



开发人员指南

# AWS 物联网 FleetWise



# AWS 物联网 FleetWise: 开发人员指南

Copyright © 2024 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
应用场景 .....	2
您不熟悉 AWS IoT FleetWise ? .....	2
访问 AWS IoT FleetWise .....	3
AWS IoT FleetWise 的定价 .....	3
AWS IoT FleetWise 的工作原理 .....	3
重要概念 .....	3
AWS IoT FleetWise 的功能 .....	7
相关服务 .....	7
设置 AWS 物联网 FleetWise .....	8
设置你的 AWS 账户 .....	8
注册获取 AWS 账户 .....	8
创建具有管理访问权限的用户 .....	8
开始使用控制台 .....	10
配置设置 .....	10
配置设置 ( 控制台 ) .....	10
配置设置 (AWS CLI) .....	11
开始使用 .....	13
要求 .....	13
使用 Edge Agent 软件演示 .....	13
开始使用 ( 控制台 ) .....	14
先决条件 .....	15
步骤 1 : 为 AWS 物联网设置边缘代理软件 FleetWise .....	15
步骤 2 : 创建车辆型号 .....	16
步骤 3 : 创建解码器清单 .....	18
步骤 4 : 配置解码器清单 .....	19
步骤 5 : 创建车辆 .....	19
步骤 6 : 创建活动 .....	21
步骤 7 : 清除 .....	22
后续步骤 .....	22
将数据摄取到云端 .....	23
对车辆进行建模 .....	26
信号目录 .....	28

配置信号 .....	30
创建信号目录 (AWS CLI) .....	36
导入信号目录 .....	40
更新信号目录 (AWS CLI) .....	50
删除信号目录 (AWS CLI) .....	51
获取信号目录信息 (AWS CLI) .....	52
车辆型号 .....	53
创建车辆型号 .....	54
更新车辆型号 (AWS CLI) .....	59
删除车辆型号 .....	60
获取车辆型号信息 (AWS CLI) .....	61
解码器清单 .....	62
配置网络接口和解码器信号 .....	64
创建解码器清单 .....	66
更新解码器清单 (AWS CLI) .....	73
删除解码器清单 .....	74
获取解码器清单信息 (AWS CLI) .....	75
车辆 .....	77
预配置车辆 .....	78
对车辆进行身份验证 .....	78
授权车辆 .....	80
保留的主题 .....	81
创建车辆 .....	83
创建车辆 (控制台) .....	83
创建车辆 (AWS CLI) .....	85
创建多辆车 (AWS CLI) .....	87
更新车辆 (AWS CLI) .....	88
更新多辆车 (AWS CLI) .....	89
删除车辆 .....	90
删除车辆 (控制台) .....	90
删除车辆 (AWS CLI) .....	90
获取车辆信息 (AWS CLI) .....	91
车队 .....	92
创建车队 (AWS CLI) .....	93
将车辆与车队关联 (AWS CLI) .....	93
解除车辆与车队的关联 (AWS CLI) .....	94

更新车队 (AWS CLI) .....	94
删除车队 (AWS CLI) .....	95
获取车队信息 (AWS CLI) .....	95
市场活动 .....	97
创建活动 .....	101
创建活动 (控制台) .....	102
创建活动 (AWS CLI) .....	108
活动的逻辑表达式 .....	111
更新活动 (AWS CLI) .....	112
删除活动 .....	113
创建活动 (控制台) .....	113
删除活动 (AWS CLI) .....	113
获取活动信息 (AWS CLI) .....	113
处理和可视化车辆数据 .....	115
在 Timestream 中处理车辆数据 .....	115
可视化存储在 Timestream 中的车辆数据 .....	116
在 S3 中处理车辆数据 .....	116
S3 对象格式 .....	117
分析存储在 S3 中的车辆数据 .....	117
AWS CLI 和 AWS SDK .....	120
排查问题 .....	121
解码器清单问题 .....	121
适用于 AWS IoT FleetWise 的边缘代理软件问题 .....	124
问题：边缘代理软件无法启动。 .....	124
问题：[错误] [IoTFleetWiseEngine::connect]: [无法初始化持久性库] .....	125
问题：边缘代理软件不收集车载诊断 (OBD) II PID 和故障诊断代码 (DTC)。 .....	125
问题：适用于 AWS IoT FleetWise 的边缘代理软件无法从网络中收集数据或无法应用数据检 查规则。 .....	126
问题：[错误] [AwsIotConnectivityModule::connect]: [连接失败并出现错误] 或 [警告] [AwsIotChannel::send:: send]: [无有效 MQTT 连接。] .....	126
安全性 .....	127
数据保护 .....	127
静态加密 .....	128
传输中加密 .....	129
数据加密 .....	129
控制访问权限 .....	136

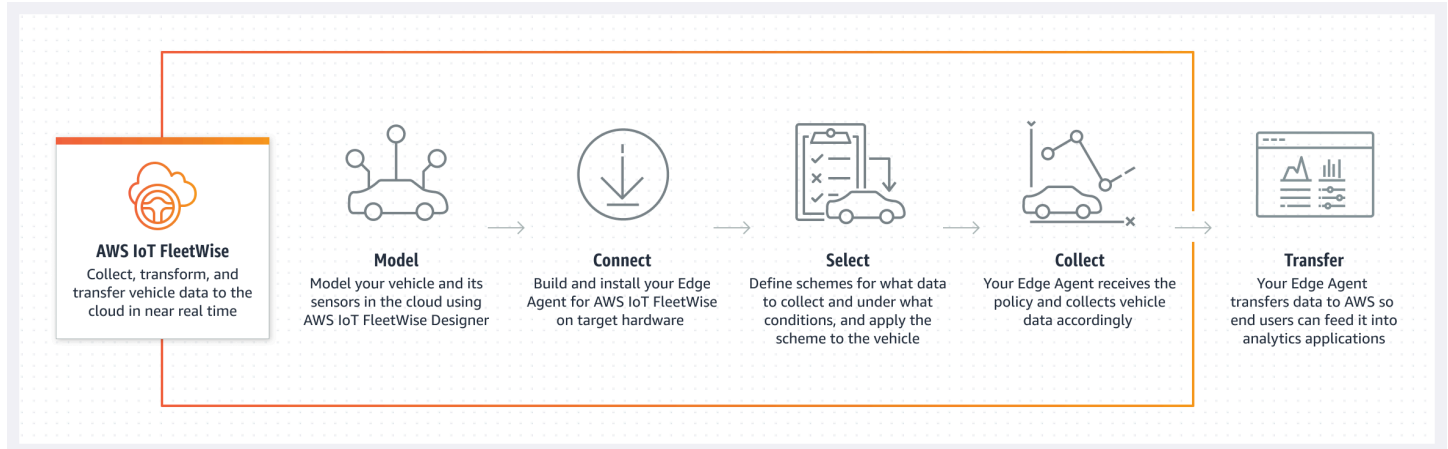
授予 AWS IoT FleetWise 对 Amazon S3 目的地的访问权限 .....	137
授予 AWS IoT FleetWise 访问亚马逊 Timestream 目的地的权限 .....	140
Identity and Access Management .....	142
受众 .....	143
使用身份进行身份验证 .....	143
使用策略管理访问 .....	146
AWS 物联网如何 FleetWise 与 IAM 配合使用 .....	148
基于身份的策略示例 .....	156
故障排除 .....	159
合规性验证 .....	160
韧性 .....	161
基础设施安全性 .....	162
FleetWise 通过接口 VPC 终端节点连接到 AWS 物联网 .....	162
配置和漏洞分析 .....	165
安全最佳实操 .....	165
授予可能的最低权限 .....	166
不要记录敏感信息 .....	166
AWS CloudTrail 用于查看 API 调用历史记录 .....	166
使设备时钟保持同步 .....	166
监控 .....	167
使用 CloudWatch 进行监控 .....	167
使用 CloudWatch Logs 进行监控 .....	171
在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志 .....	171
配置日志记录 .....	176
CloudTrail 日志 .....	178
AWS CloudTrail 中的 IoT FleetWise 信息 .....	179
了解 AWS IoT FleetWise 日志文件条目 .....	179
文档历史记录 .....	181
.....	clxxxii

# 什么是 AWS IoT FleetWise ？

AWS IoT FleetWise 是一项托管式服务，可用于收集车辆数据并将其组织到云中。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。利用 AWS IoT FleetWise，您可以从使用不同的协议和数据格式的的车辆中收集和整理数据。AWS IoT FleetWise 有助于将低级消息转换为人类可读的值，并标准化云中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。

当车辆数据存储于云端时，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。

下图演示了 AWS IoT FleetWise 的基本架构。



## 主题

- [优势](#)
- [应用场景](#)
- [您不熟悉 AWS IoT FleetWise ？](#)
- [访问 AWS IoT FleetWise](#)
- [AWS IoT FleetWise 的定价](#)
- [AWS IoT FleetWise 的工作原理](#)
- [相关服务](#)

# 优势

AWS IoT FleetWise 的主要优势包括：

## 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

获得从标准传感器（遥测数据）和视觉系统（相机、雷达和激光雷达的数据）收集的数据的统一视图，并在云中自动同步。AWS IoT FleetWise 可在云端自动同步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。这简化了整理事件全景视图并获得见解的过程。

### Note

视觉系统数据目前为预览版，可能会发生变化。

# 应用场景

您可以使用 AWS IoT FleetWise 的场景包括以下几种：

## 训练 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。

# 您不熟悉 AWS IoT FleetWise ？

如果您不熟悉 AWS IoT FleetWise，建议先阅读以下部分：



- [AWS IoT FleetWise 的工作原理](#)
- [设置 AWS 物联网 FleetWise](#)
- [边缘代理软件演示](#)
- [将数据摄取到云端](#)

## 访问 AWS IoT FleetWise

您可以使用 AWS IoT FleetWise 控制台或 API 来访问 AWS IoT FleetWise。

## AWS IoT FleetWise 的定价

车辆通过 MQTT 消息将数据发送到云端。您需要在每月月底为在 AWS IoT FleetWise 中创建的车辆付费。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](#)。

## AWS IoT FleetWise 的工作原理

以下各节概述了 AWS IoT 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 IoT FleetWise。

主题

- [重要概念](#)
- [AWS IoT FleetWise 的功能](#)

### 重要概念

AWS IoT 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 IoT FleetWise 还提供参考实现，以帮助您在车辆中安装边缘代理软件。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。

下面是 AWS IoT FleetWise 的核心概念：

## 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[创建和管理信号目录](#)。

## 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信号，AWS IoT FleetWise 希望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[创建和管理信号目录](#)。

## 车辆模型 (模型清单)

车辆模型是声明性结构，可用于标准化车辆形式并定义车辆中信号之间的关系。车辆模型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模型。有关更多信息，请参阅[创建和管理车辆型号](#)。

## 解码器清单

解码器清单包含车辆模型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，AWS IoT 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模型相关联。有关更多信息，请参阅[创建和管理解码器清单](#)。

## 网络接口

包含有关车载网络使用的协议的信息。AWS IoT 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 (ECU) 之间如何传输数据。ECU 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，定义了如何在 ECU 之间传输自我诊断数据。它提供了许多标准故障诊断代码 (DTC)，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 IoT FleetWise 支持用于视觉系统数据的 ROS 2 中间件。

## 解码器信号

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解码器信号配对。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，则它必须包含 OBD 解码器信号。

如果解码器清单还包含车辆中间件接口，则必须包含消息解码器信号。

## 车辆

您的实体车辆（例如汽车或卡车）的虚拟表示。车辆是车辆模型的实例。使用相同车辆模型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 IoT 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。

# AWS IoT FleetWise 的功能

以下是 AWS IoT FleetWise 的主要功能。

## 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 IoT FleetWise 支持[车辆信号规范 \(VSS\)](#)，您可以使用该规范来标准化车辆信号。

## 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 IoT FleetWise 的边缘代理软件

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。

## 相关服务

AWS IoT FleetWise 可以通过与以下 AWS 服务集成来提高云解决方案的可用性和可扩展性。

- AWS IoT Core – 注册并控制将传感器数据上传到 AWS IoT FleetWise 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](#)。

# 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](#)
- [配置设置](#)

## 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。

要注册 AWS 账户

1. 打开 <https://portal.aws.amazon.com/billing/signup>。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，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。作为安全最佳实践，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。在任何时候，您都可以通过转至 <https://aws.amazon.com/> 并选择我的账户来查看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 Management Console](#)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[以根用户身份登录](#)。

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网址。

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[创建权限集](#)。

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[添加组](#)。

**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. 导航到[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[创建车辆型号（控制台）](#)。

## 配置设置

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[使用 Amazon CloudWatch 监控 AWS IoT FleetWise](#)。

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 IoT FleetWise 日志记录](#)。

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[数据加密](#)。

## 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. 导航到[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 [密钥管理](#)。
    - 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。

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 FleetwiseServiceRole"
},
"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。

# 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：

- [将数据摄取到云端](#)
- [对车辆进行建模](#)
- [创建、预配置和管理车辆](#)
- [创建和管理车队](#)
- [通过活动收集和传输数据](#)

## 要求

要开始使用 AWS 物联网 AWS 账户，你必须有一个 FleetWise。如果没有，请参阅[设置 AWS 物联网 FleetWise](#)。

使用可用 AWS IoT FleetWise 的区域。有关更多信息，请参阅[AWS IoT FleetWise 终端节点和配额](#)。您可以使用中的区域选择器切换 AWS Management Console 到其中一个区域。

## 边缘代理软件演示

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 AWS 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。有关快速入门演示的更多信息，请执行以下操作，以下载《边缘代理软件开发人员指南》。

下载快速入门演示

1. 导航到[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。

Internet of Things

# AWS IoT FleetWise

## Unlock the value of vehicle data

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

### Get started with AWS IoT FleetWise

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

[View models](#)[Explore Edge Agent](#)

## 教程：AWS 物联网入门 FleetWise (控制台)

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 IoT FleetWise 的工作原理](#)。

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。

### 主题

- [先决条件](#)
- [步骤 1：为 AWS 物联网设置边缘代理软件 FleetWise](#)
- [步骤 2：创建车辆型号](#)
- [步骤 3：创建解码器清单](#)

- [步骤 4：配置解码器清单](#)
- [步骤 5：创建车辆](#)
- [步骤 6：创建活动](#)
- [步骤 7：清除](#)
- [后续步骤](#)

## 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 Account Management 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。
- 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](#)。

## 步骤 1：为 AWS 物联网设置边缘代理软件 FleetWise

### Note

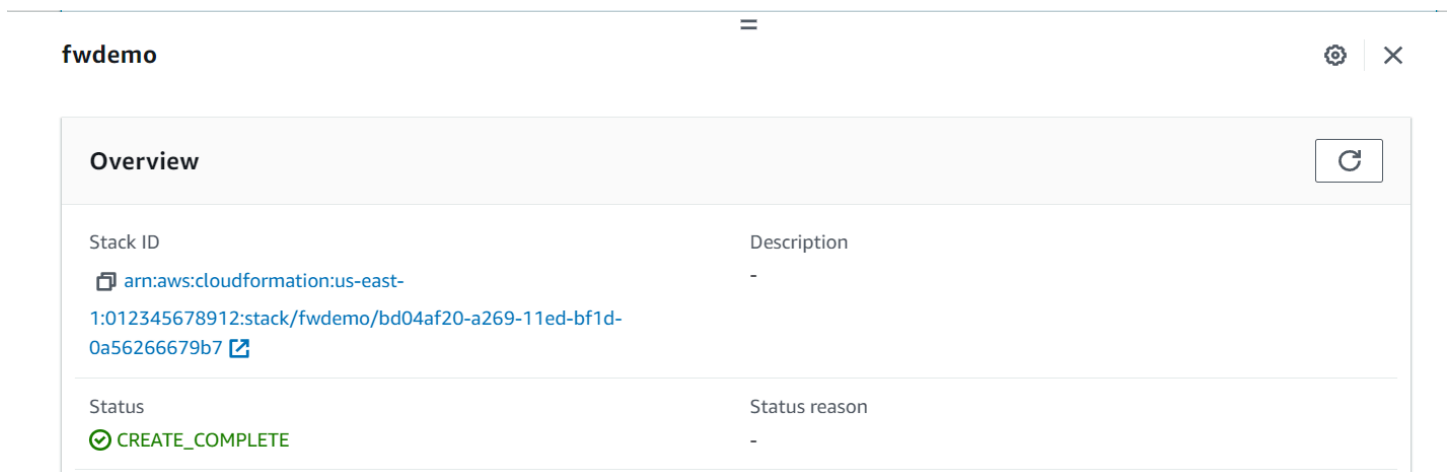
此步骤中的 CloudFormation 堆栈使用遥测数据。您也可以使用视觉系统数据创建 CloudFormation 堆栈。有关更多信息，请参阅[视觉系统数据开发人员指南](#)。视觉系统数据目前为预览版，可能会发生变化。

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 AWS 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。
  - CoreRegion - 您可以通过更新物联网 CoreRegion 参数来指定创建 AWS IoT 事物的区域。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 AWS 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。



### ⚠ 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 AWS 物联网 FleetWise 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 - 如果要使用用于定义如何在电子控制单元 (ECU) 之间传输自诊断数据的协议，请选择此接口类型。该协议提供了许多标准诊断故障代码 (DTC)，可以帮助您解决车辆问题。
  - 控制器局域网 (CAN 总线) 接口 - 如果要使用用于定义如何在 ECU 之间传输数据的协议，请选择此接口类型。ECU 可以是引擎控制单元、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，请从列表中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“已激活”，请打开 AWS IoT 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 arn:aws:iotfleetwise:us-east-1:012345678912:model-manifest/demo	Status ACTIVE	Date created February 01, 2023 at 14:40 (UTC-05)
Signal catalog ARN arn:aws:iotfleetwise:us-east-1:012345678912:signal-catalog/DefaultSignalCatalog	Description -	Last modified February 01, 2023 at 14:40 (UTC-05)

## 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 AWS CloudFormation 堆栈和所有堆栈资源。

### 删除 AWS CloudFormation 堆栈

1. 打开[AWS 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 Delete（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 [处理和可视化车辆数据](#)。
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 [排查 AWS IoT FleetWise 问题](#)。

# 将数据摄取到云端

适用于 AWS IoT FleetWise 的边缘代理软件安装到车辆并运行后，可以促进车辆与云之间的安全通信。

## Note

- 对于可能导致严重人身伤害、死亡、环境破坏或财产损坏的任何危险环境或关键系统操作，AWS IoT FleetWise 不适用，亦不与其关联。通过使用 AWS IoT FleetWise 收集的车辆数据仅供参考，您不得使用 AWS IoT FleetWise 来控制或操作车辆功能。
- 应根据您的用例评估使用 AWS IoT FleetWise 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）可能施加的任何合规性义务。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. 在您的车辆中开发并安装适用于 AWS IoT FleetWise 的边缘代理软件。有关如何使用边缘代理软件的更多信息，请执行以下操作，下载 [适用于 AWS IoT FleetWise 的边缘代理软件开发人员指南](#)。
  1. 导航到 [AWS IoT FleetWise 控制台](#)。
  2. 在服务主页上的开始使用 AWS IoT FleetWise 部分，选择浏览边缘代理。
2. 创建或导入信号目录，其中包含用于创建车辆模型的信号。有关更多信息，请参阅 [创建信号目录 \(AWS CLI\)](#) 和 [导入信号目录 \(AWS CLI\)](#)：

## Note

- 如果您使用 AWS IoT 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 [创建车辆型号 \(控制台\)](#)。
- AWS IoT FleetWise 目前支持每个 AWS 区域中每个 AWS 账户的信号目录。

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

**Note**

- 如果您使用 AWS IoT 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。创建车辆模型后，新的信号会自动添加到信号目录中。有关更多信息，请参阅[创建车辆型号（控制台）](#)。
- 如果您使用 CreateModelManifest API 操作创建车辆模型，则必须使用 UpdateModelManifest API 操作来激活车辆模型。有关更多信息，请参阅[更新车辆型号 \(AWS CLI\)](#)。
- 如果您使用 AWS IoT 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辆模型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模型关联。有关更多信息，请参阅[创建和管理解码器清单](#)。

**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[更新解码器清单 \(AWS CLI\)](#)。
- 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. 根据车辆模型创建车辆。使用相同车辆模型创建的车辆会继承相同的一组信号。必须先使用 AWS IoT Core 预配置车辆，然后才能将数据摄取到云端。有关更多信息，请参阅[创建、预配置和管理车辆](#)。
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[创建和管理车队](#)。
7. 创建活动。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[通过活动收集和传输数据](#)。

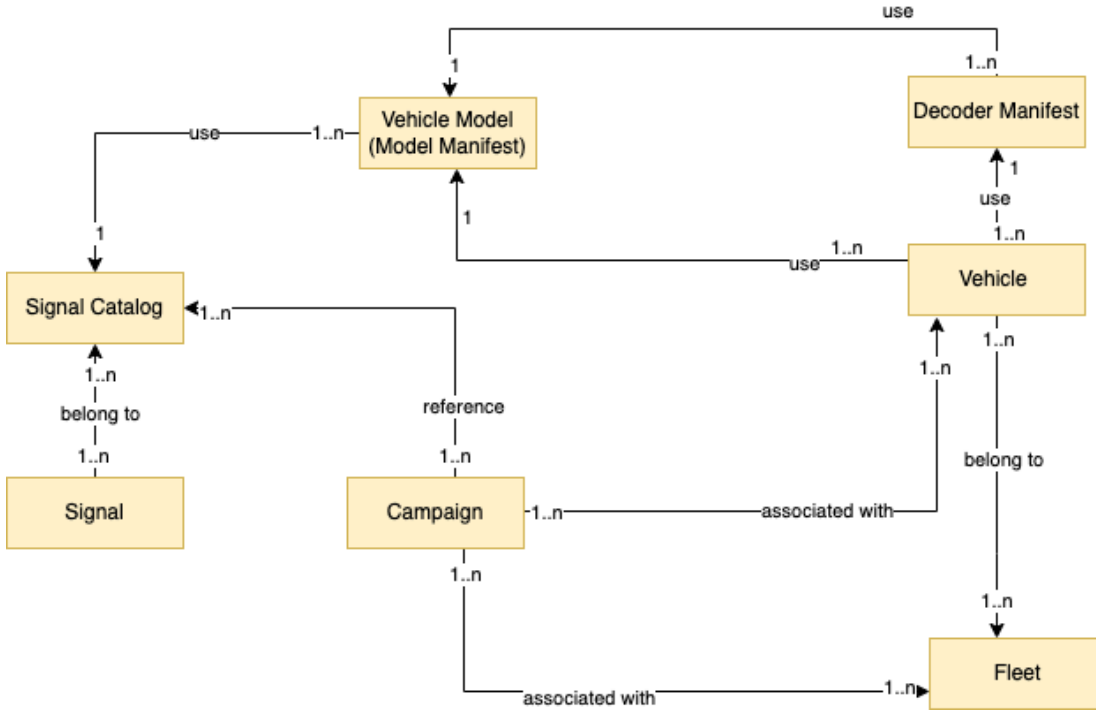
**Note**

必须先使用 UpdateCampaign API 操作批准该活动，AWS IoT FleetWise 才能将其部署到车辆或车队。有关更多信息，请参阅[更新活动 \(AWS CLI\)](#)。

边缘代理软件使用保留的主题 `$aws/iotfleetwise/vehicles/vehicleName/signals` 将车辆数据传输到 AWS IoT Core，后者会将数据发送到 AWS IoT FleetWise。AWS IoT FleetWise 然后将数据传输到 Timestream 表或 Amazon S3 存储桶。您可以使用 Timestream 来查询数据，也可以使用 Amazon QuickSight 或 Grafana 来显示数据。有关更多信息，请参阅[处理和可视化车辆数据](#)。

# 对车辆进行建模

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 [创建和管理信号目录](#)。

## 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 [创建和管理信号目录](#)。

## 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 [创建和管理车辆型号](#)。



## 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 [创建和管理解码器清单](#)。

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 [创建信号目录 \(AWS CLI\)](#) 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 [创建车辆型号 \(控制台\)](#)。
- 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 [创建车辆型号](#)。

### 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 [创建车辆型号 \(控制台\)](#)。
- 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 [更新车辆型号 \(AWS CLI\)](#)。
- 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 [创建和管理解码器清单](#)。

**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 [更新解码器清单 \(AWS CLI\)](#)。
- 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 文件来导入信号和解码器信号。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

- [创建和管理信号目录](#)
- [创建和管理车辆型号](#)
- [创建和管理解码器清单](#)

## 创建和管理信号目录

**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](#)。

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

- [配置信号](#)
- [创建信号目录 \(AWS CLI\)](#)
- [导入信号目录](#)
- [更新信号目录 \(AWS CLI\)](#)
- [删除信号目录 \(AWS CLI\)](#)
- [获取信号目录信息 \(AWS CLI\)](#)

## 配置信号

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`— 属性的数据类型必须是以下类型之一：  
INT8、UINT8、INT16、UINT16、INT32、UINT32、INT64、UINT64、BOOLEAN、FLOAT、DOUBLE、S  
或在数据类型分支中定义的自定义结构。`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、UINT8、INT16、UINT16、INT32、UINT32、INT64、UINT64、BOOLEAN、BOOLEAN、FLOAT、或在数据类型分支中定义的自定义结构。`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 - 信号的科学单位，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。属性代表结构体的成员，例如基元数据类型（例如 UINT8）或其他结构体（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

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、INT16、UINT16、INT32、UINT32、INT64、UINT64、BOOLEAN、FLOAT、DOUBLE、S 或 UNKNOWN。

- 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 )。

## 创建信号目录 (AWS CLI)

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 ) 将 `description` 替换为可帮助您识别信号目录的描述。

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

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。

## 导入信号目录

您可以使用 AWS 物联网 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 `description` 替换为可帮助您识别信号目录的描述。

- `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。

## 更新信号目录 (AWS CLI)

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[配置信号](#)。

### Important

自定义结构是不可变的。如果需要对现有自定义结构（结构体）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",
      "fullyQualifiedName": "Vehicle.Chassis.SteeringWheel.Diameter",
      "max": 55
    }
  }]
}
```

## 删除信号目录 (AWS CLI)

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：

- [删除车辆型号](#)

- [删除解码器清单](#)
- [删除车辆](#)
- [删除车队 \(AWS CLI\)](#)
- [删除活动](#)

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
```

#### Note

此命令不会生成任何输出。

## 获取信号目录信息 (AWS CLI)

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
```

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
```

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
```

 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。

## 创建和管理车辆型号

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 [创建、预配置和管理车辆](#)。

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

- 如果要使用 CreateModelManifest API 操作创建第一个车辆型号，则必须先创建信号目录。有关更多信息，请参阅 [创建信号目录 \(AWS CLI\)](#)。
- 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

- [创建车辆型号](#)
- [更新车辆型号 \(AWS CLI\)](#)
- [删除车辆型号](#)
- [获取车辆型号信息 \(AWS CLI\)](#)

## 创建车辆型号

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

### Important

在使用 CreateModelManifest API 操作创建车辆型号之前，您必须拥有信号目录。

### 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：

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

### 使用提供的模板 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 [创建和管理解码器清单](#)。

### 使用模板创建车辆型号

1. 导航到[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 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. 导航到[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 )。

- Data type – 信号的数据类型必须是以下类型之一：INT8、UINT8、INT16、UINT16、INT32、UINT32、INT64、UINT64、BOOLEAN、FLOAT、DOUBLE 或 UNKNOWN。
- 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。与 JSONPath 类似，使用点 (.) 来表示子信号。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 )

- Data type – 属性的数据类型必须是以下类型之一：  
INT8、UINT8、INT16、UINT16、INT32、UINT32、INT64、UINT64、BOOLEAN、FLOAT、DOUBLE 或 UNKNOWN
- ( 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。与 JSONPath 类似，使用点 (.) 来表示子信号。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. 导航到[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

如果要使用 FleetWise I AWS IoT API 创建第一个车辆模型，则必须先创建信号目录。有关如何创建信号目录的更多信息，请参阅 [创建信号目录 \(AWS CLI\)](#)。

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型号的名称。
- 将 *signal-catalog-ARN* 替换为信号目录的 Amazon 资源名称 (ARN)。
- ( 可选 ) 将 *description* 替换为可帮助您识别车辆型号的描述。

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 [配置信号](#)。

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

## 更新车辆型号 (AWS CLI)

您可以使用 [UpdateModelManifest](#) API 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 ) 将 *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"],  
}
```

## 删除车辆型号

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[删除车辆](#)和[删除解码器清单](#)：

### 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. 导航到[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 Delete (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
```

### Note

此命令不会生成任何输出。

## 获取车辆型号信息 (AWS CLI)

您可以使用 [ListModelManifests](#) API 操作来验证车辆模型是否已被删除。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
```

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
```

要检索有关车辆型号的信息，请运行以下命令。

将 *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。

## 创建和管理解码器清单

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。网络接口和解码器信号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 (ECU) 之间如何传输数据。ECU 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，定义了如何在 ECU 之间传输自我诊断数据。它提供了许多标准故障诊断代码 (DTC)，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。

## 解码器信号

提供特定信号的详细解码信息。车辆型号中指定的每个信号都必须与解码器信号配对。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，则它必须包含 OBD 解码器信号。

如果解码器清单还包含车辆中间件接口，则必须包含消息解码器信号。

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 API 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

- [配置网络接口和解码器信号](#)
- [创建解码器清单](#)
- [更新解码器清单 \(AWS CLI\)](#)

- [删除解码器清单](#)
- [获取解码器清单信息 \(AWS CLI\)](#)

## 配置网络接口和解码器信号

每个解码器清单都至少有一个网络接口和解码器信号，并与相关车辆型号中指定的信号配对。

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，则它必须包含 OBD 解码器信号。

主题

- [配置网络接口](#)
- [配置解码器信号](#)

### 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 – 在几秒钟内请求车辆故障诊断代码 (DTC) 的频率。例如，如果指定值为 120，则边缘代理软件会每 2 分钟收集一次存储的 DTC。

- ( 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) ISO15765-4 标准。
- ( 可选 ) pidRequestIntervalSeconds – 请求车辆 OBD II PID 的频率。例如，如果指定值为 120，则边缘代理软件会每 2 分钟收集一次 OBD II PID。
- ( 可选 ) useExtendedIds – 是否在消息中使用扩展 ID。

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：ROS\_2

## 配置解码器信号

要配置 CAN 解码器信号，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 – 消息的总字节长度。
- 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。

要配置 OBD 解码器信号，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。

要配置消息解码器信号，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。

## 创建解码器清单

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。

### 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 [创建和管理车辆型号](#)。

### 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. 导航到[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](#)
- [步骤 2：添加网络接口](#)
- [步骤 3：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。

### 步骤 2：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

- 在网络接口中，执行以下操作。
  - a. 对于网络接口类型，请选择 CAN\_INTERFACE 或 OBD\_INTERFACE。
  - b. 为网络接口输入唯一名称。

- c. 输入唯一的网络接口 ID。您可以使用 AWS 物联网生成的 ID FleetWise。
- d. 选择您的车辆型号中指定的一个或多个信号来与解码器信号配对。
- e. 要提供解码信息，请上传 .dbc 文件。AWS 物联网 FleetWise 解析.dbc 文件以检索解码器信号。
- f. 在配对信号部分，确保每个信号都与解码器信号配对。
- g. 选择下一步。

#### Note

- 您只能为每个网络接口上传一个 .dbc 文件。
- 确保您的车辆型号中指定的每个信号都与解码器信号配对。
- 选择添加其他网络接口后，您无法编辑正在编辑的网络接口。您可以删除任何现有的网络接口。

### 步骤 3：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，请先创建车辆型号。有关更多信息，请参阅 [创建车辆型号](#)。

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分解码器清单的名称。

- 将 *vehicle-model-ARN* 替换为车辆型号的 Amazon 资源名称 (ARN)。
- ( 可选 ) 将 *description* 替换为可帮助您识别解码器清单的描述。

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
      },
      "fullyQualifiedName": "Vehicle.EngineIdleTime",
      "interfaceId": "Qq1acaenBy0B3sSM39SYm",
      "type": "CAN_SIGNAL"
    },
    {
      "canSignal": {
        "name": "Engine_Run_Time",
        "factor": 1,
        "isBigEndian": true,
        "isSigned": false,
```

```

        "length": 24,
        "messageId": 271343712,
        "offset": 0,
        "startBit": 40
    },
    "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。
- 将 *vehicle-model-ARN* 替换为车辆型号的 Amazon 资源名称 (ARN)。
- ( 可选 ) 将 *description* 替换为可帮助您识别解码器清单的描述。

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"
  }, {
    "type": "VEHICLE_MIDDLEWARE",
    "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
    "vehicleMiddleware": {
      "name": "ROS2_test",
      "protocolName": "ROS_2"
    }
  }
  ],
  "signalDecoders": [{
    "canSignal": {

```

```

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

```

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

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

### Note

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。

## 更新解码器清单 (AWS CLI)

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
```

### ⚠ Important

激活解码器清单后，您无法对其进行编辑。

## 删除解码器清单

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. 导航到 [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
```

## 获取解码器清单信息 (AWS CLI)

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
```

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
```

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
```

您可以使用 [GetDecoderManifest](#) API 操作来验证解码器清单中的网络接口和解码器信号是否已更新。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。

将 *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。

# 创建、预配置和管理车辆

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

- [预配置车辆](#)
- [保留的主题](#)
- [创建车辆](#)
- [更新车辆 \(AWS CLI\)](#)
- [更新多辆车 \(AWS CLI\)](#)
- [删除车辆](#)
- [获取车辆信息 \(AWS CLI\)](#)

## 预配置车辆

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 东西。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 [创建车辆 \(AWS CLI\)](#)。

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。
  - 将 *policy-name* 替换为您要创建的策略的名称。

- 将###替换为包含策略的 JSON 文件的名称。AWS IoT Core

```
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：

- 将##替换为您创建 AWS 物联网 FleetWise资源的 AWS 区域。
- 将 *AWSAccount* 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 [保留的主题](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iot:Connect"
      ],
      "Resource": [
        "arn:aws:iot:region:awsAccount:client/
${iot:Connection.Thing.ThingName}"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "iot:Publish"
      ],
      "Resource": [
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/checkins",
        "arn:aws:iot:region:awsAccount:topic/$aws/iotfleetwise/vehicles/
${iot:Connection.Thing.ThingName}/signals"
      ]
    },
    {
```

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

## 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。
  - 将 `cert` 替换为保存 `certificatePem` 命令输出内容的文件名。
  - 将 `##` 替换为保存 `KeyPair` 命令输出内容的文件名。 `PublicKey`。

- 将##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。

- 将####替换为您创建的 AWS IoT Core 策略的名称。
- 将 *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。

- 将####替换为您的 AWS IoT 事物名称或车辆ID。
- 将 *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"
```

## 保留的主题

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。

主题	允许的客户端操作	描述
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> / checkins	Publish	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	Publish	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/vehicles/ <i>vehicleName</i> /collection_schemes	订阅	AWS IoT 为此主题 FleetWise 发布了数据收集方案。车辆会使用这些数据收集方案。
\$aws/iotfleetwise/vehicles/ <i>vehicleName</i> /decoder_manifests	订阅	AWS IoT 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。



# 创建车辆

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 [创建和管理车辆型号](#)。
- 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 [创建和管理解码器清单](#)。

## 主题

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

## 创建车辆 (控制台)

您可以使用 AWS 物联网 FleetWise 控制台创建车辆。

## ⚠ 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 [创建和管理车辆型号](#)。
- 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 [创建和管理解码器清单](#)。

## 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 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。

### 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 [创建和管理车辆型号](#)。
- 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 [创建和管理解码器清单](#)。

要创建车辆，请运行以下命令。

将 *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。将 *thing-name* 替换为您要创建的物品的名称。

```
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 [预配置车辆](#)。

- 将 *vehicle-name* 替换为以下名称之一。
  - 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 (:)。

- 将 *model-manifest-ARN* 替换为车辆型号 ( 模型清单 ) 的 ARN。
- 将 *decoder-manifest-ARN* 替换为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"
}
}
```

## 创建多辆车 (AWS CLI)

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。

将 *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 [创建车辆 \(AWS CLI\)](#)。

## 更新车辆 (AWS CLI)

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。

将 *file-name* 替换为包含车辆配置的 JSON 文件的名称。

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

### Example 车辆配置

- 将 *vehicle-name* 替换为您要更新的车辆的 ID。
- ( 可选 ) 将 *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 ) 将 *decoder-manifest-ARN* 替换为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

必须先为关联的车辆型号定义属性，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"
}
```

## 更新多辆车 (AWS CLI)

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。

将 *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[更新车辆 \(AWS CLI\)](#)。

## 删除车辆

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

### Important

删除车辆后，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[创建和管理车队](#)和[通过活动收集和传输数据](#)。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. 导航到[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。

将 *vehicle-name* 替换为您要删除的车辆的 ID。

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



## 获取车辆信息 (AWS CLI)

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
```

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。

将 *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。

# 创建和管理车队

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[通过活动收集和传输数据](#)。

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 提供以下 API 操作，您可以使用这些操作来创建和管理车队。

- [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](#) - 取消车辆与车队的关联。
- [UpdateFleet](#) - 更新现有车队的描述。
- [DeleteFleet](#) - 删除现有车队。
- [ListCampaigns](#) - 检索所有车队的分页摘要列表。
- [ListFleetsforVehicle](#) - 检索车辆所属的所有车队的分页 ID 列表。
- [ListVehiclesInFleet](#) - 检索车队中所有车辆的分页摘要列表。
- [GetFleet](#) - 检索有关车队的信息。

主题

- [创建车队 \(AWS CLI\)](#)
- [将车辆与车队关联 \(AWS CLI\)](#)
- [解除车辆与车队的关联 \(AWS CLI\)](#)
- [更新车队 \(AWS CLI\)](#)

- [删除车队 \(AWS CLI\)](#)
- [获取车队信息 \(AWS CLI\)](#)

## 创建车队 (AWS CLI)

您也可以使用 [CreateFleet](#) API 操作创建车队。下面的示例使用了 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[创建信号目录 \(AWS CLI\)](#)。

要创建车队，请运行以下命令。

- 将 *fleet-id* 替换为您要创建的车队的 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 ) 将 *description* 替换为描述。

描述可包含 1-2048 个字符。

- 将 *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
```

## 将车辆与车队关联 (AWS CLI)

您可以使用 [AssociateVehicleFleet](#) API 操作将车辆与车队关联起来。下面的示例使用了 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[创建、预配置和管理车辆](#)。

- 如果您将某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。

- 将 *fleet-id* 替换为车队的 ID。
- 将 *vehicle-name* 替换为车辆 ID。

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

## 解除车辆与车队的关联 (AWS CLI)

可以使用 [DisassociateVehicleFleet](#) API 操作来解除车辆与车队的关联。下面的示例使用了 AWS CLI。

要解除车辆与车队的关联，请运行以下命令。

- 将 *fleet-id* 替换为车队的 ID。
- 将 *vehicle-name* 替换为车辆 ID。

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

## 更新车队 (AWS CLI)

可以使用 [UpdateFleet](#) API 操作来更新车队的描述。下面的示例使用了 AWS CLI。

要更新车队，请运行以下命令。

- 将 *fleet-id* 替换为您要更新的车队的 ID。
- 将 *description* 替换为新描述。

描述可包含 1-2048 个字符。

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

## 删除车队 (AWS CLI)

您可以使用 [DeleteFleet](#) API 操作来删除车队。下面的示例使用了 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[解除车辆与车队的关联 \(AWS CLI\)](#)。

要删除车队，请运行以下命令。

将 *fleet-id* 替换为您要删除的车队的 ID。

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

## 获取车队信息 (AWS CLI)

可以使用 [ListFleets](#) API 操作来验证车队是否已被删除。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
```

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

要检索车辆所属的所有车队的分页 ID 列表，请运行以下命令。

将 *vehicle-name* 替换为车辆 ID。

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

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。

将 *fleet-id* 替换为车队的 ID。

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

可以使用 [GetFleet](#) API 操作来检索车队信息。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。

将 *fleet-id* 替换为车队的 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。

## 通过活动收集和传输数据

活动是指数据收集规则的编排。活动向适用于 AWS IoT FleetWise 的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。

您要在云中创建活动。在您或您的团队批准活动以后，AWS IoT FleetWise 会自动将其部署到车辆。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。

### Note

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. 导航到 [AWS IoT FleetWise 控制台](#)。
  2. 在服务主页上的开始使用 AWS IoT FleetWise 部分，选择浏览边缘代理。
- 您已将 AWS IoT Core 设置为预配置您的车辆。有关更多信息，请参阅 [预配置车辆](#)。

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。AWS IoT 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哪些数据。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[活动的逻辑表达式](#)。

( 可选 ) `triggerMode` 可以是以下值之一。

- `RISING_EDGE` - 仅在第一次满足条件时，边缘代理软件才会收集数据。例如，`$variable.`myVehicle.AirBagDeployed` == true`。
- `ALWAYS` - 满足条件时，边缘代理软件会收集数据。

( 可选 ) `minimumTriggerIntervalMs`

两次收集数据事件之间的最短时长，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 ) `conditionLanguageVersion`

条件表达式语言的版本。

## 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。

( 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

选择活动将传输车辆数据的目标。您可以选择在 Amazon S3 或 Amazon Timestream 中存储数据。

S3 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。

( 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 ) `description`

您可以添加描述，以帮助标识活动的目的。

( 可选 ) `diagnosticsMode`

将诊断模式配置为 `SEND_ACTIVE_DTCS` 后，活动会发送存储的标准诊断故障代码 (DTC)，以帮助识别您的车辆出了什么问题。例如，`P0097` 表示引擎控制模块 (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`

您可以为活动定义到期日期。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`

您可以指定一个整数来表示活动的优先级。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。

**⚠ Important**

必须在此字段中指定基于条件的收集方案的表达式中使用的信号。

**name**

调用数据收集方案时从中收集数据的信号的名称。

**( 可选 ) maxSampleCount**

调用数据收集方案时，边缘代理软件收集并传输到云端的最大数据样本数。

**( 可选 ) minimumSamplingIntervalMs**

两次数据样本收集事件之间的最短时长，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 TO\_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 正在处理您发出的活动创建请求。
- 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[更新活动 \(AWS CLI\)](#)。
- 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](#) - 创建新活动。
- [UpdateCampaign](#) - 更新现有活动。创建活动后，您必须使用此 API 操作来批准该活动。
- [DeleteCampaign](#) - 删除现有活动。
- [ListCampaigns](#) - 检索所有活动的分页摘要列表。
- [GetCampaign](#) - 检索有关活动的信息。

## 教程

- [创建活动](#)
- [更新活动 \(AWS CLI\)](#)
- [删除活动](#)
- [获取活动信息 \(AWS CLI\)](#)

## 创建活动

您可以使用 AWS IoT FleetWise 控制台或 API 通过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. 导航到 [AWS IoT FleetWise 控制台](#)。
  2. 在服务主页上的开始使用 AWS IoT FleetWise 部分，选择浏览边缘代理。
- 您已将 AWS IoT Core 设置为预配置您的车辆。有关更多信息，请参阅[预配置车辆](#)。

## 主题

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

## 创建活动 ( 控制台 )

您可以使用 AWS IoT 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. 导航到 [AWS IoT 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](#)
- [步骤 2：定义存储目标](#)
- [步骤 3：添加车辆](#)
- [步骤 4：审核并创建](#)
- [步骤 5：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 [创建和管理信号目录](#) 和 [创建、预配置和管理车辆](#)：
- 创建活动后，必须批准该活动。有关更多信息，请参阅 [步骤 5：部署活动](#)。

### 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 IoT FleetWise 控制台中，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，则必须定义一个逻辑表达式来识别要收集哪些数据。AWS IoT FleetWise 使用逻辑表达式来识别在采用基于条件的方案时要收集哪些数据。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，如果指定表达式 `$variable.`myVehicle.InVehicleTemperature` > 50.0`，则 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[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) 可以指定条件表达式的语言版本。默认值是 1。
5. (可选) 您可以指定最小触发间隔，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，适用于 AWS IoT FleetWise 的边缘代理软件在满足条件时始终收集数据。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) 您可以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，可以指定活动优先级。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的选项。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 [CreateCampaign](#) API 文档。

#### Note

AWS IoT FleetWise 目前支持文件格式扩展名 .json。

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

## 指定信号

您可以指定调用数据收集方案时要从中收集数据的信号。

#### Important

必须在此字段中指定基于条件的收集方案的表达式中使用的信号。

## 指定要从中收集数据的信号

1. 搜索信号的完全限定名称。

**Note**

信号的完全限定名称是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。

2. (可选) 对于最大样本计数，请输入调用数据收集方案时，边缘代理软件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 2：定义存储目标

**Note**

只有当活动包含视觉系统数据信号时，才能将车辆数据传输到 Amazon S3。  
视觉系统数据目前为预览版，可能会发生变化。

选择要存储活动收集到的数据的目标。您可以将车辆数据传输到 Amazon S3 或 Amazon Timestream。

在目标设置中，执行以下操作：

- 从下拉列表选择 S3 或 Timestream。

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《Amazon Simple Storage Service 用于指南》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。

## Amazon S3

### Important

仅当 AWS IoT FleetWise 有权写入 S3 存储桶时，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 FleetWise 控制访问权限](#)。

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。

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《Amazon Timestream 开发人员指南》中的[什么是 Amazon Timestream？](#)

## Amazon Timestream

### Important

仅当 AWS IoT FleetWise 有权将数据写入 Timestream 时，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 FleetWise 控制访问权限](#)。

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。
- 选择下一步。

### 步骤 3：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。

### 步骤 4：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。

## 步骤 5：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)

您也可以使用 [CreateCampaign](#) API 操作创建活动。下面的示例使用了 AWS CLI。

创建活动，从车辆收集的数据可以存储在 Amazon 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 以获得具有行业领先的可扩展性、数据可用性、安全性和性能的对象存储。

### ⚠ Important

仅当 AWS IoT FleetWise 有权将数据写入 S3 或 Timestream 时，才能传输车辆数据。有关授予访问权限的更多信息，请参阅[使用 AWS IoT FleetWise 控制访问权限](#)。

## 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[创建和管理信号目录](#)、[创建、预配置和管理车辆](#)和[创建和管理车队](#)。
- 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[更新活动 \(AWS CLI\)](#)。

要创建活动，请运行以下命令。

将 *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
```

- 将 *campaign-name* 替换为您正在创建的活动的名称。
- 将 *signal-catalog-arn* 替换为信号目录的 Amazon 资源名称 ( ARN )。
- 将 *target-arn* 替换为您创建的车队或车辆的 ARN。
- 将 *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"
    }
  }
]
}

```

- 将 *campaign-name* 替换为您正在创建的活动的名称。
- 将 *signal-catalog-arn* 替换为信号目录的 Amazon 资源名称 ( ARN )。
- 将 *target-arn* 替换为您创建的车队或车辆的 ARN。
- 将 *role-arn* 替换为任务执行角色的 ARN，该角色授予 AWS IoT FleetWise 向 Timestream 表传送数据的权限。
- 将 *table-arn* 替换为 Timestream 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"
    }
  }
]
}
```

## 活动的逻辑表达式

AWS IoT FleetWise 使用逻辑表达式来识别活动中收集的数据。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`
}

```

## 更新活动 (AWS CLI)

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

- 将 *campaign-name* 替换为您要更新的活动的名称。
- 将 *action* 替换为以下项之一：
  - APPROVE - 批准活动，以允许 AWS IoT FleetWise 将其部署到车辆或车队。
  - 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。该活动将重新部署到所有车辆，车辆将恢复发送数据。
- UPDATE - 通过定义属性并将其与信号关联来更新活动。

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

## 删除活动

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

### 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. 导航到 [AWS IoT 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 操作来删除活动。下面的示例使用了 AWS CLI。

要创建活动，请运行以下命令。

将 *campaign-name* 替换为要删除的车辆名称。

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

### 获取活动信息 (AWS CLI)

您可以使用 [ListCampaigns](#) API 操作来验证活动是否已被删除。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
```

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。

将 *campaign-name* 替换为您要检索的活动的名称。

```
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。



## 处理和可视化车辆数据

适用于 AWS IoT FleetWise 的边缘代理软件可将选定的车辆数据传输到 Amazon Timestream 或 Amazon Simple Storage Service (Amazon S3)。数据到达数据目的地后，您可以使用其他 AWS 服务对其进行可视化和共享。

### 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。您的数据存储在与客户管理的 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	描述
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。

## 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）和其他数据源的数据。
- 使用 [Amazon QuickSight](#) 在控制面板中分析和可视化数据。

## 在 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。S3 可用于数据湖、备份和恢复、归档、企业应用程序、AWS IoT 设备和大数据分析等各种用例。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	描述
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。

字段名称	数据类型	描述
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。

## S3 对象格式

AWS IoT FleetWise 将车辆数据传输到 S3，并在 S3 上另存为对象。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 IoT 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。在将对象写入 S3 之前，AWS IoT FleetWise 默认情况下会添加格式为 year=YYYY/month=MM/date=DD/hour=HH 的 UTC 时间前缀。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 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 开发人员指南》中的以下主题：

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

使用 AWS Glue 爬网程序对数据进行分类并在 Amazon Athena 中进行分析。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 AWS Glue 连接到 Amazon S3 中的数据来源](#)
- [将 Athena 与 AWS Glue 结合使用时的最佳实践](#)

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

**Tip**

如果直接从 S3 读取，请确认您的车辆数据采用 JSON 格式，因为 Amazon QuickSight 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](#)

# AWS CLI 和 AWS SDK

本节提供有关发出 AWS IoT FleetWise API 请求的信息。有关 AWS IoT FleetWise [操作和数据类型](#) 的更多信息，请参阅《AWS IoT FleetWise API 参考》。

要将 AWS IoT FleetWise 与各种编程语言结合使用，可以使用包含以下自动功能的 [AWS SDK](#)：

- 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](#) 结合使用。您可以从命令行控制 AWS IoT FleetWise 和您的其他服务，并通过脚本自动执行这些服务。

# 排查 AWS IoT FleetWise 问题

使用本节中的故障排除信息和解决方案可帮助解决 AWS IoT FleetWise 问题。

以下信息可帮助您处理 AWS IoT FleetWise 中的常见问题。

## 主题

- [解码器清单问题](#)
- [适用于 AWS IoT FleetWise 的边缘代理软件问题](#)

## 解码器清单问题

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 IoT FleetWise 正在尝试根据其他人修改的车辆模型（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 IoT FleetWise 找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) 密钥存在权限问题。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
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</pre>	<p>AWS IoT FleetWise 使用信号目录未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。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</pre>	<p>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。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</pre>	<p>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。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>AWS IoT FleetWise 在信号目录中发现了一个定义为 STRUCT 的信号，但解码器清单请求中未定义 structuredMessageDefinition。确保在解码器清单更新请求中将每个结构定义为 structure dMessageDefinition。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>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字段 structFullyQualifiedName。它还需要一个具有该 fullyQualifiedName 的 STRUCT 节点。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>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。</p>
<pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>	<p>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 structuredMessageListDefinition。确</p>

错误	故障排除指南
ason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)	认解码器清单更新请求中的所有数组属性是否都定义为 structuredMessageListDefinition。

## 适用于 AWS IoT FleetWise 的边缘代理软件问题

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：边缘代理软件不收集机载诊断 \(OBD\) II PID 和故障诊断代码 \(DTC\)。](#)
- [问题：适用于 AWS IoT FleetWise 的边缘代理软件无法从网络中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 \[警告\] \[AwsIotChannel::send:: send\]: \[无有效 MQTT 连接。\]](#)

### 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：确保适用于 AWS IoT FleetWise 的边缘代理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。有关该配置文件的更多信息，请执行以下操作，下载《适用于 AWS IoT FleetWise 的边缘代理软件开发人员指南》。

1. 导航到 [AWS IoT FleetWise 控制台](#)。
2. 在服务主页上的开始使用 AWS IoT FleetWise 部分，选择浏览边缘代理。

- ```
[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每个网络接口是否可用，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。将 *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。

- `mqtConnection::endpointUrl` – AWS 账户的物联网设备端点。
- `mqtConnection::clientId` – 运行边缘代理软件的车辆的 ID。
- `mqtConnection::certificateFilename` – 车辆证书文件的路径。
- `mqtConnection::privateKeyFilename` – 车辆私钥文件的路径。
- 您曾经使用 AWS IoT Core 调配过车辆。有关更多信息，请参阅[预配置车辆](#)。

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：边缘代理软件不收集车载诊断 (OBD) II PID 和故障诊断代码 (DTC)。

**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 ID，请确保 `obdInterface:useExtendedIds` 配置为 `true`。

**问题：**适用于 AWS IoT FleetWise 的边缘代理软件无法从网络中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	可以	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	可以	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	可以	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:: send]: [无有效 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，边缘代理软件每分钟向 AWS IoT Core 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。

# 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 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设施上的内容的控制。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](#)

[题](#)。有关欧洲数据保护的信息，请参阅 AWS 安全性博客 上的 [AWS 责任共担模式和 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。我们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)。
- 使用 SSL/TLS 与资源通信。AWS 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。
- 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 140-2 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 140-2 版》](#)。

我们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 SDK AWS 服务使用 AWS 物联网或其他设备时。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网址，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。

## 静态加密

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。

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”](#)。

## 数据加密

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 物联网 FleetWise 边缘处理公开了托管在 AWS 物联 FleetWise 网网关中且可通过本地网络访问的 API。这些 API 通过由 I AWS oT 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，这些 API 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

- [静态加密](#)
- [密钥管理](#)

## 静态加密

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。

### 工作方式

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 [开发人员指南](#) 中的 AWS Key Management Service 信封加密。

以下示例策略向 AWS 物联网 FleetWise 授予代表您创建客户托管密钥的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1603902045292",
      "Action": [
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:DescribeKey",
        "kms:CreateGrant",
        "kms:RetireGrant",
        "kms:RevokeGrant"
      ],
      "Effect": "Allow",
      "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[密钥管理](#)。

## 密钥管理

### 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		描述
	属性	description、allowedValues、defaultValue、min、max
	执行器	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		description
	CanInterface	protocolName、protocolVersion
	ObdInterface	requestMessageId、dtcRequestInterval秒、OBDSStandard 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 pidRespon

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
		seLength、startByte、bitMaskLength bitRightShift
车辆		attributes
活动		description
	conditionBasedCollection计划	表情、minimumTriggerInterval Ms conditionLanguageVersion、TriggerMode
	TimeBasedCollectionScheme	periodMS

#### 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。

- 将 **KMS ## ID** 替换为 KMS 密钥的 ID。

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

## 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。

```
{
  "Sid": "Allow FleetWise to encrypt and decrypt data when customer managed KMS key
based encryption is enabled",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:CreateGrant",
    "kms:RetireGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*"
}
```

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。

## 使用控制访问权限 AWS IoT 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

何授予对您的 Amazon S3 (S3) 存储桶或 Amazon Timestream 数据库和表的 AWS IoT FleetWise 访问权限以存储数据。

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](#)。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
```

```

    "Effect": "Allow",
    "Principal": {
      "Service": [
        "iotfleetwise.amazonaws.com"
      ]
    },
    "Action": [
      "s3:GetObject",
      "s3:PutObject"
    ],
    "Resource": "arn:aws:s3:::bucket-name/*",
    "Condition": {
      "StringEquals": {
        "aws:SourceArn": "campaign-arn",
        "aws:SourceAccount": "account-id"
      }
    }
  }
]
}

```

以下存储桶政策适用于某个 AWS 地区中一个账户中的所有广告活动。

```

{
  "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:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*"
    }
  ]
}

```



```

    ]
  },
  "Action": [
    "s3:GetObject",
    "s3:PutObject"
  ],
  "Resource": "arn:aws:s3:::bucket-name/*",
  "Condition": {
    "StringLike": {
      "aws:SourceArn": "arn:aws:iotfleetwise:region:account-id:campaign/*",
      "aws:SourceAccount": "account-id"
    }
  }
}
]
}

```

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 Simple Storage Service 用户指南》中的[禁用所有新存储桶的 ACL 并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。

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 美国东部（弗吉尼亚北部）和欧洲（法兰克福）提供物联网 FleetWise。
  - 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《Amazon Timestream 开发人员指南》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《Amazon Timestream 开发人员指南》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。
- 将 `####` 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
```

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

```

"Statement": [
  {
    "Sid": "timestreamTrustPolicy",
    "Effect": "Allow",
    "Principal": {
      "Service": "iotfleetwise.amazonaws.com"
    },
    "Action": "sts:AssumeRole",
    "Condition": {
      "StringEquals": {
        "aws:SourceArn": [
          "arn:aws:iotfleetwise:region:account-id:campaign/campaign-name"
        ],
        "aws:SourceAccount": [
          "account-id"
        ]
      }
    }
  }
]
}

```

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。
- 将 *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

```

```

{
  "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](#)。

## 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](#)

## 受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您在 AWS 物联网中所做的工作 FleetWise。

**服务用户**-如果您使用 AWS 物联网 FleetWise 服务完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多的 AWS IoT FleetWise 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS IoT 中的某项功能 FleetWise，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**-如果您负责公司的物 AWS 联网 FleetWise 资源，则可能拥有对 AWS 物联网的完全访问权限 FleetWise。您的工作是确定您的服务用户应访问哪些 AWS 物联网 FleetWise 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何将 IAM 与 I AWS oT 结合使用 FleetWise，请参阅[AWS 物联网如何 FleetWise 与 IAM 配合使用](#)。

**IAM 管理员** — 如果您是 IAM 管理员，则可能需要详细了解如何编写策略来管理 AWS IoT 访问权限 FleetWise。要查看您可以在 AWS IAM 中使用的 FleetWise 基于 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](#)

## 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 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。AWS IAM Identity Center（IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的[如何登录到您 AWS 账户的](#)。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的[签署 AWS API 请求](#)。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如，AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的[多重身份验证](#)和《IAM 用户指南》中的[在 AWS 中使用多重身份验证 \(MFA\)](#)。

## AWS 账户 root 用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建帐户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，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（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity Center 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。AWS Directory Service 当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和应用程序中使用。有关 IAM Identity Center 的信息，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](#)是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的[对于需要长期凭证的使用场景定期轮换访问密钥](#)。

[IAM 组](#)是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅《IAM 用户指南》中的[何时创建 IAM 用户（而不是角色）](#)。

## IAM 角色

[IAM 角色](#)是您内部具有特定权限 AWS 账户的身份。它类似于 IAM 用户，但与特定人员不关联。您可以使用 AWS Management Console 通过[切换角色在中临时担任 IAM 角色](#)。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的[使用 IAM 角色](#)。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的[为第三方身份提供商创建角色](#)。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的[权限集](#)。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。
- 跨服务访问 — 有些 AWS 服务使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
- 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅[转发访问会话](#)。
- 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 [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](#)。



- 服务相关角色-服务相关角色是一种链接到的服务角色。AWS 服务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相关角色的权限。
- 在 Amazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 [使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限](#)。

要了解是使用 IAM 角色还是 IAM 用户，请参阅《IAM 用户指南》中的 [何时创建 IAM 角色 \(而不是用户\)](#)。

## 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。策略是其中的一个对象 AWS，当与身份或资源关联时，它会定义其权限。AWS 在委托人 (用户、root 用户或角色会话) 发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 [JSON 策略概览](#)。

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。

默认情况下，用户和角色没有权限。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。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

## 基于身份的策略

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 [创建 IAM 策略](#)。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括



AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的[在托管式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。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service ( 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。

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。

## 访问控制列表 (ACL)

访问控制列表 ( ACL ) 控制哪些主体 ( 账户成员、用户或角色 ) 有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。AWS WAF 要了解有关 ACL 的更多信息，请参阅《Amazon Simple Storage Service 开发人员指南》中的[访问控制列表 \( ACL \) 概览](#)。

## 其他策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级功能，用于设置基于身份的策略可以为 IAM 实体 ( IAM 用户或角色 ) 授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界的交集。在 Principal 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅《IAM 用户指南》中的[IAM 实体的权限边界](#)。
- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有功能，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体 ( 包括每个 AWS 账户根用户实体 ) 的权限。有关 Organizations 和 SCP 的更多信息，请参阅《AWS Organizations 用户指南》中的[SCP 的工作原理](#)。
- 会话策略 - 会话策略是当您以编程方式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。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的

策略。任一项策略中的显式拒绝将覆盖允许。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="#">ACL</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="#">服务相关角色</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[创建 IAM 策略](#)。

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。您无法在基于身份的策略中指定主体，因为它适用于其附加的用户或角色。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 Simple Storage Service ( 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。

要启用跨账户存取，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。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源处于不同位置时 AWS 账户，可信账户中的 IAM 管理员还必须向委托人实体（用户或角色）授予访问资源的权限。他们通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的[IAM 角色与基于资源的策略有何不同](#)。

## 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。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

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。语句必须包含 Resource 或 NotResource 元素。作为最佳实践，请使用其 [Amazon 资源名称 \(ARN\)](#) 指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），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 ARN 的列表，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。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。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。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑 OR 运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 [IAM 策略元素：变量和标签](#)。

AWS 支持全局条件密钥和特定于服务的条件密钥。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](#)

## 物联网中的 AWS 访问控制列表 (ACL) FleetWise

支持 ACL	否
--------	---

访问控制列表 ( ACL ) 控制哪些主体 ( 账户成员、用户或角色 ) 有权访问资源。ACL 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
--------------------	----

基于属性的访问权限控制 ( ABAC ) 是一种授权策略，该策略基于属性来定义权限。在中 AWS，这些属性称为标签。您可以将标签附加到 IAM 实体 ( 用户或角色 ) 和许多 AWS 资源。标记实体和资源是 ABAC 的第一步。然后设计 ABAC 策略，以在主体的标签与他们尝试访问的资源标签匹配时允许操作。

ABAC 在快速增长的环境中非常有用，并在策略管理变得繁琐的情况下可以提供帮助。

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 [什么是 ABAC ?](#)。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**

当你使用临时证书登录时，有些 AWS 服务 不起作用。有关更多信息，包括哪些 AWS 服务 适用于临时证书，请参阅 IAM 用户指南中的[AWS 服务与 IAM 配合使用的信息](#)。

如果您使用除用户名和密码之外的任何方法登录，则 AWS Management Console 使用的是临时证书。例如，当您 AWS 使用公司的单点登录 (SSO) 链接进行访问时，该过程会自动创建临时证书。当您以用户身份登录控制台，然后切换角色时，您还会自动创建临时凭证。有关切换角色的更多信息，请参阅《IAM 用户指南》中的[切换到角色 \(控制台\)](#)。

您可以使用 AWS CLI 或 AWS API 手动创建临时证书。然后，您可以使用这些临时证书进行访问 AWS。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 中的临时安全凭证](#)。

## IoT 的跨服务主体 AWS 权限 FleetWise

支持转发访问会话 (FAS) **是**

当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，然后此操作在其他服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。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 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相关角色的权限。

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 [使用 Amazon CloudWatch 监控 AWS IoT FleetWise](#)。

服务相关角色可以更 FleetWise 快地设置 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，您才能删除服务相关角色。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](#)，并查找服务相关角色列中显示为是的服务。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 使用名为的服务相关角色 `AWSServiceRoleForIoT FleetWise`——一个 AWS 托管策略，用于 AWS IoT FleetWise 的所有 out-of-the-box 权限。

`AWSServiceRoleForIoT FleetWise` 服务相关角色信任以下服务来代入该角色：

- `IoTFleetWise`

名为的角色权限策略 `AWSIoT FleetWise ServiceRolePolicy` 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`

您必须配置权限，允许 IAM 实体（如用户、组或角色）创建、编辑或删除服务相关角色。有关更多信息，请参阅《IAM 用户指南》中的[服务相关角色权限](#)。

### 为 AWS IoT 创建服务相关角色 FleetWise

您无需手动创建服务相关角色。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相关角色。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 `AWSServiceRoleForIoT FleetWise` 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。他们也无法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 执行任务。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。管理员随后可以向角色添加 IAM 策略，用户可以代入角色。

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](#)。

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 ARN 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](#)。

### 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，请遵循以下准则和建议：

- 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 AWS 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 [IAM Access Analyzer 策略验证](#)。
- 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 [配置受 MFA 保护的 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 中的安全最佳实操](#)。

## 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:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

## 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（包括程序和示例策略），请参阅[配置设置](#)。

## 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 Management Console 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，可以担任角色。对于支持基于资源的策略或访问控制列表 ( ACL ) 的服务，您可以使用这些策略向人员授予对您的资源的访问权。

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存取之间的差别，请参阅《IAM 用户指南》中的 [IAM 角色与基于资源的策略有何不同](#)。

## 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。AWS 提供了以下资源来帮助实现合规性：



- [安全与合规性快速入门指南](#) — 这些部署指南讨论了架构注意事项，并提供了在这些基础上 AWS 部署以安全性和合规性为重点的基准环境的步骤。
- 在 [Amazon Web Services 上构建 HIPAA 安全与合规性](#) — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

 Note

并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅[符合 HIPAA 要求的服务参考](#)。

- [AWS 合规资源AWS](#) — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- [AWS 客户合规指南](#) — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践，AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- [使用AWS Config 开发人员指南中的规则评估资源](#) — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- [AWS Security Hub](#) — 这 AWS 服务 可以全面了解您的安全状态 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 [Security Hub 控件参考](#)。
- [Amazon GuardDuty](#) — 它通过监控您的 AWS 账户环境中是否存在可疑和恶意活动，来 AWS 服务检测您的工作负载、容器和数据面临的潜在威胁。GuardDuty 通过满足某些合规性框架规定的入侵检测要求，可以帮助您满足各种合规性要求，例如 PCI DSS。
- [AWS Audit Manager](#) — 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

## 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。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、容错性和可扩展性。

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](#)。

## 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。

此外，必须使用访问密钥 ID 和与 IAM 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 [AWS Security Token Service](#) (AWS STS) 生成临时安全凭证来对请求进行签名。

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 VPC 的访问。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，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 端点都由您的 VPC 子网中一个或多个使用私有 IP 地址的[弹性网络接口](#) ( ENI ) 代表。

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 AWS 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 内部的实例进行连接，或者使用 AWS Virtual Private Network (VPN) 或将您的私有网络连接到 VPC AWS 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《AWS 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 的默认服务名称终端节点采用以下格式。

```
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 [终端节点 \(AWS PrivateLink\)](#)。

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](#)。

## 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 AWS 账户 **123456789012** 使用该终端节点进行所有 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 **123 456789012** 中的用户 **lijuan** 拥有完全访问权限。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。##### IAM ## FleetWise #  
123456789012 ##### AWS ### FleetWise ##### AWS ##

```
{
  "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。以下最佳实操是一般准则，并不代表完整的安全解决方案。这些最佳实操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 AWS 物联网 FleetWise 资源的名称和 ID 中使用敏感信息，例如在活动名称、解码器清单、车辆型号和信号目录中，或者在车辆和车队的 ID 中。

## 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 oT 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 oT API 调用的历史记录，只需在 CloudTrail 中开启即可 AWS Management Console。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 IoT FleetWise

监控是保持 AWS IoT FleetWise 和您的其他 AWS 解决方案的可靠性、可用性和性能的重要方面。AWS 提供了以下一些监控工具来监控 AWS IoT FleetWise、在出现错误时进行报告并适时自动采取措施。

- Amazon CloudWatch 实时监控您的 AWS 资源以及在 AWS 上运行的应用程序。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，您可以使用 CloudWatch 跟踪 Amazon EC2 实例的 CPU 使用率或其他指标并且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 看用于监控、存储和访问来自 Amazon EC2 实例、CloudTrail 和其他来源的日志文件。CloudWatch Logs 可以监控日志文件中的信息，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 Logs 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。您可以标识哪些用户和账户调用了 AWS、发出调用的源 IP 地址以及调用的发生时间。有关更多信息，请参阅 [AWS CloudTrail 《用户指南》](#)。

## 使用 Amazon CloudWatch 监控 AWS IoT FleetWise

Amazon CloudWatch 指标是一种监控 AWS 资源及其执行情况的方式。AWS IoT FleetWise 会向 CloudWatch 发送指标。可以使用 AWS Management Console、AWS CLI 或 API 列出 AWS IoT FleetWise 发送到 CloudWatch 的指标。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 CloudWatch 发送指标。有关更多信息，请参阅 [配置设置](#)。

AWS/IoTFleetWise 命名空间包括以下指标。

## 信号指标

指标	描述
IllegalMessageFromEdge	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息与要求的格式不匹配。</p> <p>单位：计数</p> <p>维度：VehicleName</p> <p>有效统计数据：Sum</p>
MessageThrottled	<p>车辆向 AWS IoT FleetWise 发送的消息已被限制。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：VehicleName</p> <p>有效统计数据：Sum</p>
ModelingError	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息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>维度：ModelManifestName</p> <p>有效统计数据：Sum</p>
DecodingError	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息包含无法根据车辆解码器清单进行解码的信号。</p> <p>单位：计数</p> <p>维度：DecoderName</p> <p>有效统计数据：Sum</p>

## 活动指标

指标	描述
VehicleNotFound	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息，其中车辆未知。</p> <p>单位：计数</p> <p>维度：VehicleName</p> <p>有效统计数据：Sum</p>
CampaignInvalid	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息，其中活动无效。</p> <p>单位：计数</p> <p>维度：CampaignName</p> <p>有效统计数据：Sum</p>
CampaignNotFound	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息，其中活动未知。</p> <p>单位：计数</p> <p>维度：CampaignName</p> <p>有效统计数据：Sum</p>

## 活动数据目标指标

指标	描述
TimestreamWriteError	<p>AWS IoT FleetWise 无法将来自车辆的消息写入 Amazon Timestream 表。</p> <p>单位：计数</p> <p>维度：DatabaseName、TableName</p>

指标	描述
	有效统计数据 : Sum
S3WriteError	<p>AWS IoT FleetWise 无法将来自车辆的消息写入 Amazon Simple Storage Service (Amazon S3) 存储桶。</p> <p>单位 : 计数</p> <p>维度 : BucketName</p> <p>有效统计数据 : Sum</p>
S3ReadError	<p>AWS IoT FleetWise 无法从 Amazon Simple Storage Service (Amazon S3) 存储桶中的车辆读取对象键。</p> <p>单位 : 计数</p> <p>维度 : BucketName</p> <p>有效统计数据 : Sum</p>

### 客户托管的 AWS KMS 关键指标

指标	描述
KMSKeyAccessDenied	<p>由于 AWS KMS 密钥访问遭拒错误，AWS IoT FleetWise 无法将来自车辆的消息写入 Timestream 表或 Amazon S3 存储桶。</p> <p>单位 : 计数</p> <p>维度 : KMSKeyId</p> <p>有效统计数据 : Sum</p>



# 使用 Amazon CloudWatch Logs 监控 AWS IoT FleetWise

Amazon CloudWatch 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 Logs 用户指南](#)。

## 在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志

### Important

您需要满足以下条件，然后才能在 CloudWatch 控制台中查看 AWS IoT FleetWise 日志组：

- 您已在 AWS IoT FleetWise 中启用日志记录功能。有关日志记录的更多信息，请参阅 [配置 AWS IoT FleetWise 日志记录](#)。
- 已经有 AWS IoT 操作写入的日志条目。

在 CloudWatch 控制台中查看 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" }
```

有关更多信息，请参阅《Amazon CloudWatch Logs 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	描述
MODELING_ERROR	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息包含无法根据车辆模型进行验证的信号。</p> <p>属性：vehicleName、campaignName、signalCatalogName、signalId、signalValue、signalValueRangeMin、signalValueRangeMax、modelManifestName</p>
ILLEGAL_MESSAGE_FROM_EDGE	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息与要求的格式不匹配。</p> <p>属性：vehicleName、campaignName、signalCatalogName</p>
DECODING_ERROR	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息包含无法根据车辆解码器清单进行解码的信号。</p> <p>属性：campaignName、signalCatalogName、decoderManifestName、( 可选 ) signalName、( 可选 ) s3URI</p>

## 活动事件类型

事件类型	描述
VEHICLE_NOT_FOUND	<p>从车辆发送并由 AWS IoT FleetWise 接收的消息，其中车辆未知。</p>

事件类型	描述
	属性 : vehicleName、campaignName
CAMPAIGN_NOT_FOUND	从车辆发送并由 AWS IoT FleetWise 接收的消息，其中活动未知。  属性 : vehicleName ( 可选 )、campaignName
CAMPAIGN_INVALID	从车辆发送并由 AWS IoT FleetWise 接收的消息，其中活动无效。  属性 : vehicleName ( 可选 )、campaignName

### 活动数据目标事件类型

事件类型	描述
TIMESTREAM_WRITE_ERROR	AWS IoT FleetWise 无法将来自车辆的消息写入 Amazon Timestream 表。  属性 : vehicleName、campaignName、timestreamDatabaseName、timestreamTableName
S3_WRITE_ERROR	AWS IoT FleetWise 无法将来自车辆的消息写入 Amazon Simple Storage Service (Amazon S3) 存储桶。  属性 : campaignName、destinationName
S3_READ_ERROR	AWS IoT FleetWise 无法从 Amazon Simple Storage Service (Amazon S3) 存储桶中的车辆读取对象键。  属性 : campaignName、destinationName

## 客户托管的 AWS KMS 密钥事件类型

事件类型	描述
KMS_KEY_ACCESS_DENIED	由于 AWS KMS 密钥访问遭拒错误，AWS IoT FleetWise 无法将来自车辆的消息写入 Timestream 表或 Amazon S3 存储桶。

## 属性

所有 CloudWatch Logs 条目都包含以下这些属性：

accountId

您的 AWS 账户 ID。

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](#) 消息。

包含有关日志的具体详情。

timestamp

AWS IoT FleetWise 处理日志时的纪元毫秒时间戳。

## 可选属性

CloudWatch Logs 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的名称。

**signalId**

错误信号的 ID。

**signalIds**

错误信号 ID 列表。

**signalName**

信号的名称。

**signalTimestampEpochMs**

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。

## 配置 AWS IoT FleetWise 日志记录

您可以将 AWS IoT FleetWise 日志数据发送到 CloudWatch 日志组。CloudWatch Logs 可在 AWS IoT FleetWise 无法处理来自车辆的消息时提供可见性。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。

您必须创建一个 CloudWatch 日志组，然后才能向 CloudWatch 发送日志。使用与 AWS IoT FleetWise 中使用的同一账户并在同一区域配置日志组。在 AWS IoT FleetWise 中启用日志记录功能时，请提供日志组名称。启用日志记录后，AWS IoT FleetWise 会将日志以日志流的形式传送到 CloudWatch 日志组。

您可以在 CloudWatch 控制台中查看从 AWS IoT FleetWise 发送的日志数据。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](#)。

### 将日志发布到 CloudWatch 所需的权限

为 CloudWatch 日志组配置日志记录需要本节中所述的权限设置。有关管理权限的更多信息，请参阅《IAM 用户指南》中的[对 AWS 资源的访问权限管理](#)。

使用这些权限，您可以更改日志记录配置、配置 CloudWatch 的日志传输，以及检索有关日志组的信息。

```
{
  "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 "Resource" 设置为 "\*" 的策略中进行指示。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。

## 在 AWS IoT FleetWise ( 控制台 ) 中配置日志记录

本节介绍如何使用 AWS IoT FleetWise 控制台配置日志记录。

使用 AWS IoT FleetWise 控制台配置日志记录

1. 打开 [AWS IoT FleetWise 控制台](#)。
2. 在左侧窗格中，选择 Settings (设置)。
3. 在设置页面的日志记录部分，选择编辑。
4. 在 CloudWatch 日志记录部分，输入日志组。
5. 选择提交可保存更改。

启用日志记录后，可以在 [CloudWatch 控制台](#) 中查看日志数据。

## 在 AWS IoT FleetWise (CLI) 中配置默认日志记录

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 执行此流程，即使用 AWS API 中与此处显示的 CLI 命令对应的方法。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 IoT FleetWise 配置日志记录

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`

是将数据发送到 CloudWatch Logs 的日志类型。要禁用日志记录，请将该值更改为 OFF。

`logGroupName`

是该操作将数据发送到的 CloudWatch Logs 组。在启用 AWS IoT FleetWise 日志记录功能之前，请务必创建日志组名称。

启用日志记录后，请参阅[使用 AWS CLI 搜索日志条目](#)。

## 使用 AWS CloudTrail 记录 AWS IoT FleetWise API 调用

AWS IoT FleetWise 与 AWS CloudTrail 集成，后者是一项提供用户、角色或 AWS 服务在 AWS IoT FleetWise 中所执行操作的记录的服务。CloudTrail 将对 AWS IoT FleetWise 的所有 API 调用作为事件捕获。捕获的调用包含来自 AWS IoT FleetWise 控制台的调用，以及代码对 AWS IoT FleetWise 操作的调用。如果您创建跟踪，则可以将 CloudTrail 事件持续传送到 Amazon S3 存储桶（包括 AWS IoT FleetWise 事件）。如果您不配置跟踪，则仍可在 CloudTrail 控制台中的 Event history（事件历史记录）中查看最新事件。使用 CloudTrail 收集的信息，您可以确定向 AWS IoT FleetWise 发出了什么请求、发出请求的 IP 地址、请求方、请求时间以及其他详细信息。

要了解有关 CloudTrail 的更多信息，请参阅《[AWS CloudTrail 用户指南](#)》。



## AWSCloudTrail 中的 IoT FleetWise 信息

在您创建 AWS 账户时，将在该账户上启用 CloudTrail。当 AWS IoT FleetWise 中发生活动时，该活动将记录在 CloudTrail 事件中，并与其他 AWS 服务事件一同保存在事件历史记录中。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 CloudTrail 事件历史记录查看事件](#)。

要持续记录 AWS 账户中的事件（包括 AWS IoT FleetWise 事件），请创建跟踪。通过跟踪，CloudTrail 可将日志文件传送至 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亚马逊云科技区域。此跟踪在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Amazon S3 存储桶。此外，您可以配置其他 AWS 服务，进一步分析在 CloudTrail 日志中收集的事件数据并采取行动。有关更多信息，请参阅下列内容：

- [创建跟踪概览](#)
- [CloudTrail 支持的服务和集成](#)
- [为 CloudTrail 配置 Amazon SNS 通知](#)
- [从多个区域接收 CloudTrail 日志文件](#)
- [从多个账户接收 CloudTrail 日志文件](#)

所有 AWS IoT FleetWise 操作均由 CloudTrail 进行记录，并记录在[AWS IoT FleetWise API 参考](#)中。例如，对 CreateCampaign、AssociateVehicleFleet 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身份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，可用于将事件作为日志文件传送到您指定的 Amazon S3 桶。CloudTrail 日志文件包含一个或多个日志条目。一个事件表示来自任何源的一个请求，包括有关所请求的操作、操作的日期和时间、请求参数等方面的信息。CloudTrail 日志文件不是公用 API 调用的有序堆栈跟踪，因此它们不会按任何特定顺序显示。

下面的示例显示了一个 CloudTrail 日志条目，该条目说明了 *AssociateVehicleFleet* 操作。

```
{
  "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 IoT FleetWise 开发人员指南的文档历史记录

下表介绍了 AWS IoT FleetWise 的文档发布。

变更	说明	日期
<a href="#">视觉系统数据预览</a>	您可以使用来自 AWS IoT FleetWise 的视觉系统数据预览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
<a href="#">AWS KMS 客户管理密钥</a>	AWS IoT FleetWise 现在支持 AWS KMS 客户管理密钥。您可以使用 KMS 密钥对存储在 AWS Cloud 中的 AWS IoT 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。	2023 年 10 月 16 日
<a href="#">Amazon S3 中的对象存储</a>	AWS IoT FleetWise 现在支持使用 Amazon Simple Storage Service (Amazon S3) 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
<a href="#">正式发布</a>	这是 AWS IoT FleetWise 的公开版本。	2022 年 9 月 27 日
<a href="#">初始版本</a>	这是《AWS IoT FleetWise 开发人员指南》的预览版。	2021 年 11 月 30 日

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。