，因为授予在车辆上创建和运行命令的权限是 IAM 角色。有关更多信息，请参阅 [授 AWS IoT Device Management 予使用以下命令生成有效负载的权限 AWS IoT FleetWise](#)。
- `display-name` 作为 *Turn off steering mode*
- `namespace` 必须是 `AWS-IoT-FleetWise`
- `mandatory-parameters` 作为名称-值对，以 "" 和 "defaultValue" 为键，以 *\$actuatorPath.Vehicle.Chassis.SteeringWheel.TurnOffSteeringMode* 为名称，以 `{ "S": "true" }` 为值。

#### Note

您也可以在不指定任何必需参数的情况下创建命令。然后，您必须指定使用 `start-command-execution` CLI 执行命令时要使用的参数。有关示例，请参阅 [命令使用场景](#)。

#### Important

使用 `AWS-IoT-FleetWise` 命名空间时，必须确保作为命名空间一部分指定的 `Name` 字段 `mandatory-parameters` 使用 `$actuatorPath` 前缀，并且该 `Value` 字段必须使用字符串数据类型。

```
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" }
}
```

以下输出显示了 CLI 的示例响应，其中 ap-south-1 和 123456789012 是 AWS 区域和 AWS 账户 ID 的示例。

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

有关使用此命令的其他示例，请参见 [创建命令资源](#)。

## 2. 检索有关命令的信息

运行以下命令以检索有关该命令的信息，其中 `command-id` 是上面 `create-command` 操作输出中的命令 ID。

### Note

如果您创建了多个命令，则可以使用 `ListCommands` API 列出账户中的所有命令，然后使用 `GetCommand` API 获取有关特定命令的更多信息。有关更多信息，请参阅 [在你的账户中列出命令](#)。

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

运行此命令会生成以下响应。您将看到命令的创建时间和上次更新时间、您指定的任何参数以及该命令是否可以在设备上运行。

```
{
  "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
}
```

有关使用此命令的其他示例，请参见[检索关于命令的信息](#)。

### 3. 开始执行命令

运行以下命令开始执行该命令，其中 `command-arn` 是上面 `get-command` 操作输出中的命令 ARN。`target-arn` 是您正在为其执行命令的目标设备的 ARN，例如。 *myVehicle*

在此示例中，由于您在创建命令时提供了参数的默认值，因此 `start-command-execution` CLI 可以在执行命令时使用这些值。在使用 CLI 时，您也可以通过为参数指定不同的值来选择覆盖默认值。

```
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
```

运行此命令会返回一个命令执行 ID。您可以使用此 ID 查询命令执行状态、详细信息和命令执行历史记录。

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

有关使用 CLI 的其他示例，请参阅[发送命令 \(AWS CLI\)](#)。

### 4. 检索有关命令执行的信息

运行以下命令以检索有关您在目标设备上执行的命令的信息。指定作为上面 `start-command-execution` 操作输出获得的 `target-arn`，以及作为目标设备的 ARN。`execution-id`

**Note**

- 要获取最新的状态信息，您的设备必须已将更新的状态信息发布到使用 MQTT API 的命令的 M UpdateCommandExecution QTT 预留响应主题。有关更多信息，请参阅 [更新命令执行结果](#)。
- 如果您开始执行多个命令，则可以使用 ListCommandExecutions API 列出账户中的所有命令执行，然后使用 GetCommandExecution API 获取有关特定执行的更多信息。有关更多信息，请参阅 [列出您账户中的命令执行情况](#)。

```
aws iot get-command-execution \  
  --execution-id <"07e4b780-7eca-4ffd-b772-b76358da5542"> \  
  --target-arn arn:aws:iot:us-east-1:<account>:thing/myVehicle
```

运行此命令会返回有关命令执行、执行状态、开始执行的时间和完成时间的信息。例如，以下响应显示在目标设备上成功执行命令并且转向模式已关闭。

```
{  
  "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" }  
  }'
```

```
}'  
}
```

## 清理

现在，您已经创建了一个命令并在您的设备上执行了该命令，如果您不再打算使用此命令，则可以将其删除。任何正在执行的待执行命令将继续运行，不会受到删除请求的影响。

### Note

或者，如果命令已过时，您也可以将其弃用，以后可能需要使用它才能在目标设备上运行。

### 1. (可选) 弃用命令资源

运行以下命令以弃用该命令，其中 `command-id` 是上面 `get-command` 操作输出中的命令 ID。

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

运行此命令会返回一个输出，显示该命令已被弃用。您也可以使用 CLI 恢复命令。

### Note

您也可以使用 `update-command` CLI 更新命令的显示名称和描述。有关更多信息，请参阅 [更新或弃用命令资源](#)。

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

### 2. 删除该命令

运行以下命令删除由指定的命令 `command-id`。

**Note**

删除操作是永久性的，无法撤销。

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

如果删除请求成功，您将看到 202 或 204 的 HTTPstatusCode，具体取决于您是否将该命令标记为已弃用以及何时被弃用。有关更多信息以及示例，请参阅 [删除命令资源](#)。

您可以使用 get-command CLI 来验证该命令是否已从您的账户中删除。

### 3. (可选) 删除命令执行

默认情况下，所有命令执行将在您创建命令之日起六个月内删除。您可以使用 GetCommandExecution API 中的 timeToLive 参数查看此信息。

或者，如果您的命令执行已变为终止，例如当您的执行状态为 SUCCEEDEDFAILED、或之一时 REJECTED，则可以删除该命令的执行。运行以下命令删除执行，其中 execution-id 是上面 get-command-execution 操作输出中的执行 ID。

```
aws iot delete-command-execution \  
    --execution-id "07e4b780-7eca-4ffd-b772-b76358da5542"
```

您可以使用 get-command-execution CLI 验证命令执行是否已从您的账户中删除。

## 命令使用场景

**Important**

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

使用命令功能时，可以在以下情况下创建和运行命令：

- 在创建过程中，您可以省略参数并仅指定命令 ID。在这种情况下，您需要指定在目标设备上运行命令时要使用的参数。
- 您可以指定一个或多个参数，并在创建命令时为其配置默认值。提供默认值将有助于防止您发送不准确的命令。
- 您可以指定一个或多个参数，并在创建命令时为其配置值。可以提供多个参数，但只能执行其中一个，并且此参数的Name字段必须使用前\$actuatorPath缀。

本节提供了CreateCommand和 StartCommandExecution API 的一些使用场景以及参数的使用。它还向您展示了在状态模板中使用命令的一些示例。

## 主题

- [创建不带参数的命令](#)
- [使用参数的默认值创建命令](#)
- [使用参数值创建命令](#)
- [使用带有状态模板的命令](#)

## 创建不带参数的命令

以下用例展示了如何使用 CreateCommand API 或 create-command CLI 创建不带参数的命令。创建命令时，您只需要提供命令 ID 和角色 ARN。

此用例在反复出现的用例中特别有用，例如当你想多次向车辆发送相同的命令时。在这种情况下，该命令与特定的执行器无关，因此您可以灵活地在任何执行器上执行命令。使用 StartCommandExecution API 或 start-command-execution CLI 执行命令时，必须改为在运行时指定参数，包括执行器和物理信号值。

在没有**mandatory-parameters**输入的情况下创建命令

此用例展示了如何在不输入任何必填参数的情况下创建命令。

```
aws iot create-command \  
  --command-id "UserJourney1" \  
  --role-arn "arn:aws:iam:accountId:role/FwCommandExecutionRole" \  
  --description "UserJourney1 - No mandatory parameters" \  
  --namespace "AWS-IoT-FleetWise"
```

## 运行在没有mandatory-parameters输入的情况下创建的命令

在第一个示例中，上面创建的命令允许您在任何执行器上不受限制地执行命令。actuator1要将值设置为 10，请运行：

```
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"}  
  }'
```

同样，您可以运行将值设置actuator3为的命令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"}  
  }'
```

## 使用参数的默认值创建命令

此命令仅允许您在指定的执行器上执行命令。提供默认值将有助于防止您发送不准确的命令。例如，可以将锁定和解锁门的LockDoor命令配置为默认值，以避免该命令意外解锁门。

当您想要多次发送相同的命令并在同一个执行器上执行不同的操作（例如锁定和解锁车辆的车门）时，此用例特别有用。如果要将执行器设置为默认值，则无需向 start-command-execution CLI 传递 qny parameters。如果您确实在 start-command-execution CLI parameters 中为指定了不同的值，则它将覆盖默认值。

### 使用默认值为创建命令 mandatory-parameters

以下命令显示如何为 actuator1 提供默认值。

```
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"}
    }
]'
```

## 运行使用默认值创建的命令 **mandatory-parameters**

该命令 `UserJourney2` 允许您执行命令，而无需在运行时传递输入值。在这种情况下，运行时执行将使用创建期间指定的默认值。

```
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
```

您还可以在运行时为同一个执行器 `actuator1` 传递不同的值，这将覆盖默认值。

```
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"}  
  }'
```

## 使用参数值创建命令

此命令仅允许您在指定的执行器上执行命令。它还会强制您在运行时为执行器设置一个值。

当您希望最终用户在车辆上运行某些执行器时，仅对某些执行器执行某些指定操作时，此用例特别有用。

### Note

您可以为 `mandatory-parameters` 输入设置多个名称/值对，其中部分或全部为默认值。然后，在运行时，您可以确定在执行器上运行时要使用的参数，前提是执行器名称使用带 `$actuatorPath.` 前缀的完全限定名称。

## 正在创建不带默认值的命令 **mandatory-parameters**

此命令仅允许您在指定的执行器上执行命令。它还会强制您在运行时为执行器设置一个值。

```
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"  
    }  
  ]'
```

## 运行在没有默认值的情况下创建的命令 `mandatory-parameters`

在这种情况下，在运行命令时，必须为 `actuator1` 指定一个值。如下所示的命令执行将成功地将的值设置 `actuator1` 为 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"}  
  }'
```

## 使用带有状态模板的命令

您还可以使用命令 API 操作来收集和處理状态数据。例如，您可以获取一次性状态快照，或者激活或停用状态模板以开始或停止收集车辆状态数据。以下示例展示了如何将命令功能与状态模板一起使用。有关更多信息，请参阅 [用于数据收集和處理的状态模板操作](#)。

### Note

作为 `mandatory-parameters` 输入一部分指定的“名称”字段必须使用前 `$stateTemplate` 缀。

### 示例 1：使用默认值为状态模板创建命令

此示例说明如何使用 `create-command` CLI 激活状态模板。

```
aws iot create-command \  
  --command-id <COMMAND_ID> \  
  --state-template <STATE_TEMPLATE> \  
  --role-arn <ROLE_ARN> \  
  --mandatory-parameters '[  
    {  
      "name": "$stateTemplate.Vehicle.actuator1"  
    }  
  ]'
```

```

--display-name "Activate State Template" \
--namespace AWS-IoT-FleetWise \
--mandatory-parameters '[
  {
    "name": "$stateTemplate.name"
  },
  {
    "name": "$stateTemplate.operation",
    "defaultValue": {"S": "activate"}
  }
]'

```

同样，以下命令显示了如何将 `start-command-execution` CLI 用于状态模板的示例。

```

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"}
}'

```

## 示例 2：为没有默认值的状态模板创建命令

以下命令创建多个状态模板，其中任何参数都没有默认值。它会强制你使用这些参数及其值来运行命令。

```

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"}
  }
]'

```

```
]'
```

以下命令显示了如何使用上述示例的 `start-command-execution` CLI。

```
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"}  
  }'
```

# 监控车辆的最近已知状态

## Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

通过创建状态模板并将其与您的车辆相关联，您可以近乎实时地监控车辆的最新已知状态。与状态模板关联的车辆使用 `onChange` 或 `periodic` 更新策略流式传输遥测数据。通过变更更新策略，当发生变化时，关联的车辆会流式传输遥测数据。在定期更新策略期间，关联的车辆会在指定时间段内流式传输遥测数据。

通过按需操作，您可以一次性请求当前的车辆状态（获取）。您还可以激活或停用先前部署的状态模板，以开始或停止报告车辆状态数据。上次已知的状态操作是使用 AWS IoT 命令执行的 APIs。

每个状态模板都包含以下信息。

`name`

状态模板的唯一别名。

`signalCatalogArn`

与状态模板关联的信号目录的 Amazon 资源名称 (ARN)。

`stateTemplateProperties`

从中收集数据的信号列表。状态模板属性决定了车辆向云端发送的特定信号更新。

`dataExtraDimensions`

要包含在协议缓冲区 (Protobuf) 编码的已处理数据中的车辆属性列表。

`metadataExtraDimensions`

要发布的车辆属性列表，处理后的数据将作为 MQTT 5 用户属性发布。

`id`

由服务生成的唯一标识符。

有关收集使用 AWS 物联网边缘代理 FleetWise 软件的车辆发送的数据的方法，请参阅[使用 MQTT 消息处理上次已知的状态车辆数据](#)。有关如何将状态模板与车辆关联的更多信息，请参阅[创建物 AWS 联网 FleetWise 汽车](#)。

## 主题

- [创建 AWS 物联网 FleetWise 状态模板](#)
- [更新物 AWS 联网 FleetWise 状态模板](#)
- [删除物 AWS 联网 FleetWise 状态模板](#)
- [获取 AWS 物联网 FleetWise 状态模板信息](#)
- [用于数据收集和处理的模板操作](#)

## 创建 AWS 物联网 FleetWise 状态模板

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 AWS IoT FleetWise API 或控制台来创建状态模板。状态模板提供了一种跟踪车辆状态的机制。在车辆上运行的 AWS 物联网边缘代理 FleetWise 软件收集信号更新并将其发送到云端。

### 创建状态模板（控制台）

您可以使用 AWS IoT FleetWise 控制台创建状态模板。

#### 创建状态模板

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格上，选择状态模板。
3. 在状态模板页面上，选择创建状态模板。
4. 在状态模板详细信息中，输入状态模板的名称，也可以输入描述。
5. 在选择信号中，添加要从中获取车辆状态信息的信号。
6. 选择“创建状态模板”。

成功创建状态模板后，您将在状态模板页面上看到该模板列出。现在，您可以将其与车辆相关联。

## 创建状态模板 (AWS CLI)

您可以使用 [CreateStateTemplate](#) API 操作来创建状态模板。下面的示例使用了 AWS CLI。

要创建状态模板，请运行以下命令。

`create-state-template` 替换为包含状态模板配置的 .json 文件的名称。

```
aws iotfleetwise create-state-template \  
  --cli-input-json file://create-state-template.json
```

### Example 状态模板配置

`stateTemplateProperties` 应包含信号的完全限定名称。

`dataExtraDimensions` 并 `metadataExtraDimensions` 应包含车辆属性的完全限定名称。指定的维度将替换状态模板中的任何现有维度值。

```
{  
  "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"  
  ]  
}
```

## 将物 AWS 联网 FleetWise 状态模板与车辆关联

将状态模板与车辆相关联 (控制台)

您可以使用 AWS IoT FleetWise 控制台向车辆添加关联状态模板。

## 关联状态模板

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆。
3. 从列表中选择一辆车以打开其详细信息页面。
4. 在州模板选项卡上，选择管理州模板。
5. 选择“添加状态模板”。
6. 选择状态模板并选择其报告方法。
  - a. 更改时 — 状态模板将报告车辆状态的变化。
  - b. 定期-状态模板将在指定的时间间隔内报告更新。
7. 选择保存更改。

## 将物 AWS 联网 FleetWise 状态模板与车辆关联 (AWS CLI)

将创建的状态模板与车辆相关联，以允许将状态更新从车辆收集到云端。为此，请使用：

- 创建车辆时，请使用create-vehicle命令stateTemplates字段。有关更多信息，请参阅[创建物 AWS 联网 FleetWise 汽车](#)。
- 更新车辆时，请使用update-vehicle命令的stateTemplatesToAdd或stateTemplatesToRemove字段。有关更多信息，请参阅[更新 AWS 物联网 FleetWise 车辆](#)。

## 更新物 AWS 联网 FleetWise 状态模板

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅[AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [UpdateStateTemplate](#) API 操作或 AWS IoT FleetWise 控制台更新现有状态模板。

### 更新状态模板 (控制台)

要从控制台更新状态模板，请转到 AWS 物联网 FleetWise 控制台的[状态模板](#)页面并执行以下步骤。

1. 选择要更新的状态模板，然后选择“编辑”。
2. 编辑状态模板的详细信息，然后选择保存更改。

## 更新状态模板 (AWS CLI)

要更新状态模板，请运行以下命令。

*update-state-template* 替换为包含状态模板配置的.json 文件的名称。

```
aws iotfleetwise update-state-template \  
  --cli-input-json file://update-state-template.json
```

### Example 状态模板配置

stateTemplateProperties 应包含信号的完全限定名称。

dataExtraDimensions 和 metadataExtraDimensions 应包含车辆属性的完全限定名称。

```
{  
  "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"  
  ]  
}
```

## 删除物 AWS 联网 FleetWise 状态模板

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [DeleteStateTemplate](#) API 操作或 AWS IoT FleetWise 控制台删除状态模板。

### 删除状态模板 ( 控制台 )

要从控制台删除状态模板，请转到 AWS 物联网 FleetWise 控制台的 [状态模板](#) 页面并执行以下步骤。

1. 选择要删除的状态模板，然后选择“删除”。
2. 确认要删除状态模板，然后选择删除。

### 删除状态模板 (AWS CLI)

要删除状态模板，请运行以下命令。

*identifier* 替换为状态模板的名称或 ID。

```
aws iotfleetwise delete-state-template \  
  --identifier idenitfier
```

## 获取 AWS 物联网 FleetWise 状态模板信息

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [GetStateTemplate](#) API 操作来检索有关状态模板的信息。下面的示例使用了 AWS CLI。

*identifier* 替换为状态模板的名称。

```
aws iotfleetwise get-state-template \  
  --name identifier
```

```
--identifier idenitfier
```

您可以使用 [ListStateTemplates](#) API 操作来检索您创建的状态模板的列表。下面的示例使用了 AWS CLI。

```
aws iotfleetwise list-state-templates
```

如果您启用了[使用客户托管 AWS KMS 密钥的加密](#)，请包括以下政策声明，以便您的角色可以调用 `GetStateTemplate` 或 `ListStateTemplates` API 操作。

JSON

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

## 用于数据收集和处理的状态模板操作

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下各节介绍如何使用状态模板来激活和停用数据收集、执行提取操作以及处理车辆的状态数据。

主题

- [使用状态模板激活和停用状态数据收集](#)
- [使用状态模板获取车辆状态快照](#)
- [使用 MQTT 消息处理上次已知的状态车辆数据](#)

## 使用状态模板激活和停用状态数据收集

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下各节介绍如何使用状态模板激活和停用数据提取。AWS CLI

### ⚠ Important

在开始之前，请确保您已经创建了[状态模板](#)，并将其及其更新策略与车辆相关联。

您必须激活状态模板，这样 Edge Agent 才能向云端发送信号更新。

要使用状态模板执行这些操作，请先创建命令资源，然后在车辆上开始执行命令。以下部分介绍如何使用此 API 以及如何激活和停用数据提取。

### 主题

- [使用 CreateCommand API](#)
- [示例：激活状态模板](#)
- [示例：停用状态模板](#)

## 使用 CreateCommand API

在“AWS-IoTFleetwise”命名空间中创建命令资源，并在为状态模板创建或发送命令资源时使用以下参数：

- `$stateTemplate.name`— 要对其执行操作的状态模板的名称。必须先将状态模板应用于车辆，然后才能执行操作。有关更多信息，请参阅 [将物 AWS 联网 FleetWise 状态模板与车辆关联](#)。
- `$stateTemplate.operation`— 要对状态模板执行的操作。为此参数使用以下值之一：

- `activate`— Edge Agent 开始根据 `stateTemplateUpdateStrategy` 您在将状态模板应用于车辆时指定的（变更或定期）向云端发送信号更新。有关更多信息，请参阅 [将物 AWS 联网 FleetWise 状态模板与车辆关联](#)。

此外，您可以定义自动状态模板停用时间，以便在指定时间段后停止更新。如果未提供自动停用时间，则状态模板将继续发送更新，直到发出停用呼叫。

收到 `activate` 命令后，设备应立即根据更新策略发送状态模板中指定的信号。AWS 物联网 FleetWise 建议，当设备收到激活命令时，它发送的第一条消息应包含状态模板中所有信号的快照。后续消息应根据更新策略发送。

- `deactivate`— 边缘代理停止向云端发送信号更新。
- `fetchSnapshot`— 无论 `stateTemplateUpdateStrategy` 您在将状态模板应用于车辆时指定了什么，Edge Agent 都会发送状态模板中定义的信号的一次性快照。
- （可选）`$stateTemplate.deactivateAfterSeconds`-状态模板在指定时间后自动停用。只有当参数的值为“激活”时，才能使用 `$stateTemplate.operation` 此参数。如果未指定此参数，或者此参数的值为 0，则 Edge Agent 会一直向云端发送信号更新，直到收到状态模板的“停用”操作。状态模板永远不会自动停用。

最小值：0，最大值：4294967295。

#### Note

- API 会根据已处于活动状态的模板的激活请求返回成功。
- 针对已处于停用状态的模板发出的停用请求，API 会返回成功。
- 您最近在状态模板上提出的请求是生效的请求。例如，如果您请求在一小时内停用状态模板，然后再次请求在四小时内停用该模板，则四小时的停用将生效，因为这是最新的请求。

#### Important

在以下任何一种情况下都可能发生验证异常：

- 提供了一个状态模板，该模板不 ASSOCIATED 包含在车辆上。
- 有人请求激活状态模板，但该模板尚未出现在车辆 DEPLOYED 上。
- 有人向状态模板发出了请求，但它在车辆 DELETED 上。

## 示例：激活状态模板

要激活状态模板，请先创建一个命令资源。然后，您可以向要激活状态模板的车辆发送以下命令。此示例说明如何在创建命令时为参数指定默认值。这些参数及其值用于启动命令执行以激活状态模板。

### 1. 创建命令资源

必须先创建命令资源，然后才能向车辆发送命令。向车辆发送命令时，您可以为必填参数指定替代值。有关更多信息，请参阅 [创建命令资源](#)。

#### Important

`$stateTemplate.name` 并且必须以字符串数据类型提供 `$stateTemplate.operation` 参数。如果提供了任何其他数据类型，或者缺少这两个参数中的任何一个，则命令执行失败并出现验证异常。必须将 `$stateTemplate.deactivateAfterSeconds` 参数作为 Long 数据类型提供。

```
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. 开始在车辆上执行命令

创建命令后，将命令发送到车辆。如果您在创建命令资源时没有为必填参数指定值，则必须立即指定它们。有关更多信息，请参阅 [发送命令 \(AWS CLI\)](#)。

### Important

确保使用特定于账户的 AWS IoT 任务数据平面 API 端点进行 API 操作。

```
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. 检索状态模板操作的状态

开始执行命令后，您可以使用 GetCommandExecution API 来检索状态模板。

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

## 示例：停用状态模板

要停用状态模板，请先创建一个命令资源。然后，您可以向要停用状态模板的车辆发送以下命令。此示例说明如何在创建命令时为参数指定默认值。这些参数及其值将在开始执行命令时使用，以停用状态模板。

### 1. 创建命令资源

必须先创建命令资源，然后才能向车辆发送命令。向车辆发送命令时，您可以为必填参数指定替代值。有关更多信息，请参阅 [创建命令资源](#)。

```
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. 开始在车辆上执行命令

创建命令后，将命令发送到车辆。如果您在创建命令资源时没有为必填参数指定值，则必须立即指定它们。有关更多信息，请参阅 [发送命令 \(AWS CLI\)](#)。

```
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. 检索状态模板操作的状态

开始执行命令后，您可以使用 `GetCommandExecution` API 来检索状态模板。

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

## 使用状态模板获取车辆状态快照

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

您可以使用 [CreateCommand](#) AWS IoT Core 控制平面 API 操作或 AWS 物联网控制 FleetWise 台来获取车辆的最后已知状态。

### Important

在以下任何一种情况下都可能发生验证异常：

- 提供了一个状态模板，该模板不ASSOCIATED包含在车辆上。

- 有人请求激活状态模板，但该模板尚未出现在车辆DEPLOYED上。
- 有人向状态模板发出了请求，但它在车辆DELETED上。

## 获取车辆状态快照 (控制台)

您可以使用 AWS 物联网 FleetWise 控制台获取车辆的最后已知状态。AWS IoT FleetWise 将创建一个命令供您获取数据。

### 获取车辆的状态

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在导航窗格中，选择车辆。
3. 从列表中选择一辆车以打开其详细信息页面。
4. 在“状态模板”选项卡上，选择“获取数据”。
5. 选择向 AWS 物联网授予发送命令和获取数据的 FleetWise 权限的 IAM 角色。请参阅[控制访问权限](#)。
6. 选择“获取状态”。

## 获取车辆状态快照 (AWS CLI)

要获取状态快照，请先创建一个命令资源。然后，您可以向要获取状态快照的车辆发送以下命令。有关使用 CreateCommand API 及其参数的更多信息，请参阅[使用 CreateCommand API](#)。

### 1. 创建命令资源

以下示例说明如何创建用于执行读取操作的命令资源。向车辆发送命令时，您可以为必填参数指定替代值。有关更多信息，请参阅[创建命令资源](#)。

```
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"}
    }
  ]'
```

响应：

```
{
  "commandId": "<COMMAND_ID>",
  "commandArn": "arn:aws:iot:<REGION>:111122223333:command/<COMMAND_ID>"
}
```

## 2. 开始执行命令以获取状态快照

创建命令后，将命令发送到车辆。如果您在创建命令资源时没有为必填参数指定值，则必须立即指定它们。有关更多信息，请参阅 [发送命令 \(AWS CLI\)](#)。

```
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>
```

响应：

```
{
  "executionId": "<UNIQUE_UUID>"
}
```

## 3. 检索状态模板操作的状态

开始执行命令后，您可以使用 `GetCommandExecution` API 来检索状态模板。

```
aws iot get-command-execution --execution-id <EXECUTION_ID>
```

## 使用 MQTT 消息处理上次已知的状态车辆数据

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

要接收车辆的更新并处理其数据，请订阅以下 MQTT 主题。有关更多信息，请参阅《AWS IoT Core 开发人员指南》中的 [MQTT 主题](#)。

```
$aws/iotfleetwise/vehicles/$vehicle_name/last_known_state/$state_template_name/data
```

由于 MQTT 不保证排序，因此收到的最后一个已知状态信号更新消息可能会失序。任何使用 MQTT 接收和处理车辆数据的客户端都必须处理这个问题。最新的已知状态信号更新消息遵循 MQTT 5 消息协议。

每条 MQTT 消息的消息头都具有以下用户属性：

- 车辆名称 — [车辆的唯一标识符](#)。
- stateTemplateName — 上次已知[状态模板](#)的唯一标识符。

此外，通过在更新或创建状态模板时指定 metadataExtraDimensions 请求参数，您可以指定要包含在 MQTT 消息标头中的 [车辆属性](#)。（参见 [状态模板](#)。）

MQTT 消息标头中的用户属性可用于在不检查负载的情况下将消息路由到不同的目的地。

MQTT 消息有效载荷包含从车辆收集的数据。通过在创建或更新状态模板时指定 extraDimensions 请求参数，您可以指定要包含在 MQTT 消息负载中的车辆属性（请参阅 [创建 AWS 物联网 FleetWise 状态模板](#)）。额外的维度通过将额外的维度与车辆关联来丰富从车辆收集的数据。

MQTT 消息有效载荷采用协议缓冲区 (Protobuf) 编码，MQTT 消息头包含定义为应用程序/八字字节流的内容类型指示器。Protobuf 的编码架构如下所示：

```
// 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;
  }
}

```

其中：

- `time_ms`:

车辆中触发事件时的绝对时间戳（以 Unix Epoch 以来的毫秒为单位）。Edge Agent 软件在车辆的时钟上使用此时间戳。

- `signal`:

包含信号信息：`name(string)` 的 `Signal s` 数组，`signalValue` 它支持以下数据类型

型-`double`、`bool`、`int8`、`uint8`、`int16`、`uint16`、`int32`、`uint32`、`int64`

- `extra_dimensions`:

ExtraDimensions包含车辆属性信息的数组：name ( 字符串 ) extraDimensionValue , 目前仅支持该string数据类型。

# 教程：使用自定义解码接口配置与网络无关的数据收集

## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

## 简介

本教程概述了如何使用网络不可知的数据收集（使用自定义解码接口）配置 AWS 物联网 FleetWise 网以收集数据和运行命令。通过与网络无关的数据收集，您可以使用自己的方法对信号进行解码，然后再将信号发送到指定的数据目的地。这可以节省时间，因为您无需专门为 AWS 物联网 FleetWise 创建信号解码器。您可以使用自己的实现对信号子集进行解码，也可以在创建或更新解码器清单 defaultForUnmappedSignals 时使用。这也为收集车辆内各种信号源的信号和触发器提供了灵活性。

本教程适用于不在标准控制器局域网 (CAN 总线) 接口上的车辆信号。例如，以自定义车载格式或方案编码的数据。

## 环境设置

本教程假设您已经完成了设置环境以访问 AWS 物联网 FleetWise 云以及边缘实现 APIs 和代码库的步骤。

## 数据模型

下一节说明如何使用自定义解码接口对车辆属性进行建模。这适用于数据收集和命令用例。例如，它也适用于车辆中使用的任何基础数据源建模 IDLs。

在示例中，有两个车辆属性：要收集的车辆传感器（当前车辆位置）和用于远程控制的车辆执行器（空调）。这两个都是在这个方案中定义的：

```
// Vehicle WGS84 Coordinates
double Latitude;
double Longitude;

// Vehicle AC
```

```
Boolean ActivateAC;
```

下一步是 FleetWise 使用自定义解码接口 APIs 将这些定义导入 AWS 物联网。

## 信号目录更新

将这些定义导入您的信号目录。如果您在 AWS 物联网中 FleetWise 已经有信号目录，请直接使用更新 API。如果没有，请先创建一个信号目录，然后调用更新 API。

首先，必须创建这些车辆信号的 VSS 表示形式。VSS 用作分类法来表示物联网 AWS 中的车辆数据。FleetWise 创建一个名为 “vehicle-signals.json” 的 json 文件，其中包含以下内容：

```
// 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": {
      "fullyQualifiedNames": ["Vehicle.ActivateAC"],
      "description": "AC Controller",
      "dataType": "BOOLEAN"
    }
  }
]

```

如果您没有信号目录，则需要调用 `create-signal-catalog`：

```

VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise create-signal-catalog \
  --name my-signal-catalog \
  --nodes "${VEHICLE_NODES}"

```

如果您已经有信号目录，则可以使用 `update-signal-catalog` API 添加这些信号：

```

VEHICLE_NODES=`cat vehicle-signals.json`
aws iotfleetwise update-signal-catalog \
  --name my-signal-catalog \
  --nodes-to-add "${VEHICLE_NODES}"

```

## 车辆模型和解码器

在信号目录中插入信号后，下一步是创建车辆模型并实例化这些信号。为此，您可以使用 `create-model-manifest` 和 `create-decoder-manifest` APIs。

首先，格式化要插入车辆模型的信号名称：

```

# Prepare the signals for insertion into the vehicle model.
VEHICLE_NODES=`cat vehicle-signals.json`
VEHICLE_NODES=`echo ${VEHICLE_NODES} | jq -r ".[] | .actuator,.sensor
| .fullyQualifiedNames" | grep Vehicle\\.\\. `
VEHICLE_NODES=`echo "${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

```

现在，使用自定义解码接口创建解码器清单。

### Note

如果要指定自定义，则只需要创建网络接口和信号 IDs，这不是本示例的一部分。有关在完全限定名称 (FQN) 与自定义解码信号 ID 不同时映射解码信息的信息，请参阅 [Edge Agent](#) 开发者指南。

```

// 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

```

此时，您已经在 AWS 物联网中对这些信号进行了全面建模 FleetWise。接下来，创建车辆并将其与您创建的模型相关联。您可以使用 `create-vehicle` API 来做到这一点：

```

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"
```

下一步是重点关注 AWS IoT FleetWise Edge 代码库并编写必要的代码扩展。

### Note

有关 Edge 实现的信息，请参阅 [《边缘代理开发者指南》](#)。

## 发送命令

现在，编译软件（确保将头文件和 C++ CMake 文件添加到文件中），然后返回云端 APIs 在此执行器上测试命令：

```
// 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]
```

## 使用 AWS CLI 和 AWS SDKs 搭配 AWS 物联网 FleetWise

本节提供有关发出 AWS IoT FleetWise API 请求的信息。有关 AWS 物联网 FleetWise [操作和数据类型的更多信息](#)，请参阅 [FleetWise I AWS IoT API 参考](#)。

要将 AWS IoT FleetWise 与各种编程语言一起使用 [AWS SDKs](#)，请使用，其中包含以下自动功能：

- 对服务请求进行加密签名
- 重试请求
- 处理错误响应

要访问命令行，请将 AWS IoT FleetWise 与[AWS CLI](#)。您可以从命令行控制 AWS 物联网 FleetWise 和其他服务，并通过脚本实现自动化。

# AWS 物联网故障排除 FleetWise

使用本节中的故障排除信息和解决方案来帮助解决 AWS 物联网问题 FleetWise。

以下信息可以帮助您解决 AWS 物联网的常见问题 FleetWise。

## 主题

- [AWS 物联网 FleetWise 解码器存在问题](#)
- [用于解决 AWS 物联网 FleetWise 软件问题的边缘代理](#)
- [存储和转发问题](#)

## AWS 物联网 FleetWise 解码器存在问题

对解码器清单问题进行故障排除。

### 诊断解码器清单 API 调用问题

错误	故障排除指南
<code>UpdateOperationFailure.ConflictingDecoderUpdate</code>	同一个解码器清单有多个更新请求。等待，然后重试。
<code>UpdateOperationFailure.InternalFailure</code>	<code>InternalFailure</code> 作为封装的异常启动。问题本身取决于封装的异常。
<code>UpdateOperationFailure.ActiveDecoderUpdate</code>	解码器清单处于 <code>Active</code> 状态，无法更新。将解码器清单状态更改为 <code>DRAFT</code> ，然后重试。
<code>UpdateOperationFailure.ConflictingModelUpdate</code>	AWS 物联网 FleetWise 正在尝试对照其他人正在修改的车辆模型（模型清单）进行验证。等待，然后重试。
<code>UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND</code>	车辆模型没有任何与之关联的信号。向车辆模型添加信号，并确认信号是否可以在相关的信号目录中找到。

错误	故障排除指南
<code>UpdateOperationFailure.Mode 1ManifestValidationResponse : FailureReason.MODEL_NOT_ACTIVE</code>	更新车辆模型使其处于 ACTIVE 状态，然后重试。
<code>UpdateOperationFailure.Mode 1ManifestValidationResponse : FailureReason.MODEL_NOT_FOUND</code>	AWS 物联网找 FleetWise 不到与解码器清单关联的车辆模型。确认车辆模型的 Amazon 资源名称 (ARN)，然后重试。
<code>UpdateOperationFailure.Mode 1ManifestValidationResponse (FailureReason.MODEL_DATA_E NTRIES_READ_FAILURE</code>	由于在信号目录中找不到来自车辆模型的信号名称，因此车辆模型验证失败。确认车辆模型中的信号是否全部包含在关联的信号目录中。
<code>UpdateOperationFailure.Vali dationFailure</code>	在更新解码器清单的请求中发现了无效的信号或网络接口。确认异常返回的所有信号和网络接口都存在，所有使用的信号都与可用接口相关联，并且您不会删除具有与之关联的信号的接口。
<code>UpdateOperationFailure.KmsK eyAccessDenied</code>	用于操作的 AWS Key Management Service (AWS KMS) 键存在权限问题。确认您使用的角色是否有权访问此密钥，然后重试。
<code>UpdateOperationFailure.Deco derDoesNotExist</code>	解码器清单不存在。确认解码器清单名称，然后重试。

带有 `SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG` 原因的视觉系统数据错误消息将在响应中包含提示，提供有关请求失败原因的信息。您可以使用该提示来确定要遵循哪些故障排除指南。

#### Note

视觉系统数据目前为预览版，可能会发生变化。

## 诊断解码器清单视觉系统数据验证问题

错误	故障排除指南
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</code>	AWS 物联网 FleetWise 未使用信号目录找到信号解码器中使用的根信号结构。确认信号目录中是否正确定义了结构的根信号。
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</code>	未在解码器清单更新请求中使用相同的数据类型定义信号目录中的原始消息。确认请求中定义的原始消息是否与其相应的信号目录定义相匹配。
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</code>	在信号目录结构中定义的属性数量与您在解码器清单中尝试解码的属性数量不匹配。通过将信号与信号目录中定义的信号进行比较，确认要解码的信号数量是否正确。
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	AWS 物联网在信号目录中 FleetWise 发现了一个定义为 STRUCT 的信号，但解码器清单请求中没有 <code>structuredMessageDefinition</code> 定义。确保在解码器清单更新请求 <code>structuredMessageDefinition</code> 中将每个结构定义为。
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	未在信号目录中将解码器清单中使用的结构的根信号正确定义为结构。解码器清单中使用的根信号结构必须定义其 <code>structFullyQualified</code> 字段 <code>Name</code> 。它还需要一个带有它的 STRUCT 节点。 <code>fullyQualifiedName</code>
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	解码器清单请求中使用的叶消息之一未定义为原始消息。确认请求中的所有叶对象是否都定义为原始消息。
<code>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</code>	信号目录中的数组对象未在解码器清单更新请求中定义为 <code>structuredMessageList</code> 定义。在

错误	故障排除指南
ason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)	解码器清单更新请求中，确认所有数组属性均 structuredMessageList 定义为“定义”。

## 用于解决 AWS 物联网 FleetWise 软件问题的边缘代理

排查边缘代理软件问题。

问题

- [问题：边缘代理软件无法启动。](#)
- [问题：\[错误\] \[IoTFleetWiseEngine:: connect\]: \[无法初始化持久性库\]](#)
- [问题：Edge Agent 软件不收集板载诊断 \(OBD\) II PIDs 和诊断故障代码 \(DTCs\)。](#)
- [问题：适用于 AWS IoT FleetWise 软件的 Edge Agent 无法从网络收集数据或无法应用数据检查规则。](#)
- [问题：\[错误\] \[AwsIotConnectivityModule:: connect\]: \[连接失败并出现错误\] 或 \[警告\] \[AwsIotChannel:: send\]: \[没有活动的 MQTT 连接。\]](#)

### 问题：边缘代理软件无法启动。

当边缘代理软件无法启动时，您可能会看到以下错误。

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

解决方案：确保 AWS 物联网边缘代理 FleetWise 软件配置文件使用有效的 JSON 格式。例如，请确保逗号使用正确。有关配置文件的更多信息，请执行以下操作下载适用于 AWS IoT 的 Edge Agent FleetWise 软件开发人员指南。

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在服务主页的 AWS 物联网入门 FleetWise 部分，选择探索 Edge Agent。

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

**解决方案：**当边缘代理软件无法与配置文件中定义的网络接口建立套接字通信时，您可能会看到此错误。

要检查配置中定义的网络接口是否可用，请运行以下命令。

```
ip link show
```

要使网络接口联机，请运行以下命令。*network-interface-id*替换为网络接口的 ID。

```
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]
```

**解决方案：**当边缘代理软件无法与 AWS IoT Core 建立 MQTT 连接时，您可能会看到此错误。检查以下各项是否配置正确，然后重新启动边缘代理软件。

- `mqttnConnection::endpointUrl`— AWS 账户的物联网设备端点。
- `mqttnConnection::clientId` – 运行边缘代理软件的车辆的 ID。
- `mqttnConnection::certificateFilename` – 车辆证书文件的路径。
- `mqttnConnection::privateKeyFilename` – 车辆私钥文件的路径。
- 你曾经 AWS IoT Core 调配过车辆。有关更多信息，请参阅 [配置 AWS 物联网 FleetWise 车辆](#)。

有关更多信息，请参阅 [AWS IoT Device SDK for C++ 常见问题解答](#)。

**问题：** [错误] [IoTFleetWiseEngine:: connect]: [无法初始化持久性库]

**解决方案：**当边缘代理软件找不到持久性存储时，您可能会看到此错误。检查以下各项是否配置正确，然后重新启动边缘代理软件。

`persistency:persistencyPath` – 用于保存收集方案、解码器清单和数据快照的本地路径。

**问题：** Edge Agent 软件不收集板载诊断 (OBD) II PIDs 和诊断故障代码 (DTCs)。

**解决方案：** 如果 `obdInterface:pidRequestIntervalSeconds` 或 `obdInterface:dtcRequestIntervalSeconds` 配置为 0，您可能会看到此错误。

如果边缘代理软件在自动变速箱车辆中运行，请确保 `obdInterface:hasTransmissionEcu` 配置为 `true`。

如果您的车辆支持扩展控制器局域网 (CAN 总线) 仲裁 IDs，请确保将其配置 `obdInterface:useExtendedIds` 为 `true`。

**问题：** 适用于 AWS IoT FleetWise 软件的 Edge Agent 无法从网络收集数据或无法应用数据检查规则。

**解决方案：** 当违反默认配额时，您可能会看到此错误。

资源	限额	可调整	备注
信号 ID 的值。	信号 ID 必须小于或等于 50,000	是	边缘代理软件不会从 ID 大于 50,000 的信号中收集数据。建议您在更改此配额之前，先检查信号目录包含多少个信号。
每辆车的活跃数据收集方案数量	256	是	建议您在更改此配额之前，先查看已在云端创建了多少个活动以及每个活动包含多少方案。
信号历史记录缓冲区的大小。	20 MB	是	如果超过此配额，边缘代理软件将停止收集新数据。

问题：[错误] [AwsIotConnectivityModule:: connect]: [连接失败并出现错误] 或 [警告] [AwsIotChannel:: send]: [没有活动的 MQTT 连接。]

解决方案：当边缘代理软件未连接到云端时，您可能会看到此错误。默认情况下，Edge Agent 软件 AWS IoT Core 每分钟发送一次 ping 请求并等待三分钟。如果没有响应，边缘代理软件会自动重新建立与云的连接。

## 存储和转发问题

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

问题：收到 **AccessDeniedException** 具有所有必需的 IAM 权限的

解决方案：用于在广告活动中进行数据分区的存储和转发功能需要通过允许列表进行封闭式访问。通过允许上架联系服务团队，确保您的资源拥有足够的权限。

问题：上传到 AWS IoT Jobs 的数据忽略了 **endTime**

解决方案：您在任务文档 `endtime` 中指定了无效内容。例如，`endtime` 不遵循 ISO 8601 UTC 格式)。在 AWS IoT FleetWise Agent 日志上，可能有一条警告级别的语句说，。Malformed IoT Job `endTime: customer configured endTime. Not setting endTime`

问题：上传到 AWS IoT 作业的数据处于 **REJECTED** 执行状态。

解决方案：您在任务文档 `campaignArn` 中指定了无效内容。例如，如果您为未在车辆上运行的活动指定 ARN，则代理日志 `CampaignArn value in the received job document does not match the ARN of a Store and Forward campaign` 中可能会显示错误级别的语句。AWS IoT FleetWise

# AWS 物联网安全 FleetWise

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。[责任共担模式](#)将此描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在 AWS 云中运行 AWS 服务的基础架构。AWS 还为您提供可以安全使用的服务。作为[AWS 合规计划](#)的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 AWS 物联网的合规计划 FleetWise，请参阅按合规计划划分的[AWS 范围内的服务 AWS 按合规计划](#)。
- 云端安全-您的责任由您使用的 AWS 服务决定。您还需要对其他因素负责，包括您的数据的敏感性、您的公司的要求以及适用的法律法规。

本文档可帮助您了解在使用 AWS IoT 时如何应用分担责任模型 FleetWise。它向您展示了如何配置 AWS 物联网 FleetWise 以满足您的安全和合规目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS 物联网 FleetWise 资源。

## 内容

- [AWS 物联网中的数据保护 FleetWise](#)
- [使用控制访问权限 AWS IoT FleetWise](#)
- [适用于 AWS 物联网的 Identity and Access 管理 FleetWise](#)
- [AWS AWS IoT 的托管策略更新 FleetWise](#)
- [AWS 物联网合规性验证 FleetWise](#)
- [AWS 物联网的弹性 FleetWise](#)
- [AWS 物联网中的基础设施安全 FleetWise](#)
- [AWS 物联网中的配置和漏洞分析 FleetWise](#)
- [AWS 物联网安全最佳实践 FleetWise](#)

## AWS 物联网中的数据保护 FleetWise

[责任 AWS 共担模型](#)适用于 AWS 物联网中的数据保护 FleetWise。如本模型所述 AWS，负责保护运行所有内容的全球基础架构 AWS Cloud。您负责维护对托管在此基础结构上的内容的控制。您还负

责您所使用的 AWS 服务 的安全配置和管理任务。有关数据隐私的更多信息，请参阅[数据隐私常见问题](#)。有关欧洲数据保护的信息，请参阅 AWS Security Blog 上的 [AWS Shared Responsibility Model and GDPR](#) 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户 凭证并使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 设置个人用户。这样，每个用户只获得履行其工作职责所需的权限。还建议您通过以下方式保护数据：

- 对每个账户使用多重身份验证 ( MFA )。
- 用于 SSL/TLS 与 AWS 资源通信。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 使用设置 API 和用户活动日志 AWS CloudTrail。有关使用 CloudTrail 跟踪捕获 AWS 活动的信息，请参阅 AWS CloudTrail 用户指南中的[使用跟 CloudTrail 跟踪](#)。
- 使用 AWS 加密解决方案以及其中的所有默认安全控件 AWS 服务。
- 使用高级托管安全服务（例如 Amazon Macie），它有助于发现和保护存储在 Amazon S3 中的敏感数据。
- 如果您在 AWS 通过命令行界面或 API 进行访问时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准 ( FIPS ) 第 140-3 版》<https://aws.amazon.com/compliance/fips/>。

强烈建议您切勿将机密信息或敏感信息（如您客户的电子邮件地址）放入标签或自由格式文本字段（如名称字段）。这包括您使用控制台、API FleetWise 或 AWS 服务 使用 AWS 物联网或其他设备时 AWS SDKs。AWS CLI 在用于名称的标签或自由格式文本字段中输入的任何数据都可能会用于计费或诊断日志。如果您向外部服务器提供 URL，强烈建议您不要在网址中包含凭证信息来验证对该服务器的请求。

AWS 物联网 FleetWise 旨在与您开发并安装在支持的车辆硬件上的边缘代理一起使用，以便将车辆数据传输到 AWS 云端。从车辆中提取数据可能受某些司法管辖区的数据隐私法规的约束。在使用 AWS 物联网 FleetWise 和安装边缘代理之前，我们强烈建议您评估适用法律规定的合规义务。这包括任何适用的法律要求，以便提供法律上充分的隐私声明并获得提取车辆数据的任何必要同意。

## AWS 物联网中的静态加密 FleetWise

从车辆收集的数据通过采用 MQTT 消息协议的 AWS IoT Core 消息传输到云端。AWS 物联网将数据 FleetWise 传输到您的亚马逊 Timestream 数据库。在 Timestream 中，您的数据将被加密。默认情况下，所有数据都 AWS 服务 加密静态数据。有关更多信息，请参阅 Amazon S3 用户指南中的[使用加密保护数据](#)和有关的 [Timestream 中的数据保护](#)。LiveAnalytics

静态加密与 AWS Key Management Service (AWS KMS) 集成，用于管理用于加密数据的加密密钥。您可以选择使用客户托管密钥对 AWS 物联网收集的数据进行加密 FleetWise。您可以通过创建、管理和查看您的加密密钥 AWS KMS。有关更多信息，请参阅[什么是 AWS Key Management Service?](#) 在《AWS Key Management Service 开发人员指南》中。

## 传输中加密

与 AWS IoT 服务交换的所有数据在传输过程中都使用传输层安全 (TLS) 进行加密。有关更多信息，请参阅《AWS IoT 开发人员指南》中的[传输安全](#)。

此外，还 AWS IoT Core 支持[身份验证](#)和[授权](#)，以帮助安全地控制对 AWS 物联网 FleetWise 资源的访问。车辆可以使用 X.509 证书进行身份验证（登录）以使用物 AWS 联网， FleetWise 并使用 AWS IoT Core 策略获得授权（拥有权限）以执行指定操作。有关更多信息，请参阅 [the section called “预配置车辆”](#)。

## AWS 物联网中的数据加密 FleetWise

数据加密是指保护传输中的数据（当数据往返 AWS 物联网时 FleetWise，以及在网关和服务器之间传输时）和静态数据（存储在本地设备上或存储在本地设备中时 AWS 服务）。可以使用客户端加密保护静态数据。

### Note

AWS 托管在物联网网关内并可通过本地网络访问的 AWS 物联 FleetWise 网 FleetWise 边缘处理公开 APIs 信息。APIs 它们通过由 AWS IoT FleetWise Edge 连接器拥有的服务器证书支持的 TLS 连接公开。对于客户端身份验证，它们 APIs 使用访问控制密码。服务器证书私钥和访问控制密码都存储在磁盘上。AWS 物联网 FleetWise 边缘处理依赖文件系统加密来保护这些静态凭证的安全。

有关服务器端加密和客户端加密的更多信息，请查看下面的主题。

### 内容

- [AWS 物联网中的静态加密 FleetWise](#)
- [AWS 物联网中的密钥管理 FleetWise](#)

## AWS 物联网中的静态加密 FleetWise

AWS 物联网将您的数据 FleetWise 存储在 AWS 云端和网关上。

## AWS 云端的静态数据

AWS 默认情况下 AWS 服务，物联网将数据 FleetWise 存储在静态数据进行加密的其他设备中。静态加密与 [AWS Key Management Service \(AWS KMS\)](#) 集成，用于管理加密密钥，该密钥用于加密 AWS 物联网中的资产属性值和聚合值 FleetWise。您可以选择使用客户托管密钥对物 AWS 联网中的资产属性值和聚合值进行加密 FleetWise。您可以通过创建、管理和查看您的加密密钥 AWS KMS。

您可以选择 AWS 拥有的密钥 或客户管理的密钥来加密您的数据。

### 工作原理

静态加密与集成，AWS KMS 用于管理用于加密数据的加密密钥。

- AWS 拥有的密钥 — 默认加密密钥。AWS 物联网 FleetWise 拥有这把钥匙。您无法在 AWS 账户中查看、管理或使用此密钥。您也无法在 AWS CloudTrail 日志中看到对密钥的操作。使用此密钥不会产生额外的费用。
- 客户托管密钥 - 此密钥存储在您的账户中，由您创建、拥有和管理。您对 KMS 密钥拥有完全控制权。需 AWS KMS 支付额外费用。

### AWS 拥有的密钥

AWS 拥有的密钥 未存储在您的账户中。它们是 KMS 密钥集合的一部分，这些密钥 AWS 拥有并管理多个密钥 AWS 账户。AWS 服务 可以 AWS 拥有的密钥 用来保护您的数据。

您无法查看、管理 AWS 拥有的密钥、使用或审核其使用情况。但是无需执行任何操作或更改任何计划即可保护用于加密数据的密钥。

如果您使用，则无需支付任何费用 AWS 拥有的密钥，也不会计入您账户的 AWS KMS 配额。

### 客户管理密钥

客户管理密钥是在您的账户中由您创建、拥有和管理的 KMS 密钥。您可以完全控制这些 KMS 密钥，例如：

- 制定和维护密钥策略、IAM 策略和授权
- 启用和禁用 KMS 密钥
- 轮换 KMS 密钥的加密材料
- 添加 标签
- 创建引用 KMS 密钥的别名
- 安排删除 KMS 密钥

您还可以使用 CloudTrail 和 Amaz CloudWatch on Logs 来跟踪 AWS 物联网 AWS KMS 代表您 FleetWise 发送的请求。

如果您使用的是客户托管密钥，则必须向 AWS 物联网授予对存储在您账户中的 KMS 密钥的 FleetWise 访问权限。AWS 物联网 FleetWise 使用信封加密和密钥层次结构来加密数据。您的 AWS KMS 加密密钥用于加密此密钥层次结构的根密钥。有关更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[信封加密](#)。

以下示例策略授予 AWS IoT 使用您的 AWS KMS 密钥的 FleetWise 权限。

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

在 KMS 密钥策略中添加新部分时，请勿更改策略中的任何现有部分。AWS 如果为物联网 FleetWise 启用了加密，FleetWise 并且存在以下任一情况，AWS 物联网将无法对您的数据执行操作：

- 所提供的 KMS 密钥已被禁用。

- 未为该服务正确配置 KMS 密钥策略。

## 对视觉系统数据使用静态加密

### Note

视觉系统数据目前为预览版，可能会发生变化。

如果您在 AWS 物联网 FleetWise 账户上启用了 AWS KMS 密钥的客户托管加密，并且想要使用视觉系统数据，请重置加密设置以与复杂数据类型兼容。这使 AWS 物联网 FleetWise 能够建立视觉系统数据所需的额外权限。

### Note

如果尚未重置视觉系统数据的加密设置，您的解码器清单可能会停留在验证状态。

1. 使用 [GetEncryptionConfiguration](#) API 操作检查是否启用了 AWS KMS 加密。如果加密类型为 FLEETWISE\_DEFAULT\_ENCRYPTION，则无需采取进一步操作。
2. 如果加密类型为 KMS\_BASED\_ENCRYPTION，则使用 [PutEncryptionConfiguration](#) API 操作将加密类型重置为 FLEETWISE\_DEFAULT\_ENCRYPTION。

```
aws iotfleetwise put-encryption-configuration \  
  --encryption-type FLEETWISE_DEFAULT_ENCRYPTION
```

3. 使用 [PutEncryptionConfiguration](#) API 操作将加密类型重新启用。KMS\_BASED\_ENCRYPTION

```
aws iotfleetwise put-encryption-configuration \  
  --encryption-type KMS_BASED_ENCRYPTION \  
  --kms-key-id kms_key_id
```

有关启用加密的更多信息，请参阅[AWS 物联网中的密钥管理 FleetWise](#)。

## AWS 物联网中的密钥管理 FleetWise

### ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

### AWS 物联网 FleetWise 云密钥管理

默认情况下，AWS 物联网 FleetWise 使用 AWS 托管式密钥来保护您的数据 AWS Cloud。您可以更新设置以使用客户管理的密钥对 AWS IoT 中的数据进行加密 FleetWise。您可以通过 AWS Key Management Service (AWS KMS) 创建、管理和查看您的加密密钥。

AWS IoT FleetWise 支持服务器端加密，客户托管密钥存储在中 AWS KMS 以加密以下资源的数据。

AWS 物联网 FleetWise 资源	数据类型	使用客户托管密钥进行静态加密的字段
信号目录	属性	description、allowedValues、defaultValue、min、max
	Actuator	description、allowedValues、min、max
	传感器	description、allowedValues、min、max
车辆型号 (模型清单)		description
解码器清单		描述
	CanInterface	protocolName、protocolVersion
	ObdInterface	requestMessageId、dtcRequestInterval秒、OBDSStandard

AWS 物联网 FleetWise 资源	数据类型	使用客户托管密钥进行静态加密的字段
		hasTransmissionEcu、Secondard、pidRequestInterval useExtendedIds
	CanSignal	因子 isBigEndian、isSigned、长度、消息 ID、偏移量、startBit
	ObdSignal	字节长度、偏移量、pid、scaling、serviceMode pidResponseLength、startByte、bitMaskLength bitRightShift
车辆		attributes
活动		描述
	conditionBasedCollection计划	表情、minimumTriggerInterval Ms conditionLanguageVersion、TriggerMode
	TimeBasedCollectionScheme	periodMS
状态模板		描述

### Note

其他数据和资源使用由 AWS 物联网管理的密钥使用默认加密进行加密 FleetWise。此密钥已创建并存储在 AWS 物联网 FleetWise 帐户中。

有关更多信息，请参阅[什么是 AWS Key Management Service ?](#) 在《AWS Key Management Service 开发人员指南》中。

使用 KMS 密钥启用加密 (控制台)

要在 AWS 物联网中使用客户托管密钥 FleetWise，您必须更新您的 AWS 物联网 FleetWise 设置。

## 使用 KMS 密钥启用加密 (控制台)

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 导航到设置。
3. 在加密中，选择编辑以打开编辑加密页面。
4. 对于“加密密钥类型”，选择“选择其他 AWS KMS 密钥”。这允许使用存储在 AWS KMS 中的客户托管密钥进行加密。

### Note

您只能对 AWS 物联网 FleetWise 资源使用客户托管密钥加密。这包括信号目录、车辆型号 (模型清单)、解码器清单、车辆、车队和活动。

5. 通过以下选项之一选择 KMS 密钥：
  - 使用现有 KMS 密钥 - 从列表中选择您的 KMS 密钥别名。
  - 要创建新的 KMS 密钥，请选择创建 AWS KMS 密钥。

### Note

这将打开 AWS KMS 控制台。有关创建 KMS 密钥的更多信息，请参阅《AWS Key Management Service 开发人员指南》中的[创建密钥](#)。

6. 选择保存以保存您的设置。

## 使用 KMS 密钥启用加密 (AWS CLI)

您可以使用 [PutEncryptionConfiguration](#) API 操作为您的 AWS 物联网 FleetWise 账户启用加密。以下示例使用 AWS CLI。

要启用加密，请运行以下命令。

- *kms\_key\_id* 替换为 KMS 密钥的 ID。

```
aws iotfleetwise put-encryption-configuration \  
  --encryption-type KMS_BASED_ENCRYPTION \  
  --kms-key-id kms_key_id
```

## Example响应

```
{
  "kmsKeyId": "customer_kms_key_id",
  "encryptionStatus": "PENDING",
  "encryptionType": "KMS_BASED_ENCRYPTION"
}
```

## KMS 密钥策略

创建 KMS 密钥后，您必须至少在 KMS 密钥策略中添加以下语句才能使其与 AWS IoT 配合使用 FleetWise。KMS 密钥策略声明 `iotfleetwise.amazonaws.com` 中的 AWS 物联网 FleetWise 服务主体 FleetWise 允许 AWS 物联网访问 KMS 密钥。

```
{
  "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": "*"
}
```

作为安全最佳实践，请在 KMS 密钥策略中添加密钥 `aws:SourceArn` 并对其进行 `aws:SourceAccount` 条件处理。IAM 全局条件密钥 `aws:SourceArn` 有助于确保 AWS 物联网仅 FleetWise 将 KMS 密钥用于服务特定的资源 Amazon 资源名称 (ARNs)。

如果将值设置为 `aws:SourceArn`，则必须始终设置为 `arn:aws:iotfleetwise:us-east-1:account_id:*`。这允许 KMS 密钥为此访问所有 AWS IoT FleetWise 资源 AWS 账户。AWS IoT FleetWise 支持每个账户为其中的所有资源使用一个 KMS 密钥 AWS 区域。在 ARN 资源字段中 `SourceArn` 使用任何其他值或不使用通配符 (\*)，都会阻止 AWS IoT FleetWise 访问 KMS 密钥。

的值 `aws:SourceAccount` 是您的账户 ID，用于进一步限制 KMS 密钥，使其只能用于您的特定账户。如果您向 KMS 密钥添加密钥 `aws:SourceAccount` 并对其进行 `aws:SourceArn` 条件处理，请确保该密钥未被任何其他服务或账户使用。这有助于避免故障。

以下政策包括服务主体（服务的标识符），以及根据 `aws:SourceAccount` AWS 区域 和您的账户 ID 进行 `aws:SourceArn` 设置以供使用。

```
{
  "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:*"
    }
  }
}
```

有关编辑用于 AWS IoT 的 KMS 密钥策略的更多信息 FleetWise，请参阅 AWS Key Management Service 开发人员指南中的 [更改密钥策略](#)。

#### Important

在 KMS 密钥策略中添加新部分时，请勿更改策略中的任何现有部分。AWS 如果为物联网 FleetWise 启用了加密，FleetWise 并且存在以下任一情况，AWS 物联网将无法对您的数据执行操作：

- 所提供的 KMS 密钥已被禁用。

- 未为该服务正确配置 KMS 密钥策略。

## AWS KMS 加密权限

如果您启用了 AWS KMS 加密，则必须在角色策略中指定权限才能调用 AWS IoT FleetWise APIs。以下策略允许访问所有 AWS IoT FleetWise 操作以及 AWS KMS 特定权限。

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:*",
        "kms:GenerateDataKey*",
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

您的角色需要以下策略声明才能调用加密 APIs。本政策声明允许 PutEncryptionConfiguration 物联网 GetEncryptionConfiguration 采取行动 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": [
  "*"
]
}
]
```

## 删除 AWS KMS 密钥后恢复

如果您在启用 AWS 物联网加密后删除了 AWS KMS 密钥 FleetWise，则必须通过删除所有数据来重置您的帐户，然后才能 FleetWise 再次使用 AWS 物联网。您可以使用列表和删除 API 操作来清理账户中的资源。

### 清理账户中的资源

1. 使用 `listResponseScope` 参数设置为 `list APIs t METADATA_ONLY`。这提供了资源列表，包括资源名称和其他元数据，例如 ARNs 和时间戳。
2. 使用删除 APIs 来移除单个资源。

必须按以下顺序清理资源。

1. 市场活动
  - a. 列出 `listResponseScope` 参数设置为的所有广告系列 `METADATA_ONLY`。
  - b. 删除广告系列。
2. 车队和车辆
  - a. 列出 `listResponseScope` 参数设置为的所有舰队。 `METADATA_ONLY`
  - b. 列出每个车队的所有车辆，`listResponseScope` 参数设置为 `METADATA_ONLY`。
  - c. 解除所有车辆与每个车队的关联。

- d. 删除舰队。
  - e. 删除车辆。
3. 解码器清单
    - a. 列出listResponseScope参数设置为的所有解码器清单。METADATA\_ONLY
    - b. 删除所有解码器清单。
4. 车辆模型 ( 车型清单 )
    - a. 列出listResponseScope参数设置为的所有车辆型号METADATA\_ONLY。
    - b. 删除所有车辆型号。
5. 状态模板
    - a. 列出listResponseScope参数设置为的所有状态模板METADATA\_ONLY。
    - b. 删除所有状态模板。
6. 信号目录
    - a. 列出所有信号目录。
    - b. 删除所有信号目录。

## 使用控制访问权限 AWS IoT FleetWise

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

以下各节介绍如何控制对 AWS IoT FleetWise 资源的访问和访问权限。它们涵盖的信息包括如何授予您的应用程序访问权限，以便 AWS 物联网 FleetWise 可以在活动期间传输车辆数据。它们还描述了如何授予 AWS IoT FleetWise 访问您的 Amazon S3 (S3) 存储桶或 Amazon Timestream 数据库和表的访问权限以存储数据，或者授予用于从车辆发送数据的 MQTT 消息的访问权限。

管理所有这些形式的访问的技术是 AWS Identity and Access Management (IAM)。有关 IAM 的更多信息，请参阅[什么是 IAM ?](#)。

### 内容

- [授 AWS IoT FleetWise 予发送和接收 MQTT 主题数据的权限](#)
- [授予 AWS IoT FleetWise 对 Amazon S3 目的地的访问权限](#)
- [授予 AWS IoT FleetWise 访问亚马逊 Timestream 目的地的权限](#)
- [授 AWS IoT Device Management 予使用以下命令生成有效负载的权限 AWS IoT FleetWise](#)

## 授 AWS IoT FleetWise 予发送和接收 MQTT 主题数据的权限

当您使用 [MQTT 主题](#) 时，您的车辆使用 AWS IoT MQTT 消息代理发送数据。您必须授予订阅您指定的 MQTT 主题的 AWS IoT FleetWise 权限。如果您还使用 AWS IoT 规则采取行动或将数据路由到其他目的地，则必须将策略附加到 IAM 角色以允许将数据转发 AWS IoT FleetWise 到 IoT 规则。

此外，您的其他应用程序或设备可以订阅您指定的主题，以近乎实时地接收车辆数据，并且必须根据需向这些应用程序或设备授予权限和访问权限。

有关使用 MQTT 以及所需角色和权限的更多信息，请参阅：

- [设备通信协议](#)
- [的规则 AWS IoT](#)
- [授予 AWS IoT 规则所需的访问权限](#)
- [传递角色权限](#)

在开始之前，请检查以下事项：

### Important

- 在为 AWS 物联网创建车辆广告系列资源时，必须使用相同的 AWS 区域 FleetWise。如果您切换 AWS 区域，则访问资源可能会遇到问题。
- AWS 美国东部（弗吉尼亚北部）和欧洲（法兰克福）提供物联网 FleetWise。

您可以使用创建具有 MQTT 消息传递信任策略的 IAM 角色。AWS CLI 要创建 IAM 角色，请运行以下命令。

创建具有信任策略的 IAM 角色

- `IotTopicExecutionRole` 替换为您正在创建的角色名称。

- *trust-policy* 替换为包含信任策略的 JSON 文件。

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

创建权限策略，授予 AWS IoT 向您指定的 MQTT 主题发布消息的 FleetWise 权限。要创建权限策略，请运行以下命令。

### 创建权限策略

- *AWSIoT Fleetwise Access Iot Topic Permissions Policy* 替换为您正在创建的策略的名称。
- *permissions-policy* 替换为包含权限策略的 JSON 文件的名称。

```
aws iam create-policy --policy-name AWSIoT FleetwiseAccessIotTopicPermissionsPolicy --  
policy-document file://permissions-policy.json
```

## JSON

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:Publish"  
      ],  
      "Resource": [  
        "topic-arn"  
      ]  
    }  
  ]  
}
```

### 将权限策略附加到 IAM 角色

1. 从输出中复制权限策略的 Amazon 资源名称 (ARN)。
2. 要将 IAM 权限策略附加到您的 IAM 角色，请运行以下命令。
  - *permissions-policy-arn* 替换为您在上一步中复制的 ARN。
  - *IotTopicExecutionRole* 替换为您创建的 IAM 角色的名称。

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-  
name IotTopicExecutionRole
```

有关更多信息，请参阅《IAM 用户指南》中的 [AWS 资源的访问管理](#)。

## 授予 AWS IoT FleetWise 对 Amazon S3 目的地的访问权限

当您使用 Amazon S3 目标时，AWS IoT FleetWise 会将车辆数据传输到您的 S3 存储桶，并且可以选择使用您拥有的 AWS KMS 密钥进行数据加密。如果启用了错误记录，AWS IoT FleetWise 还会将数据传输错误发送到您的 CloudWatch 日志组和流。创建传输流时，您需要拥有 IAM 角色。

AWS IoT FleetWise 使用带有 S3 目标服务主体的存储桶策略。有关添加存储桶策略的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[使用 Amazon S3 控制台添加存储桶策略](#)。

使用以下访问策略启用 AWS IoT FleetWise 对您的 S3 存储桶的访问权限。如果您没有 S3 存储桶，请将 `s3:PutObjectAcl` 添加到 Amazon S3 操作列表中。这会授予存储桶所有者对所交付对象的完全访问权限 AWS IoT FleetWise。有关如何安全访问存储桶中对象的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的[存储桶策略示例](#)。

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"
  }
}
]
```

以下存储桶策略适用于某个 AWS 区域中一个账户中的所有活动。

## 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"
      }
    }
  }
]
}

```

如果您将 KMS 密钥附加到 S3 存储桶，则该密钥将需要以下策略。有关密钥管理的信息，请参阅 Amazon 简单存储服务用户指南中的[使用服务器端 AWS Key Management Service 密钥加密 \(SSE-KMS\) 保护数据](#)。

```

{
  "Version": "2012-10-17",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "key-arn"
}

```

### Important

创建存储桶时，S3 将创建一个默认访问控制列表 (ACL)，以授予资源拥有者对资源的完全控制权限。如果 AWS 物联网 FleetWise 无法将数据传输到 S3，请确保在 S3 存储桶上禁用 ACL。有关更多信息，请参阅 Amazon 简单存储服务用户指南中的[禁用 ACLs 所有新存储桶和强制执行对象所有权](#)。

## 授予 AWS IoT FleetWise 访问亚马逊 Timestream 目的地的权限

当您使用 Timestream 目的地时，会将车辆数据传 AWS IoT FleetWise 送到 Timestream 表。您必须将策略附加到 IAM 角色 AWS IoT FleetWise 才能向 Timestream 发送数据。

如果您使用控制台[创建活动](#)，AWS IoT FleetWise 会自动将所需的策略附加到该角色。

### Note

Amazon Timestream 不在亚太地区（孟买）区域推出。

在开始之前，请检查以下事项：

### Important

- 在为 AWS 物联网 FleetWise 创建 Timestream 资源时，必须使用相同的 AWS 区域。如果您切换 AWS 区域，则在访问 Timestream 资源时可能会遇到问题。
  - AWS 物联网 FleetWise 已在美国东部（弗吉尼亚北部）、欧洲（法兰克福）和亚太地区（孟买）推出。
  - 有关受支持区域的列表，请参见《AWS 一般参考》中的 [Timestream 端点和配额](#)。
- 
- 您必须拥有 Timestream 数据库。有关教程，请参阅《亚马逊 Timestream 开发者指南》中的 [创建数据库](#)。
  - 您必须在指定的 Timestream 数据库中创建表。有关教程，请参阅《亚马逊 Timestream 开发者指南》中的 [创建表](#)。

您可以使用创建带有针对 Timestream 的信任策略的 IAM 角色。AWS CLI 要创建 IAM 角色，请运行以下命令。

创建具有信任策略的 IAM 角色

- *TimestreamExecutionRole* 替换为您正在创建的角色名称。
- *trust-policy* 替换为包含信任策略的 .json 文件。

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

创建权限策略，向 AWS 物联网授予将数据写入 Timestream 的 FleetWise 权限。要创建权限策略，请运行以下命令。

## 创建权限策略

- *AWSIoT Fleetwise Access Timestream Permissions Policy* 替换为您正在创建的策略的名称。
- *permissions-policy* 替换为包含权限策略的 JSON 文件的名称。

```
aws iam create-policy --policy-name AWSIoT Fleetwise Access Timestream Permissions Policy --
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": "*"
    }
  ]
}
```

## 将权限策略附加到 IAM 角色

1. 从输出中复制权限策略的 Amazon 资源名称 (ARN)。
2. 要将 IAM 权限策略附加到您的 IAM 角色，请运行以下命令。
  - *permissions-policy-arn* 替换为您在上一步中复制的 ARN。
  - *TimestreamExecutionRole* 替换为您创建的 IAM 角色的名称。

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-  
name TimestreamExecutionRole
```

有关更多信息，请参阅《IAM 用户指南》中的 [AWS 资源的访问管理](#)。

## 授 AWS IoT Device Management 予使用以下命令生成有效负载的权限 AWS IoT FleetWise

当您使用命令功能开始执行命令时，AWS IoT Device Management 将从传入的请求中获取命令和命令参数。然后，它需要访问 AWS 物联网 FleetWise 资源的权限来验证请求并生成有效负载。然后，通过 AWS IoT Device Management MQTT 将有效载荷发送到您的车辆已订阅的命令请求主题。

您必须先创建一个 IAM 角色来授予 AWS IoT Device Management 生成有效负载所需的权限。然后，使用字段向 [CreateCommandAPI](#) 提供此角色的 ARN。roleArn 以下是一些策略示例。

### Important

对于 IAM 角色，您必须使用与创建车辆和指挥资源时 AWS 区域 相同的角色。如果您切换 AWS 区域，则访问资源可能会遇到问题。

IAM 角色需要具有以下信任策略。

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "RemoteCommandsTrustPolicy",
      "Effect": "Allow",
      "Principal": {
        "Service": "iot.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

向所有车辆（物联网事物）授予权限

以下示例说明如何授予为所有注册为 AWS IoT 事物的车辆生成有效载荷的权限。

**Note**

- 此政策可能过于宽松。使用最小权限原则，确保您只授予必要的权限。
- 要改为拒绝权限，"Effect": "Allow"请"Effect": "Deny"在 IAM 策略中更改为。

在此示例中：

- *region*以及你 AWS 区域 在哪里使用物 AWS 联网 FleetWise 资源。
- *111122223333*用你的 AWS 账户 电话号码。

JSON

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

向特定车辆授予权限 ( 物联网事物 )

以下示例说明如何授予为注册为 AWS IoT 事物的特定车辆生成有效载荷的权限。

在此示例中：

- *region*以及你 AWS 区域 在哪里使用物 AWS 联网 FleetWise 资源。
- *111122223333*用你的 AWS 账户 电话号码。
- *<VEHICLE\_NAME>*使用您的车辆的物联网事物名称。

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iotfleetwise:GenerateCommandPayload",
      "Resource": "arn:aws:iot:us-east-1:111122223333:thing/<VEHICLE_NAME>"
    }
  ]
}
```

向特定车辆和信号授予权限

以下示例说明如何授予为特定车辆的执行器生成有效载荷的权限。

在此示例中：

- *region* 以及你 AWS 区域 在哪里使用物 AWS 联网 FleetWise 资源。
- *111122223333* 用你的 AWS 账户 电话号码。
- *<VEHICLE\_NAME>* 使用您的车辆的物联网事物名称。
- *<SIGNAL\_FQN>* 使用信号的名称，例如 *<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": [
```

```

    "SignalFqn": "<SIGNAL_FQN>"
  }
}
]
}

```

## 向特定车辆和状态模板授予权限

以下示例说明如何授予为特定车辆和状态模板生成有效载荷的权限。

在此示例中：

- *region*是您使用 AWS 物联网 FleetWise 资源 AWS 区域的地方。
- *111122223333*是你的 AWS 账户 电话号码。
- *<VEHICLE\_NAME>*是您车辆的物联网事物名称。
- *<STATE\_TEMPLATE\_ID>*使用您的状态模板的标识符。

## 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>"
      ]
    }
  ]
}

```

## 授予使用客户托管的 KMS 密钥的权限

如果您已为启用客户托管 KMS 密钥 AWS IoT FleetWise，则以下示例说明如何授予生成有效负载的权限。

在此示例中：

- *region* 以及你 AWS 区域 在哪里使用物 AWS 联网 FleetWise 资源。
- *111122223333* 用你的 AWS 账户 电话号码。
- *<KMS\_KEY\_ID>* 使用您的 KMS 密钥的 ID。

### 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>"
    }
  ]
}
```

## 适用于 AWS 物联网的 Identity and Access 管理 FleetWise

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 管理员控制谁可以进行身份验证（登录）和授权（有权限）使用 AWS 物联网 FleetWise 资源。您可以使用 IAM AWS 服务，无需支付额外费用。

### 主题

- [受众](#)

- [使用身份进行身份验证](#)
- [使用策略管理访问](#)
- [AWS 物联网如何 FleetWise 与 IAM 配合使用](#)
- [基于身份的物联网策略示例 AWS FleetWise](#)
- [对 AWS 物联网 FleetWise 身份和访问进行故障排除](#)
- [AWS 物联网 FleetWise API 操作和资源参考](#)

## 受众

您的使用方式 AWS Identity and Access Management (IAM) 因您的角色而异：

- 服务用户：如果您无法访问功能，请从管理员处请求权限（请参阅[对 AWS 物联网 FleetWise 身份和访问进行故障排除](#)）
- 服务管理员：确定用户访问权限并提交权限请求（请参阅[AWS 物联网如何 FleetWise 与 IAM 配合使用](#)）
- IAM 管理员：编写用于管理访问权限的策略（请参阅[基于身份的物联网策略示例 AWS FleetWise](#)）

## 使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 AWS 账户根用户，或者通过担任 IAM 角色进行身份验证。

您可以使用来自身份源的证书 AWS IAM Identity Center（例如（IAM Identity Center）、单点登录身份验证或 Google/Facebook 证书，以联合身份登录。有关登录的更多信息，请参阅《AWS 登录用户指南》中的[如何登录您的 AWS 账户](#)。

对于编程访问，AWS 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的[适用于 API 请求的 AWS 签名版本 4](#)。

## AWS 账户 root 用户

创建时 AWS 账户，首先会有一个名为 AWS 账户 root 用户的登录身份，该身份可以完全访问所有资源 AWS 服务和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的[需要根用户凭证的任务](#)。

## 联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 AWS 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Directory Service ，或者 AWS 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

要集中管理访问权限，建议使用。AWS IAM Identity Center 有关更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[什么是 IAM Identity Center ?](#)。

## IAM 用户和群组

[IAM 用户](#)是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的[要求人类用户使用身份提供商的联合身份验证才能 AWS 使用临时证书进行访问](#)。

[IAM 组](#)指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的[IAM 用户使用案例](#)。

## IAM 角色

[IAM 角色](#)是具有特定权限的身份，可提供临时凭证。您可以通过[从用户切换到 IAM 角色 \(控制台\)](#)或调用 AWS CLI 或 AWS API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的[担任角色的方法](#)。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon EC2 上运行的应用程序非常有用。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

## 使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。AWS 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的[JSON 策略概述](#)。

管理员使用策略，通过定义哪个主体可以在什么条件下对哪些资源执行哪些操作来指定谁有权访问什么。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以担任这些角色。IAM 策略定义权限，与执行操作所用的方法无关。

## 基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可以执行什么操作、对哪些资源执行以及在什么条件下执行。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管策略与内联策略之间进行选择](#)。

## 基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。您必须在基于资源的策略中[指定主体](#)。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

## 其他策略类型

AWS 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 AWS Organizations。有关更多信息，请参阅《AWS Organizations 用户指南》中的[服务控制策略](#)。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《AWS Organizations 用户指南》中的[资源控制策略 \(RCPs\)](#)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的[会话策略](#)。

## 多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 AWS 确定是否允许请求，请参阅 IAM 用户指南中的[策略评估逻辑](#)。

## AWS 物联网如何 FleetWise 与 IAM 配合使用

在使用 IAM 管理 AWS 物联网访问权限之前 FleetWise，请先了解有哪些 IAM 功能可用于 AWS 物联网 FleetWise。

可在物联网中使用的 AWS IAM 功能 FleetWise

IAM 功能	AWS 物联网 FleetWise 支持
<a href="#">基于身份的策略</a>	是
<a href="#">基于资源的策略</a>	否
<a href="#">策略操作</a>	是
<a href="#">策略资源</a>	是
<a href="#">策略条件键</a>	是
<a href="#">ACLs</a>	否
<a href="#">ABAC (策略中的标签)</a>	部分
<a href="#">临时凭证</a>	是
<a href="#">主体权限</a>	是
<a href="#">服务角色</a>	否
<a href="#">服务关联角色</a>	否

要全面了解 AWS 物联网和其他 AWS 服务如何 FleetWise 与大多数 IAM 功能配合使用，请参阅 IAM 用户指南中与 IAM [配合使用的AWS 服务](#)。

### 基于身份的物联网策略 AWS FleetWise

支持基于身份的策略：是

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的[使用客户管理型策略定义自定义 IAM 权限](#)。

通过使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源以及允许或拒绝操作的条件。要了解可在 JSON 策略中使用的所有元素，请参阅《IAM 用户指南》中的[IAM JSON 策略元素引用](#)。

## 基于身份的物联网策略示例 AWS FleetWise

要查看 FleetWise 基于 AWS IoT 身份的策略示例，请参阅。[基于身份的物联网策略示例 AWS FleetWise](#)

## 物联网内部 AWS 基于资源的策略 FleetWise

支持基于资源的策略：否

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Amazon S3 存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中[指定主体](#)。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的主体。有关更多信息，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

## AWS 物联网的政策行动 FleetWise

支持策略操作：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

JSON 策略的 Action 元素描述可用于在策略中允许或拒绝访问的操作。在策略中包含操作以授予执行关联操作的权限。

要查看 AWS IoT FleetWise 操作列表，请参阅《服务授权参考》FleetWise 中的[AWS IoT 定义的操作](#)。

AWS IoT 中的策略操作在操作前 FleetWise 使用以下前缀：

```
iotfleetwise
```

要在单个语句中指定多项操作，请使用逗号将它们隔开。

```
"Action": [  
    "iotfleetwise:action1",  
    "iotfleetwise:action2"  
]
```

您也可以使用通配符 ( \* ) 指定多个操作。例如，要指定以单词 List 开头的所有操作，包括以下操作：

```
"Action": "iotfleetwise:List*"
```

要查看 FleetWise 基于 AWS IoT 身份的策略示例，请参阅 [基于身份的物联网策略示例 AWS FleetWise](#)

## AWS 物联网政策资源 FleetWise

支持策略资源：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Resource JSON 策略元素指定要向其应用操作的一个或多个对象。作为最佳实践，请使用其 [Amazon 资源名称 \( ARN \)](#) 指定资源。对于不支持资源级权限的操作，请使用通配符 ( \* ) 指示语句应用于所有资源。

```
"Resource": "*"
```

要查看 AWS IoT FleetWise 资源类型及其列表 ARNs，请参阅《服务授权参考》FleetWise 中的 [AWS IoT 定义的资源](#)。要了解您可以使用哪些操作来指定每种资源的 ARN，请参阅 [IoT AWS 定义的操作](#)。FleetWise

要查看 FleetWise 基于 AWS IoT 身份的策略示例，请参阅 [基于身份的物联网策略示例 AWS FleetWise](#)

## AWS 物联网的策略条件密钥 FleetWise

支持特定于服务的策略条件键：是

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

Condition 元素根据定义的条件指定语句何时执行。您可以创建使用[条件运算符](#)（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的[AWS 全局条件上下文密钥](#)。

要查看 AWS IoT FleetWise 条件键列表，请参阅《服务授权参考》FleetWise 中的[AWS IoT 条件密钥](#)。要了解可以使用条件键的操作和资源，请参阅[AWS IoT 定义的操作 FleetWise](#)。

要查看 FleetWise 基于 AWS IoT 身份的策略示例，请参阅。[基于身份的物联网策略示例 AWS FleetWise](#)

## AWS 物联网中的访问控制列表 (ACLs) FleetWise

支持 ACLs：否

访问控制列表 (ACLs) 控制哪些委托人（账户成员、用户或角色）有权访问资源。ACLs 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

## 使用物联网实现基于属性的访问控制 (ABAC) AWS FleetWise

支持 ABAC（策略中的标签）：部分支持

基于属性的访问权限控制 (ABAC) 是一种授权策略，该策略基于称为标签的属性来定义权限。您可以将标签附加到 IAM 实体和 AWS 资源，然后设计 ABAC 策略以允许在委托人的标签与资源上的标签匹配时进行操作。

要基于标签控制访问，您需要使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的[条件元素](#)中提供标签信息。

如果某个服务对于每种资源类型都支持所有这三个条件键，则对于该服务，该值为是。如果某个服务仅对于部分资源类型支持所有这三个条件键，则该值为部分。

有关 ABAC 的更多信息，请参阅《IAM 用户指南》中的[使用 ABAC 授权定义权限](#)。要查看设置 ABAC 步骤的教程，请参阅《IAM 用户指南》中的[使用基于属性的访问权限控制 \(ABAC\)](#)。

**Note**

AWS FleetWise 仅支持物联网 iam:PassRole ，这是 CreateCampaign API 操作所必需的。

## 在 AWS IoT 中使用临时证书 FleetWise

支持临时凭证：是

临时证书提供对 AWS 资源的短期访问权限，并且是在您使用联合身份或切换角色时自动创建的。AWS 建议您动态生成临时证书，而不是使用长期访问密钥。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 中的临时安全凭证](#) 和 [使用 IAM 的 AWS 服务](#)

## IoT 的跨服务主体 AWS 权限 FleetWise

支持转发访问会话 ( FAS )：是

转发访问会话 (FAS) 使用调用主体的权限 AWS 服务，再加上 AWS 服务 向下游服务发出请求的请求。有关发出 FAS 请求时的策略详情，请参阅 [转发访问会话](#)。

## AWS 物联网的服务角色 FleetWise

支持服务角色：否

服务角色是由一项服务担任、代表您执行操作的 [IAM 角色](#)。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的 [创建向 AWS 服务委派权限的角色](#)。

**Warning**

更改服务角色的权限可能会中断 AWS IoT FleetWise 功能。仅当 AWS IoT FleetWise 提供相关指导时才编辑服务角色。

## 物联网的 AWS 服务相关角色 FleetWise

支持服务相关角色：否

服务相关角色是一种与服务相关联的 AWS 服务服务角色。服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务关联角色的权限。

有关创建或管理服务相关角色的详细信息，请参阅[能够与 IAM 搭配使用的AWS 服务](#)。在表中查找服务相关角色列中包含 Yes 的表。选择是链接以查看该服务的服务相关角色文档。

## 在 AWS IoT 中使用服务相关角色 FleetWise

AWS IoT FleetWise 使用 AWS Identity and Access Management (IAM) [服务相关角色](#)。服务相关角色是一种与 AWS IoT 直接关联的独特的 IAM 角色。服务相关角色由 AWS IoT 预定义 FleetWise，包括 AWS IoT 向亚马逊 CloudWatch 发送指标 FleetWise 所需的权限。有关更多信息，请参阅 [FleetWise 使用 Amazon 监控 AWS 物联网 CloudWatch](#)。

服务相关角色可以更快速地为 FleetWise 设置 AWS IoT，因为您不必手动添加必要的权限。AWS IoT FleetWise 定义了其服务相关角色的权限，除非另有定义，否则只有 AWS IoT FleetWise 可以担任其角色。定义的权限包括信任策略和权限策略。该权限策略不能附加到任何其他 IAM 实体。

只有在首先删除相关资源后，您才能删除服务关联角色。这样可以保护您的 AWS IoT FleetWise 资源，因为您不会无意中移除访问这些资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅与 [IAM 配合使用的AWS 服务](#)，并在服务相关角色列中查找标有“是”的服务。要查看该服务的服务相关角色文档，请选择带有链接的是。

### AWS IoT 的服务相关角色权限 FleetWise

AWS IoT FleetWise 使用名为 `AWSServiceRoleForIoTFleetWise` 的服务相关角色，这是一项 AWS 托管策略，用于获得 AWS IoT FleetWise 的所有 out-of-the-box 权限。

`AWSServiceRoleForIoTFleetWise` 服务相关角色信任以下服务来代替该角色：

- `IoTFleetWise`

名为的角色权限策略 [AWSIoTFleetWiseServiceRolePolicy](#) 允许 AWS IoT FleetWise 对指定资源完成以下操作：

- 针对资源 \* 的操作 `cloudwatch:PutMetricData`

有关本策略变更的信息，请参阅 [AWSIoTFleetWiseServiceRolePolicy 政策更新](#)。

服务相关角色有权将指标发布到以下 CloudWatch 命名空间：

- `AWS/IoTFleetWise`— 用于特定于服务的指标

- AWS/Usage— 有关使用情况指标

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务关联角色。有关更多信息，请参阅《IAM 用户指南》中的[服务关联角色权限](#)。

### 为 AWS IoT 创建服务相关角色 FleetWise

您无需手动创建服务关联角色。当您在 AWS IoT FleetWise 控制台、AWS CLI、或 AWS API 中注册账户时，AWS IoT FleetWise 会为您创建服务相关角色。有关更多信息，请参阅[配置您的 AWS 物联网 FleetWise 设置](#)。

### 在 AWS IoT 中创建服务相关角色 FleetWise（控制台）

您无需手动创建服务关联角色。当您在 AWS IoT FleetWise 控制台、CLI 或 AWS AP AWS I 中注册账户时，AWS IoT FleetWise 会为您创建服务相关角色。

### 编辑 AWS IoT 的服务相关角色 FleetWise

您无法在 AWS IoT FleetWise 中编辑 W AWSService RoleForlo TFleet ise 服务相关角色。可能有多个实体引用您创建的服务相关角色，因此无法更改角色名称。但是可以使用 IAM 编辑角色说明。有关更多信息，请参阅《IAM 用户指南》中的[编辑服务相关角色](#)。

### 清除服务相关角色

必须先删除服务相关角色使用的所有资源，然后才能使用 IAM 删除该角色。

#### Note

如果您尝试删除资源时 AWS IoT FleetWise 正在使用该角色，则删除可能会失败。如果发生这种情况，请等待几分钟后重试。要了解如何 service-linked-role 通过控制台、AWS CLI 或 AWS API 删除，请参阅 IAM 用户指南中的[使用服务相关角色](#)。

如果您删除了此服务相关角色，然后需要重新创建它，则可以在 AWS IoT FleetWise 中注册一个账户。FleetWise 然后，AWS IoT 会再次为您创建服务相关角色。

## 基于身份的物联网策略示例 AWS FleetWise

默认情况下，用户和角色无权创建或修改 AWS IoT FleetWise 资源。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。

要了解如何使用这些示例 JSON 策略文档创建基于 IAM 身份的策略，请参阅《IAM 用户指南》中的[创建 IAM 策略 \(控制台\)](#)。

有关 AWS 物联网 FleetWise 定义的操作和资源类型 (包括每种资源类型的格式) 的详细信息，请参阅《服务授权参考》FleetWise 中的[AWS IoT 操作、资源和条件密钥](#)。ARNs

## 主题

- [策略最佳实践](#)
- [使用 AWS 物联网 FleetWise 控制台](#)
- [允许用户查看他们自己的权限](#)
- [访问 Amazon Timestream 中的资源](#)

## 策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS 物联网 FleetWise 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下指南和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的 AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的[AWS 托管策略或工作职能的 AWS 托管策略](#)。
- 应用最低权限：在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的[IAM 中的策略和权限](#)。
- 使用 IAM 策略中的条件进一步限制访问权限：您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的[IAM JSON 策略元素：条件](#)。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性：IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的[使用 IAM Access Analyzer 验证策略](#)。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的[使用 MFA 保护 API 访问](#)。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 [IAM 中的安全最佳实践](#)。

## 使用 AWS 物联网 FleetWise 控制台

要访问 AWS 物联网 FleetWise 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关 AWS 物联网 FleetWise 资源的详细信息 AWS 账户。如果创建比必需的最低权限更为严格的基于身份的策略，对于附加了该策略的实体（用户或角色），控制台将无法按预期正常运行。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与其尝试执行的 API 操作相匹配的操作。

为确保用户和角色仍然可以使用 AWS 物联网 FleetWise 控制台，还需要将 AWS 物联网 FleetWise ConsoleAccess 或 ReadOnly AWS 托管策略附加到实体。有关更多信息，请参阅《IAM 用户指南》中的 [为用户添加权限](#)。

## 允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",

```

```
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

## 访问 Amazon Timestream 中的资源

在使用 AWS 物联网之前 FleetWise，您必须注册您的 AWS 账户、IAM 和 Amazon Timestream 资源，以 FleetWise 授予 AWS 物联网 AWS Cloud 代表您向其发送车辆数据的权限。要注册，您需要：

- Amazon Timestream 数据库。
- 在指定的 Amazon Timestream 数据库中创建表。
- 一个 IAM 角色，允许 AWS 物联网 FleetWise 向 Amazon Timestream 发送数据。

有关更多信息，包括过程和策略示例，请参阅[配置您的 AWS 物联网 FleetWise 设置](#)。

## 对 AWS 物联网 FleetWise 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS IoT FleetWise 和 IAM 时可能遇到的常见问题。

### 主题

- [我无权在 AWS IoT 中执行操作 FleetWise](#)
- [我无权执行 iam : PassRole](#)
- [我想允许我以外的人 AWS 账户 访问我的 AWS IoT FleetWise 资源](#)

### 我无权在 AWS IoT 中执行操作 FleetWise

如果 AWS 管理控制台 告诉您您无权执行某项操作，则必须联系管理员寻求帮助。管理员是向您提供登录凭证的人。

当 mateojackson IAM 用户尝试使用控制台查看有关虚构 *myVehicle* 资源的详细信息，但不拥有 `iotfleetwise:GetVehicleStatus` 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotfleetwise:GetVehicleStatus on resource: myVehicle
```

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `iotfleetwise:GetVehicleStatus` 操作访问 `myVehicle` 资源。

## 我无权执行 iam : PassRole

如果您收到错误消息，提示您无权执行 `iam:PassRole` 操作，则必须更新您的策略以允许您将角色传递给 AWS IoT FleetWise。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为的 IAM 用户 `marymajor` 尝试使用控制台在 AWS IoT 中执行操作时，会出现以下示例错误 FleetWise。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

## 我想允许我以外的人 AWS 账户 访问我的 AWS IoT FleetWise 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以代入角色。对于支持基于资源的策略或访问控制列表 (ACLs) 的服务，您可以使用这些策略向人们授予访问您的资源的权限。

要了解更多信息，请参阅以下内容：

- 要了解 AWS IoT 是否 FleetWise 支持这些功能，请参阅 [AWS 物联网如何 FleetWise 与 IAM 配合使用](#)。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户，请参阅 [IAM 用户指南中的向您拥有 AWS 账户的另一个 IAM 用户提供访问权限](#)。
- 要了解如何向第三方提供对您的资源的访问 [权限 AWS 账户](#)，请参阅 [IAM 用户指南中的向第三方提供访问权限](#)。AWS 账户

- 要了解如何通过身份联合验证提供访问权限，请参阅《IAM 用户指南》中的[为经过外部身份验证的用户（身份联合验证）提供访问权限](#)。
- 要了解使用角色和基于资源的策略进行跨账户访问之间的差别，请参阅《IAM 用户指南》中的[IAM 中的跨账户资源访问](#)。

## AWS 物联网 FleetWise API 操作和资源参考

当您使用 [IAM 身份的策略管理访问权限](#) 时，您可以使用下表作为参考。该表列出了每个 FleetWise I AWS oT API、您可以为其授予执行操作权限的相应操作，以及您可以为其授予权限的 AWS 资源。

在策略的 Action 字段中指定操作，并在策略的 Resource 字段中指定资源值。要指定操作，请使用 `iotfleetwise:` 前缀和操作名称。例如 `iotfleetwise:CreateSignalCatalog`。

目前，AWS IoT FleetWise 支持以下 [操作、资源和条件键](#)。

使用滚动条查看表格的其余部分。

AWS 物联网 FleetWise 行动	所需的权限	资源
AssociateVehicleFleet	iotfleetwise : AssociateVehicleFleet	arn:: iotfleetwise::: fleet/ <i>partition region account_id fleet-id</i>  arn:: iotfleetwise::: vehicle/ <i>partition region account_id vehicle-id</i>
CreateCampaign	iotfleetwise : CreateCampaign	arn:: iotfleetwise::: <i>partition</i> campaign/ <i>region account_id campaign-name</i>  arn:: iotfleetwise::: <i>partition</i> signal-catalog/ <i>region account_id name</i>  arn:: iotfleetwise::: vehicle/ <i>partition region account_id vehicle-id</i>  arn:: iotfleetwise::: fleet/ <i>partition region account_id fleet-id</i>

AWS 物联网 FleetWise 行动	所需的权限	资源
CreateDecoderManifest	iotfleetwise : CreateDecoderManifest	arn:: iotfleetwise:: <i>partition</i> decoder-manifest/ <i>region account_id name</i>  arn:: iotfleetwise:: model-m <i>partition</i> manifest/ <i>region account_id name</i>
CreateFleet	iotfleetwise : CreateFleet	arn:: iotfleetwise:: fleet/ <i>partition region account_id fleet-id</i>  arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
CreateModelManifest	iotfleetwise : CreateModelManifest	arn:: iotfleetwise:: model-m <i>partition</i> manifest/ <i>region account_id name</i>  arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
CreateSignalCatalog	iotfleetwise : CreateSignalCatalog	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
CreateStateTemplate	iotfleetwise : CreateStateTemplate	arn:: iotfleetwise:: state- <i>partition</i> template/ <i>region account_id state-template-id</i>  arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
CreateVehicle	iotfleetwise : CreateVehicle	arn:: iotfleetwise:: vehicle/ <i>partition region account_id vehicle-id</i>  arn:: iotfleetwise:: model-m <i>partition</i> manifest/ <i>region account_id name</i>  arn:: iotfleetwise:: <i>partition</i> decoder-manifest/ <i>region account_id name</i>

AWS 物联网 FleetWise 行动	所需的权限	资源
BatchCreateVehicle	iotfleetwise : CreateVehicle	arn:: iotfleetwise:: vehicle/ <i>partition</i> <i>region</i> <i>account_id</i> <i>vehicle-id</i>  arn:: iotfleetwise:: model-m <i>partition</i> anifest/ <i>region</i> <i>account_id</i> <i>name</i>  arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region</i> <i>account_id</i> <i>name</i>
DeleteCampaign	iotfleetwise : DeleteCampaign	arn:: iotfleetwise:: <i>partition</i> campaign/ <i>region</i> <i>account_id</i> <i>campaign-name</i>
DeleteDecoderManifest	iotfleetwise : DeleteDecoderManifest	arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region</i> <i>account_id</i> <i>name</i>
DeleteFleet	iotfleetwise : DeleteFleet	arn:: iotfleetwise:: fleet/ <i>partition</i> <i>region</i> <i>account_id</i> <i>fleet-id</i>
DeleteModelManifest	iotfleetwise : DeleteModelManifest	arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region</i> <i>account_id</i> <i>name</i>
DeleteSignalCatalog	iotfleetwise : DeleteSignalCatalog	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region</i> <i>account_id</i> <i>name</i>
DeleteStateTemplate	iotfleetwise : DeleteStateTemplate	arn:: iotfleetwise:: state- <i>partition</i> template/ <i>region</i> <i>account_id</i> <i>state-tem</i> <i>plate-id</i>
DeleteVehicle	iotfleetwise : DeleteVehicle	arn:: iotfleetwise:: vehicle/ <i>partition</i> <i>region</i> <i>account_id</i> <i>vehicle-id</i>
DisassociateVehicleFleet	iotfleetwise : DisassociateVehicleFleet	arn:: iotfleetwise:: vehicle/ <i>partition</i> <i>region</i> <i>account_id</i> <i>vehicle-id</i>  arn:: iotfleetwise:: fleet/ <i>partition</i> <i>region</i> <i>account_id</i> <i>fleet-id</i>

AWS 物联网 FleetWise 行动	所需的权限	资源
GetCampaign	iotfleetwise : GetCampaign	arn:: iotfleetwise:: <i>partition</i> campaign/ <i>region account_id campaign-name</i>
GetDecoderManifest	iotfleetwise : GetDecoderManifest	arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region account_id name</i>
GetEncryptionConfiguration	iotfleetwise : GetEncryptionConfiguration	
GetFleet	iotfleetwise : GetFleet	arn:: iotfleetwise:: fleet/ <i>partition region account_id fleet-id</i>
GetLoggingOptions	iotfleetwise : GetLoggingOptions	
GetModelManifest	iotfleetwise : GetModelManifest	arn:: iotfleetwise:: model-m <i>partition</i> anifest/ <i>region account_id name</i>
GetRegisterAccountStatus	iotfleetwise : GetRegisterAccountStatus	
GetSignalCatalog	iotfleetwise : GetSignalCatalog	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
GetStateTemplate	iotfleetwise : GetStateTemplate	arn:: iotfleetwise:: state- <i>partition</i> template/ <i>region account_id state-tem plate-id</i>
GetVehicle	iotfleetwise : GetVehicle	arn:: iotfleetwise:: vehicle/ <i>partition region account_id vehicle-id</i>
GetVehicleStatus	iotfleetwise : GetVehicleStatus	arn:: iotfleetwise:: vehicle/ <i>partition region account_id vehicle-id</i>
ImportDecoderManifest	iotfleetwise : ImportDecoderManifest	arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region account_id name</i>

AWS 物联网 FleetWise 行动	所需的权限	资源
ImportSignalCatalog	iotfleetwise : ImportSignalCatalog	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
ListCampaigns	iotfleetwise : ListCampaigns	
ListDecoderManifestNetworkInterfaces	iotfleetwise : ListDecoderManifestNetworkInterfaces	arn:: iotfleetwise:: <i>partition</i> decoder-manifest/ <i>region account_id name</i>
ListDecoderManifests	iotfleetwise : ListDecoderManifests	
ListDecoderManifestSignals	iotfleetwise : ListDecoderManifestSignals	arn:: iotfleetwise:: <i>partition</i> decoder-manifest/ <i>region account_id name</i>
ListFleets	iotfleetwise : ListFleets	
ListFleetsForVehicle	iotfleetwise : ListFleetsForVehicle	arn:: iotfleetwise:: vehicle/ <i>partition region account_id vehicle-id</i>
ListModelManifestNodes	iotfleetwise : ListModelManifestNodes	arn:: iotfleetwise:: model-m <i>partition</i> manifest/ <i>region account_id name</i>
ListModelManifests	iotfleetwise : ListModelManifests	
ListSignalCatalogNodes	iotfleetwise : ListSignalCatalogNodes	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>
ListSignalCatalogs	iotfleetwise : ListSignalCatalogs	

AWS 物联网 FleetWise 行动	所需的权限	资源
ListStateTemplates	iotfleetwise : ListStateTemplates	
ListVehicles	iotfleetwise : ListVehicles	
ListVehiclesInFleet	iotfleetwise : ListVehiclesInFleet	arn:: iotfleetwise:: fleet/ <i>partition</i> <i>region</i> <i>account_id</i> <i>fleet-id</i>
ListTagsForResource	iotfleetwise : ListTagsForResource	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region</i> <i>account_id</i> <i>name</i>  arn:: iotfleetwise:: model-m <i>partition</i> manifest/ <i>region</i> <i>account_id</i> <i>name</i>  arn:: iotfleetwise:: <i>partition</i> decoder-manifest/ <i>region</i> <i>account_id</i> <i>name</i>  arn:: iotfleetwise:: vehicle/ <i>partition</i> <i>region</i> <i>account_id</i> <i>vehicle-id</i>  arn:: iotfleetwise:: fleet/ <i>partition</i> <i>region</i> <i>account_id</i> <i>fleet-id</i>  arn:: iotfleetwise:: <i>partition</i> campaign/ <i>region</i> <i>account_id</i> <i>campaign-name</i>  arn:: iotfleetwise:: state- <i>partition</i> template/ <i>region</i> <i>account_id</i> <i>state-template-id</i>
PutEncryptionConfiguration	iotfleetwise : PutEncryptionConfiguration	
PutLoggingOptions	iotfleetwise : PutLoggingOptions	

AWS 物联网 FleetWise 行动	所需的权限	资源
RegisterAccount	iotfleetwise : RegisterAccount	
TagResource	iotfleetwise : TagResource	<p>arn:: iotfleetwise:: <i>partition</i> signal-catalog/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: model-m <i>partition</i> anifest/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: <i>partition</i> decoder-manifest/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: vehicle/<i>partition region account_id vehicle-id</i></p> <p>arn:: iotfleetwise:: fleet/<i>partition region account_id fleet-id</i></p> <p>arn:: iotfleetwise:: <i>partition</i> campaign/<i>region account_id campaign-name</i></p> <p>arn:: iotfleetwise:: state-<i>partition</i> template/<i>region account_id state-template-id</i></p>

AWS 物联网 FleetWise 行动	所需的权限	资源
UntagResource	iotfleetwise : UntagResource	<p>arn:: iotfleetwise:: <i>partition</i> signal-catalog/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: model-m <i>partition</i> anifest/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/<i>region account_id name</i></p> <p>arn:: iotfleetwise:: vehicle/<i>partition region account_id vehicle-id</i></p> <p>arn:: iotfleetwise:: fleet/<i>partition region account_id fleet-id</i></p> <p>arn:: iotfleetwise:: <i>partition</i> campaign/<i>region account_id campaign-name</i></p> <p>arn:: iotfleetwise:: state-<i>partition</i> template/<i>region account_id state-template-id</i></p>
UpdateCampaign	iotfleetwise : UpdateCampaign	arn:: iotfleetwise:: <i>partition</i> campaign/ <i>region account_id campaign-name</i>
UpdateDecoderManifest	iotfleetwise : UpdateDecoderManifest	arn:: iotfleetwise:: <i>partition</i> decoder-m anifest/ <i>region account_id name</i>
UpdateFleet	iotfleetwise : UpdateFleet	arn:: iotfleetwise:: fleet/ <i>partition region account_id fleet-id</i>
UpdateModelManifest	iotfleetwise : UpdateModelManifest	arn:: iotfleetwise:: model-m <i>partition</i> anifest/ <i>region account_id name</i>
UpdateSignalCatalog	iotfleetwise : UpdateSignalCatalog	arn:: iotfleetwise:: <i>partition</i> signal-catalog/ <i>region account_id name</i>

AWS 物联网 FleetWise 行动	所需的权限	资源
UpdateStateTemplate	iotfleetwise : UpdateStateTemplate	arn:: iotfleetwise::: state- <i>partition</i> template/ <i>region account_id state-template-id</i>
UpdateVehicle	iotfleetwise : UpdateVehicle	arn:: iotfleetwise::: vehicle/ <i>partition region account_id vehicle-id</i>  arn:: iotfleetwise::: model-m <i>partition</i> anifest/ <i>region account_id name</i>  arn:: iotfleetwise::: <i>partition</i> decoder-m anifest/ <i>region account_id name</i>
BatchUpdateVehicle	iotfleetwise : UpdateVehicle	arn:: iotfleetwise::: vehicle/ <i>partition region account_id vehicle-id</i>  arn:: iotfleetwise::: model-m <i>partition</i> anifest/ <i>region account_id name</i>  arn:: iotfleetwise::: <i>partition</i> decoder-m anifest/ <i>region account_id name</i>

## AWS AWS IoT 的托管策略更新 FleetWise

查看 FleetWise 自该服务开始跟踪这些更改以来 AWS IoT AWS 托管策略更新的详细信息。要获取有关此页面变更的自动提醒，请在 AWS IoT FleetWise 文档历史记录页面上订阅 RSS 提要。

### AWSIoT FleetwiseServiceRolePolicy

此策略允许 AWS IoT FleetWise CloudWatch 代表您向亚马逊发布指标。

更改	描述	日期
AWSIoT FleetWiseServiceRolePolicy : 对现有策略的更新	除了现有的AWS/IoTFleetWise 命名空间权限外，还增加了向 AWS/Usage 命名空间发布使用情况指标的权限。	2025 年 6 月 13 日
AWSIoT FleetWiseServiceRolePolicy — AWS IoT FleetWise 开始跟踪更改	AWS IoT FleetWise 已开始跟踪该政策的变更。	2022 年 9 月 27 日

## AWS 物联网合规性验证 FleetWise

### Note

AWS 物联网 FleetWise不在任何合 AWS 规计划的范围内。

要了解是否属于特定合规计划的范围，请参阅AWS 服务“[按合规计划划分的范围](#)”，然后选择您感兴趣的合规计划。AWS 服务 有关一般信息，请参阅[AWS 合规计划AWS](#)。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的“[下载报告](#)”中的“[AWS Artifact](#)”。

您在使用 AWS 服务 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。有关您在使用时的合规责任的更多信息 AWS 服务，请参阅[AWS 安全文档](#)。

## AWS 物联网的弹性 FleetWise

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。各区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现故障转移的应用程序和数据库。与传统的单个或多个数据中心基础设施相比，可用区具有更高的可用性、容错能力和可扩展性。

有关 AWS 区域和可用区的更多信息，请参阅[AWS 全球基础设施](#)。

### Note

AWS 物联网处理的数据存储 FleetWise 在亚马逊 Timestream 数据库中。Timestream 支持备份到其他 AWS 可用区或区域。但是，您可以使用 Timestream SDK 编写自己的应用程序来查询数据并将其保存到您选择的目标。

有关 Amazon Timestream 的更多信息，请参阅[Amazon Timestream 开发人员指南](#)。

Amazon Timestream 不在亚太地区（孟买）区域推出。

## AWS 物联网中的基础设施安全 FleetWise

作为一项托管服务，AWS 物联网 FleetWise 受到 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅[AWS 云安全](#)。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS security Pillar Well-Architected Framework 中的[基础设施保护](#)。

您可以使用 AWS 已发布的 API 调用 FleetWise 通过网络访问 AWS 物联网。客户端必须支持以下内容：

- 传输层安全性协议（TLS）。我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密（PFS）的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

您可以从任何网络位置调用这些 API 操作，但是 AWS IoT FleetWise 确实支持基于资源的访问策略，其中可能包括基于源 IP 地址的限制。您还可以使用 AWS 物联网 FleetWise 策略来控制来自特定亚马逊虚拟私有云（Amazon VPC）终端节点或特定终端节点的访问 VPCs。实际上，这可以将对给定 AWS 物联网 FleetWise 资源的网络访问与 AWS 网络中的特定 VPC 隔离开来。

### 主题

- [FleetWise 通过接口 VPC 终端节点连接到 AWS 物联网](#)

## FleetWise 通过接口 VPC 终端节点连接到 AWS 物联网

您可以使用虚拟私有云（VPC）Private Cloud 中的[接口 VPC 终端节点（AWS PrivateLink）](#)直接连接到 AWS 物联网 FleetWise 网络，而不必通过互联网进行连接。当您使用接口 VPC 终端节点时，您 FleetWise

的 VPC 和 AWS 物联网之间的通信完全在 AWS 网络内进行。每个 VPC 终端节点都由一个或多个[弹性网络接口](#) (ENIs) 表示，其私有 IP 地址位于您的 VPC 子网中。

接口 VPC 终端节点将您的 VPC 直接连接到 AWS 物联网，FleetWise 无需互联网网关、NAT 设备、VPN Direct Connect 连接或连接。您的 VPC 中的实例不需要公有 IP 地址即可与 AWS 物联网 FleetWise API 通信。

要 FleetWise 通过您的 VPC 使用 AWS 物联网，您必须从 VPC 内的实例进行连接，或者使用 AWS Virtual Private Network (VPN) 或将您的私有网络连接到 VPC Direct Connect。有关 Amazon VPN 的信息，请参阅《Amazon Virtual Private Cloud 用户指南》中的[VPN 连接](#)。有关信息 AWS Direct Connect，请参阅《Direct Connect 用户指南》中的[创建连接](#)。

您可以使用 AWS 控制台或 AWS Command Line Interface (AWS CLI) 命令创建用于连接 AWS 物联网 FleetWise 的接口 VPC 终端节点。有关更多信息，请参阅[创建接口端点](#)。

创建接口 VPC 终端节点后，如果您为该终端节点启用私有 DNS 主机名，则默认 AWS IoT 终端节点将解析为您的 VPC FleetWise 终端节点。AWS IoT FleetWise 的默认服务名称端点采用以下格式。

```
iotfleetwise.Region.amazonaws.com
```

如果您不启用私有 DNS 主机名，Amazon VPC 将提供一个您可以使用的 DNS 端点名称，格式如下。

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

有关更多信息，请参阅《Amazon VPC 用户指南》中的[接口 VPC 端点 \(AWS PrivateLink\)](#)。

AWS 物联网 FleetWise 支持在您的 VPC 内调用其所有[API 操作](#)。

您可以将 VPC 端点策略附加到 VPC 端点，以控制 IAM 委托人的访问权限。您还可以将安全组与 VPC 端点关联，以便根据网络流量的源和目标（例如 IP 地址范围）控制入站和出站访问。有关更多信息，请参阅[使用 VPC 端点控制对服务的访问](#)。

#### Note

AWS 物联网 FleetWise 支持所有采用双堆栈模式的 VPC 终端节点。有关服务终端节点的信息，请参阅[AWS 物联网 FleetWise 终端节点和配额](#)。

## 为 AWS 物联网创建 VPC 终端节点策略 FleetWise

您可以为 AWS 物联网的 Amazon VPC 终端节点创建策略 FleetWise，以指定以下内容：

- 可以或不能执行操作的主体
- 可执行或不可执行的操作

有关更多信息，请参阅《Amazon VPC 用户指南》中的[使用 VPC 端点控制对服务的访问](#)。

Example— 拒绝来自指定 AWS 账户的所有访问的 VPC 终端节点策略

以下 VPC 终端节点策略拒绝使用该终端节点的 AWS 账户 **123456789012** 所有 API 调用。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

Example : 仅允许 VPC 访问指定的 IAM 委托人 ( 用户 ) 的 VPC 端点策略

以下 VPC 终端节点策略仅允许 AWS 账户 **lijuan** 中的用户拥有完全访问权限 **123456789012**。它拒绝所有其他 IAM 主体访问端点。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
```

```
    "Principal": {
      "AWS": [
        "arn:aws:iam::123456789012:user/Lijuan"
      ]
    }
  ]
}
```

### Example—用于 AWS 物联网 FleetWise 操作的 VPC 终端节点策略

以下是 AWS 物联网终端节点策略的示例 FleetWise。当连接到终端节点时，此策略授予 IAM 用户访问 *fleetWise* 中列出的 AWS 物联网 FleetWise 操作的权限 AWS 账户 **123456789012**。

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/fleetWise"
        ],
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:ListFleets",
        "iotfleetwise:ListCampaigns",
        "iotfleetwise:CreateVehicle",
      ]
    }
  ]
}
```

## AWS 物联网中的配置和漏洞分析 FleetWise

IoT 环境可能由大量具有不同功能、长期存在且地理位置分散的设备组成。这些特性导致设备设置复杂且容易出错。此外，由于设备的计算能力、内存和存储功能通常有限，因而限制了在设备本身上对加密和其他形式的安全功能的使用。设备经常使用具有已知漏洞的软件。这些因素使物联网设备（包括为 AWS 物联网 FleetWise 收集数据的车辆）成为黑客的有吸引力的目标，也使得持续保护它们变得困难。

配置和 IT 控制由您（我们的客户）共同 AWS 负责。有关更多信息，请参阅[责任 AWS 共担模型](#)。

# AWS 物联网安全最佳实践 FleetWise

AWS 物联网 FleetWise 提供了许多安全功能，供您在制定和实施自己的安全策略时考虑。以下最佳实践是一般准则，并不代表完整的安全解决方案。这些最佳实践可能不适合您的环境或不满足您的环境要求，请将其视为有用的考虑因素而不是惯例。

要了解安全性，AWS IoT 请参阅《AWS IoT 开发人员指南》AWS IoT Core [中的安全最佳实践](#)

## 授予可能的最低权限

通过在 IAM 角色中使用最低权限集，遵循最低特权原则。在 IAM 策略中限制对 Action 和 Resource 属性使用 \* 通配符。而是在可能的情况下声明一组有限的操作和资源。有关最低权限和其他策略最佳实践的更多信息，请参阅 [the section called “策略最佳实践”](#)。

## 不要记录敏感信息

您应该禁止记录凭证和其他个人身份信息 (PII)。建议您实施以下安全措施：

- 不要在设备名称中使用敏感信息。
- 请勿在名称和 IDs AWS 物联网 FleetWise 资源中使用敏感信息，例如在广告系列、解码器清单、车辆型号和信号目录或车辆和车队 IDs 的名称中使用敏感信息。

## AWS CloudTrail 用于查看 API 调用历史记录

您可以查看使用您的账户进行 FleetWise 的 AWS IoT API 调用的历史记录，以进行安全分析和操作故障排除。要接收使用您的账户进行 FleetWise 的 AWS IoT API 调用的历史记录，只需在 CloudTrail 中开启即可 AWS 管理控制台。有关更多信息，请参阅 [the section called “CloudTrail 日志”](#)。

## 使设备时钟保持同步

请务必确保您的设备上有准确的时间。X.509 证书具有到期日期和时间。设备上的时钟用于验证服务器证书是否仍有效。设备时钟可能会在一段时间后出现偏差，或者电池可能会放电。

有关更多信息，请参阅《AWS IoT Core 开发人员指南》中的 [保持设备时钟同步](#) 最佳实践。

# 监控 AWS 物联网 FleetWise

监控是维护 AWS IoT FleetWise 和其他 AWS 解决方案的可靠性、可用性和性能的重要组成部分。AWS 提供以下监控工具，用于监视 AWS IoT FleetWise、报告问题并在适当时自动采取措施：

- Amazon 会实时 CloudWatch 监控您的 AWS 资源和您运行 AWS 的应用程序。您可以收集和跟踪指标，创建自定义的控制面板，以及设置警报以在指标达到您指定的阈值时通知您或采取措施。例如，您可以 CloudWatch 跟踪您的 Amazon EC2 实例的 CPU 使用率或其他指标，并在需要时自动启动新实例。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。
- Amazon CloudWatch Logs 可用于监控、存储和访问来自 Amazon EC2 实例和其他来源的日志文件。CloudTrail CloudWatch 日志可以监视日志文件中的信息，并在达到特定阈值时通知您。您还可以在高持久性存储中检索您的日志数据。有关更多信息，请参阅 [Amazon CloudWatch 日志用户指南](#)。
- AWS CloudTrail 捕获由某个 AWS 账户发出或代表该账户发出的 API 调用和相关事件。然后它将日志文件传送到您指定的 Amazon S3 存储桶。您可以识别哪些用户和帐户拨打了电话 AWS、发出呼叫的源 IP 地址以及呼叫发生的时间。有关更多信息，请参阅 [AWS CloudTrail 《用户指南》](#)。

## FleetWise 使用 Amazon 监控 AWS 物联网 CloudWatch

### Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

Amazon CloudWatch 指标是监控您的 AWS 资源及其表现的一种方式。AWS 物联网 FleetWise 将指标发送到 CloudWatch。您可以使用 AWS 管理控制台 AWS CLI、或 API 列出 AWS 物联网 FleetWise 发送到的指标 CloudWatch。有关更多信息，请参阅 [Amazon CloudWatch 用户指南](#)。

### Important

您必须配置设置，以便 AWS IoT FleetWise 可以向其发送指标 CloudWatch。有关更多信息，请参阅 [配置您的 AWS 物联网 FleetWise 设置](#)。

AWS/IoTFleetWise 命名空间包括以下指标。

## 信号指标

指标	说明
IllegalMessageFromEdge	<p>从车辆发送并由 AWS 物联网接收的消息与要求的格式 FleetWise 不匹配。</p> <p>单位：计数</p> <p>维度：None</p> <p>有效统计数据：Sum</p>
MessageThrottled	<p>从车辆发送到 AWS 物联网的消息 FleetWise 受到限制。这是因为在当前区域内的此账户中，您已超出<a href="#">服务限制</a>。</p> <p>单位：计数</p> <p>维度：None</p> <p>有效统计数据：Sum</p>
ModelingError	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise 包含无法根据车辆模型进行验证的信号。</p> <p>单位：计数</p> <p>尺寸：ModelName、StateTemplateName (可选)、SignalCatalogName (可选)</p>
DecodingError	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise 包含无法根据车辆的解码器清单进行解码的信号。</p> <p>单位：计数</p> <p>尺寸：DecoderName</p> <p>有效统计数据：Sum</p>

指标	说明
MessageSizeLimitExceeded	<p>从车辆发送到 AWS 物联网的消息 FleetWise 被丢弃。这是因为您超出了当前区域中该账户的最大消息<a href="#">服务限制</a>。</p> <p>单位：计数</p> <p>维度：None</p> <p>有效统计数据：Sum</p>
CallCount	<p>在指定时间段内收录的消息数量。</p> <p>单位：计数</p> <p>维度：账户 ID</p>
CheckInThrottled	<p>车辆向 AWS 物联网发送的办理登机手续 FleetWise 受到限制。这是因为您在当前地区已超出该账户的<a href="#">服务限制</a>。</p> <p>单位：计数</p> <p>尺寸：VehicleName</p> <p>有效统计数据：Sum</p>
VehicleAttributeNotFound	<p>FleetWise 无法使用指定的车辆属性来丰富从车辆发送并由 AWS 物联网接收的消息。</p> <p>单位：计数</p> <p>尺寸：广告活动名称（可选）、stateTemplateLateName（可选）、车辆名称</p> <p>有效统计数据：Sum</p>

## 车辆指标

指标	说明
VehicleNotFound	<p>AWS 物联网收到的消息 FleetWise，其中车辆未知。</p> <p>单位：计数</p> <p>维度：None</p> <p>有效统计数据：Sum</p>

## 部署指标

指标	说明
PayloadSize	<p>从 AWS 物联网 FleetWise 发送到车辆的消息的大小。</p> <p>单位：计数</p> <p>尺寸：VehicleName、ResourceTypes (StateTemplates、广告系列、DecoderManifest)</p>
PayloadSizeLimitExceeded	<p>从 AWS 物联网发送 FleetWise 到车辆的消息超过了当前区域中该账户的最大有效载荷<a href="#">服务限制</a>。</p> <p>单位：计数</p> <p>尺寸：VehicleName</p> <p>有效统计数据：Sum</p>

## 活动指标

指标	说明
CampaignInvalid	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中活动无效。</p> <p>单位：计数</p> <p>尺寸：CampaignName</p> <p>有效统计数据：Sum</p>
CampaignNotFound	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中活动未知。</p> <p>单位：计数</p> <p>尺寸：CampaignName</p> <p>有效统计数据：Sum</p>

## 状态模板指标

指标	说明
NoStateTemplatesAssociated	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中没有状态模板与车辆关联。</p> <p>单位：计数</p> <p>有效统计数据：Sum</p>

## 活动数据目标指标

指标	说明
TimestreamWriteError	<p>AWS 物联网 FleetWise 无法将来自车辆的消息写入亚马逊 Timestream 表。</p>

指标	说明
	单位：计数  尺寸： DatabaseName , TableName  有效统计数据：Sum
S3 WriteError	AWS 物联网 FleetWise 无法将消息从车辆写入亚马逊简单存储服务 (Amazon S3) 存储桶。  单位：计数  尺寸： BucketName  有效统计数据：Sum
S3 ReadError	AWS 物联网 FleetWise 无法从亚马逊简单存储服务 (Amazon S3) 存储桶中的车辆中读取对象密钥。  单位：计数  尺寸： BucketName  有效统计数据：Sum

### 客户管理的 AWS KMS 关键指标

指标	说明
KMSKeyAccessDenied	AWS 由于 AWS KMS 密钥访问被拒绝错误，物联网 FleetWise 无法将消息从车辆写入 Timestream 表或 Amazon S3 存储桶。  单位：计数  尺寸：同 KMSKey上  有效统计数据：Sum

# FleetWise 使用 Amazon CloudWatch 日志监控 AWS 物联网

## ⚠ Important

目前对某些 AWS 物联网 FleetWise 功能的访问受到限制。有关更多信息，请参阅 [AWS AWS 物联网中的区域和功能可用性 FleetWise](#)。

Ama CloudWatch zon Logs 会监控您的资源中发生的事件，并在出现任何问题时提醒您。如果您收到警报，可以访问日志文件以获取有关具体事件的信息。有关更多信息，请参阅 [Amazon CloudWatch 日志用户指南](#)。

## 在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志

## ⚠ Important

在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志组之前，请确保满足以下条件：

- 您已启用 AWS 物联网登录功能 FleetWise。有关日志记录的更多信息，请参阅[配置 AWS 物联网 FleetWise 日志](#)。
- 已经有 AWS IoT 操作写入的日志条目。

在 CloudWatch 控制台中查看您的 AWS IoT FleetWise 日志

1. 打开 [CloudWatch 控制台](#)。
2. 在导航窗格中，依次选择日志和日志组。
3. 选择相应的日志组。
4. 选择搜索日志组。您将看到为您的账户生成的日志事件的完整列表。
5. 选择展开图标可查看各个日志流并查找日志级别为 ERROR 的所有日志。

您也可以在筛选事件搜索框中输入查询。例如，您可以尝试执行以下操作：

```
{ $.logLevel = "ERROR" }
```

有关创建筛选条件表达式的更多信息，请参阅 Amazon L CloudWatch ogs 用户指南中的[筛选条件和模式语法](#)。

## Example 日志条目

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
  "eventType": "MODELING_ERROR",
  "logLevel": "ERROR",
  "timestamp": 1685743214239,
  "campaignName": "test-campaign",
  "signalCatalogName": "test-catalog",
  "signalId": 10242
}
```

## 信号事件类型

事件类型	说明
MODELING_ERROR	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise 包含无法根据车辆模型进行验证的信号。</p> <p>属性：VehicleName、CampaignName ( 可选 )、signalCatalogName、signalID ( 可选 )、signalValue ( 可选 )、M signalValueRange in ( 可选 )、signalValueRange Max ( 可选 )、( 可选 )、signalIds、modelManifestName stateTemplateName</p>
ILLEGAL_MESSAGE_FROM_EDGE	<p>从车辆发送并由 AWS 物联网接收的消息与要求的格式 FleetWise 不匹配。</p> <p>属性：车辆名称、战役名称、signalCatalogName</p>
DECODING_ERROR	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise 包含无法根据车辆的解码器清单进行解码的信号。</p>

事件类型	说明
	<p>属性：campaignName、 signalCatalogName、 decoderManifestName、 ( 可选 ) signalName、 ( 可选 ) s3uri</p>
MESSAGE_TTLEDED	<p>从车辆发送到 AWS 物联网的消息 FleetWise 受到限制。这是因为在当前区域内的此账户中，您已超出服务限制。</p> <p>属性：账户 ID、车辆名称、消息、事件类型、日志级别、时间戳</p>
已超出消息大小限制	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise 超过了消息<a href="#">服务限制</a>的最大大小。</p> <p>属性：账户 ID、车辆名称</p>
CHECKIN_THROTTLED	<p>车辆向 AWS 物联网发送的办理登机手续 FleetWise 受到限制。这是因为您在当前地区已超出该账户的<a href="#">服务限制</a>。</p> <p>属性：车辆名称</p>
未找到车辆属性	<p>FleetWise 无法使用指定的车辆属性来丰富从车辆发送并由 AWS 物联网接收的消息。</p> <p>属性：CampaignName ( 可选 )、 stateTemplateName ( 可选 )、 VehicleName、 vehicleAttributeNames</p>

## 车辆事件类型

事件类型	说明
VEHICLE_NOT_FOUND	<p>AWS 物联网收到的消息 FleetWise ，其中车辆未知。</p>

事件类型	说明
	属性：车辆名称、活动名称（可选）、stateTemplateName（可选）

### 部署事件类型

事件类型	说明
已超过有效载荷大小限制	<p>从 AWS 物联网发送 FleetWise 到车辆的消息超过了最大大小服务限制。</p> <p>属性：车辆名称、活动名称（可选）、stateTemplateName（可选）</p>

### 活动事件类型

事件类型	说明
CAMPAIGN_NOT_FOUND	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中活动未知。</p> <p>属性：vehicleName（可选）、campaignName</p>
CAMPAIGN_INVALID	<p>从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中活动无效。</p> <p>属性：vehicleName（可选）、campaignName</p>

### 活动数据目标事件类型

事件类型	说明
TIMESTREAM_WRITE_ERROR	<p>AWS 物联网 FleetWise 无法将来自车辆的消息写入亚马逊 Timestream 表。</p> <p>属性：车辆名称、战役名称、timestreamDatabaseName timestreamTableName</p>

事件类型	说明
S3_WRITE_ERROR	AWS 物联网 FleetWise 无法将消息从车辆写入亚马逊简单存储服务 (Amazon S3) 存储桶。  属性：campaignName、destinationName
S3_READ_ERROR	AWS 物联网 FleetWise 无法从亚马逊简单存储服务 (Amazon S3) 存储桶中的车辆中读取对象密钥。  属性：campaignName、destinationName

### 状态模板事件类型

事件类型	说明
未找到状态模板	从车辆发送并由 AWS 物联网接收的消息 FleetWise，其中状态模板未知。  属性：车辆名称（可选），stateTemplateName

### 客户管理的 AWS KMS 密钥事件类型

事件类型	说明
KMS_KEY_ACCESS_DENIED	AWS 由于 AWS KMS 密钥访问被拒绝错误，物联网 FleetWise 无法将消息从车辆写入 Timestream 表或 Amazon S3 存储桶。  属性：kmsKeyId（可选）、resourceArn（可选）

### 属性

所有 CloudWatch 日志条目都包含以下属性：

## accountId

你的 AWS 账户 身份证。

## eventType

已为其生成日志的事件类型。事件类型的值取决于生成日志条目的事件。每个日志条目描述都包括该日志条目的 eventType 的值。

## logLevel

正在使用的日志级别。有关更多信息，请参阅《AWS IoT Core 开发人员指南》中的 [日志级别](#)。

## message

包含有关日志的具体详情。

## timestamp

AWS 物联网 FleetWise 处理日志时的纪元毫秒时间戳。

## 可选属性

CloudWatch 日志条目可以选择包含以下属性，具体取决于 eventType：

### decoderManifestName

包含信号的解码器清单名称。

### destinationName

车辆数据目标的名称。例如 Amazon S3 存储桶名称。

### campaignName

活动的名称。

### signalCatalogName

包含信号的信号目录名称。

### signalId

错误信号的 ID。

### signalIds

错误信号列表 IDs。

## signalName

信号的名称。

## signalTimestampEpoch

错误信号的时间戳。

## signalValue

错误信号的值。

## signalValueRangeMax

错误信号的最大范围。

## signalValueRangeMin

错误信号的最小范围。

## s3URI

车辆消息中 Amazon Ion 文件的 Amazon S3 唯一标识符。

## timestreamDatabaseName

Timestream 数据库的名称。

## timestreamTableName

Timestream 表的名称。

## vehicleName

车辆的名称。

## vehicleAttributeNames

找不到的车辆属性名称列表。

## 配置 AWS 物联网 FleetWise 日志

您可以将 AWS IoT FleetWise 日志数据发送到 CloudWatch 日志组。CloudWatch 如果 AWS 物联网 FleetWise 无法处理来自车辆的消息，日志可以提供可见性。例如，这个问题可能是由于配置错误或其他客户端错误而导致的。您会收到任何错误的通知，以便可以识别和缓解问题。

必须先创建一个日志组 CloudWatch，然后才能向发送 CloudWatch 日志。使用与 AWS 物联网相同的账户和区域配置日志组 FleetWise。在 AWS IoT 中启用登录功能时 FleetWise，请提供日志组名称。启用日志记录后，AWS IoT 会将日志以 CloudWatch 日志流的形式 FleetWise 传送到日志组。

您可以在 CloudWatch 控制台中查看从 AWS IoT 发送 FleetWise 的日志数据。有关配置 CloudWatch 日志组和查看日志数据的更多信息，请参阅[使用日志组](#)。有关为 AWS 服务设置日志的更多信息，请参阅 Amazon Logs 用户指南中的向 CloudWatch 日志发布 CloudWatch 日志的[AWS 服务](#)。

## 向其发布日志的权限 CloudWatch

为日志组配置 CloudWatch 日志记录需要本节中所述的权限设置。有关管理权限的信息，请参阅 IAM 用户指南中的[AWS 资源访问管理](#)。

使用这些权限，您可以更改日志配置、配置日志传送以及检索有关您的日志组的信息。 CloudWatch

### 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"
  }
]
```

如果允许对所有 AWS 资源执行操作，则会在策略中指明，“Resource”设置为“\*”。这意味着允许对每个操作支持的所有 AWS 资源执行这些操作。

## 使用控制台配置日志记录

本节介绍如何使用 AWS 物联网 FleetWise 控制台配置日志记录。

使用 AWS IoT FleetWise 控制台配置日志

1. 打开[AWS 物联网 FleetWise 控制台](#)。
2. 在左侧窗格中，选择设置。
3. 在设置页面的日志记录部分，选择编辑。
4. 在 CloudWatch 日志部分，输入日志组。
5. 选择提交可保存更改。

启用日志记录后，您可以在[CloudWatch 控制台](#)中查看日志数据。

## 使用 CLI 配置日志

本节介绍如何使用 CLI 为 AWS IoT FleetWise 配置日志记录。

您也可以使用 API 中与此处显示的 CLI 命令相对应的方法，对 AWS API 执行此过程。您可以使用 [GetLoggingOptions](#) API 操作来获取当前配置，使用 [PutLoggingOptions](#) API 操作来修改配置。

使用 CLI 为 AWS 物联网配置日志 FleetWise

1. 使用 `get-logging-options` 命令为您的账户设置日志记录选项。

```
aws iotfleetwise get-logging-options
```

## 2. 要启用日志记录，请使用 `put-logging-options` 命令。

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery
logType=ERROR,logGroupName=MyLogGroup
```

其中：

`logType`

要向日志发送数据的 CloudWatch 日志类型。要禁用日志记录，请将该值更改为 `OFF`。

`logGroupName`

操作将数据发送到的 CloudWatch 日志组。在启用 IoT AWS T 日志记录之前，请务必创建日志组名称 `FleetWise`。

启用日志记录后，请参阅[使用 AWS CLI 搜索日志条目](#)。

## 使用记录 AWS 物联网 FleetWise API 调用 AWS CloudTrail

AWS 物联网 FleetWise 与 AWS CloudTrail 一项服务集成，该服务提供用户、角色或 AWS 服务在 AWS 物联网中采取的操作的记录 FleetWise。CloudTrail 将 AWS 物联网的所有 API 调用捕获 FleetWise 为事件。捕获的调用包括来自 AWS 物联网 FleetWise 控制台的调用和对 AWS 物联网 FleetWise API 操作的代码调用。如果您创建跟踪，则可以将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括物 AWS 联网事件 FleetWise。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的“事件历史记录”中查看最新的事件。使用收集的信息 CloudTrail，您可以确定向 AWS 物联网发出的请求 FleetWise、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅《[AWS CloudTrail 用户指南](#)》。

## AWS 中的物联网 FleetWise 信息 CloudTrail

CloudTrail 在您创建 AWS 账户时已在您的账户上启用。当 AWS 物联网中发生活动时 FleetWise，该活动会与其他 AWS 服务 CloudTrail 事件一起记录在事件历史记录中。您可以在 AWS 账户中查看、搜索和下载最新事件。有关更多信息，请参阅[使用事件历史记录查看 CloudTrail 事件](#)。

要持续记录 AWS 账户中的事件，包括 AWS 物联网事件 FleetWise，请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。默认情况下，在控制台中创建跟踪记录时，此跟踪记录应用于所有 AWS 区域。跟踪记录 AWS 分区中所有区域的事件，并将日志文件传送到您指定的

Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅以下内容：

- [创建跟踪记录概述](#)
- [CloudTrail 支持的服务和集成](#)
- [配置 Amazon SNS 通知 CloudTrail](#)
- [接收来自多个区域的 CloudTrail 日志文件](#)
- [接收来自多个账户的 CloudTrail 日志文件](#)

所有 AWS 物联网 FleetWise 操作均由《物联网 API 参考》记录 CloudTrail 并记录在《[AWS 物联网 FleetWise API 参考](#)》中。例如，对 `CreateCampaignAssociateVehicleFleet`、和 `GetModelManifest` 操作的调用会在 CloudTrail 日志文件中生成条目。

每个事件或日志条目都包含有关生成请求的人员信息。身份信息可帮助您确定以下内容：

- 请求是使用根用户凭证还是 IAM 用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其他 AWS 服务发出。

有关更多信息，请参阅 [CloudTrail userIdentity 元素](#)。

## 了解 AWS IoT FleetWise 日志文件条目

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

以下示例显示了演示该 `AssociateVehicleFleet` 操作的 CloudTrail 日志条目。

```
{
  "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"
}
```

# 《AWS 物联网 FleetWise 开发者指南》的文档历史记录

下表描述了 AWS IoT 的文档版本 FleetWise。

变更	说明	日期
<a href="#">监控更新</a>	更新了 AWS IoT FleetWise 监控以包含更多指标和日志选项。有关更多信息，请参阅使用 Amazon Logs <a href="#">监控 AWS 物联网 Amazon CloudWatch</a> 和 FleetWise 使用 <a href="#">Amazon CloudWatch 日志监控 AWS 物联网</a> 。	2025 年 12 月 23 日
<a href="#">服务相关角色策略更新</a>	更新了 AWS IoT FleetWise ServiceRolePolicy 以包括向 AWS/Usage 命名空间发布使用情况指标的权限。有关更多信息，请参阅 <a href="#">AWS IoT FleetWise ServiceRolePolicy 政策更新</a> 。	2025 年 6 月 13 日
<a href="#">区域扩展</a>	AWS 物联网 FleetWise 现已在亚太地区（孟买）推出（仅限门禁访问）。	2024 年 11 月 21 日
<a href="#">新功能的公开上市时间有限</a>	AWS 物联网 FleetWise 现在支持对活动进行封闭式访问，以存储和转发数据、将 MQTT 主题配置为数据目标以及收集诊断故障代码数据。它现在还支持门禁访问，使用自定义解码接口、配置命令和监控车辆的最新已知状态来收集与网络无关的数据。	2024 年 11 月 21 日

[向 MQTT 主题发送活动数据](#)

AWS 物联网 FleetWise 现在支持将活动期间收集的数据发送到您指定的 MQTT 主题，此外还可以将数据存储到 Amazon S3 或 Amazon Timestream 中。

2024 年 5 月 1 日

[视觉系统数据预览](#)

您可以使用来自 AWS 物联网的视觉系统数据的预览 FleetWise 来收集和整理来自车辆视觉系统的数据，包括来自摄像头、雷达和激光雷达的数据。它可使结构化和非结构化视觉系统数据、元数据（事件 ID、活动、车辆）和标准传感器（遥测数据）在云中自动同步。

2023 年 11 月 26 日

[AWS KMS 客户管理的密钥](#)

AWS 物联网 FleetWise 现在支持 AWS KMS 客户托管密钥。您可以使用 KMS 密钥对存储在中的 AWS 物联网 FleetWise 资源（信号目录、车辆模型、解码器清单、车辆和数据收集活动配置）相关的服务器端数据进行加密。AWS Cloud

2023 年 10 月 16 日

[Amazon S3 中的对象存储](#)

AWS 物联网 FleetWise 现在支持使用亚马逊简单存储服务 (Amazon S3) Service 存储数据。除了 Amazon Timestream 外，还可以将活动期间收集的数据存储在 Amazon S3 中。

2023 年 6 月 1 日

[正式发布](#)

这是 AWS 物联网的公开发布 FleetWise。

2022 年 9 月 27 日

[初始版本](#)

这是《AWS 物联网 FleetWise 2021 年 11 月 30 日  
开发者指南》的预览版。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。