



開發人員指南

# AWS IoT FleetWise



# AWS IoT FleetWise: 開發人員指南

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

Amazon 的商標和商業外觀不得用於任何非 Amazon 的產品或服務，也不能以任何可能造成客戶混淆、任何貶低或使 Amazon 名譽受損的方式使用 Amazon 的商標和商業外觀。所有其他非 Amazon 擁有的商標均為其各自擁有者的財產，這些擁有者可能附屬於 Amazon，或與 Amazon 有合作關係，亦或受到 Amazon 贊助。

# Table of Contents

|   |    |
|---|----|
| 什麼是 AWS IoT FleetWise ? .....               | 1  |
| 優勢 .....                                    | 2  |
| 使用案例 .....                                  | 2  |
| 您是 AWS IoT 的新手 FleetWise嗎? .....            | 2  |
| 存取 AWS IoT FleetWise .....                  | 3  |
| AWSIoT 定價 FleetWise .....                   | 3  |
| AWSIoT 如何 FleetWise 運作 .....                | 3  |
| 重要概念 .....                                  | 3  |
| AWSIoT 的特色 FleetWise .....                  | 6  |
| 相關服務 .....                                  | 7  |
| 設定 AWS IoT FleetWise .....                  | 8  |
| 設定您的 AWS 帳戶 .....                           | 8  |
| 註冊一個 AWS 帳戶 .....                           | 8  |
| 建立具有管理權限的使用者 .....                          | 8  |
| 在主控台中開始使用 .....                             | 10 |
| 進行設定 .....                                  | 10 |
| 進行設定 (主控台) .....                            | 10 |
| 配置設置 ( AWS CLI ) .....                      | 11 |
| 開始使用 .....                                  | 13 |
| 要求 .....                                    | 13 |
| 使用邊緣代理程式軟體示範 .....                          | 13 |
| 開始使用 (主控台) .....                            | 14 |
| 必要條件 .....                                  | 15 |
| 步驟 1 : 設定 AWS IoT 的邊緣代理程式軟體 FleetWise ..... | 15 |
| 步驟 2 : 建立車輛模型 .....                         | 17 |
| 步驟 3 : 創建解碼器清單 .....                        | 18 |
| 步驟 4 : 配置解碼器清單 .....                        | 19 |
| 步驟 5 : 建立車輛 .....                           | 19 |
| 步驟 6 : 建立廣告活動 .....                         | 21 |
| 步驟 7 : 清除 .....                             | 22 |
| 後續步驟 .....                                  | 22 |
| 將資料導入雲端 .....                               | 23 |
| 車輛模型 .....                                  | 26 |
| 信號目錄 .....                                  | 28 |

|                             |    |
|-----------------------------|----|
| 設定訊號 .....                  | 30 |
| 建立信號目錄 (AWS CLI) .....      | 35 |
| 匯入信號目錄 .....                | 40 |
| 更新信號目錄 (AWS CLI) .....      | 49 |
| 刪除信號目錄 (AWS CLI) .....      | 51 |
| 取得訊號目錄資訊 (AWS CLI) .....    | 52 |
| 車輛型號 .....                  | 52 |
| 建立車輛模型 .....                | 53 |
| 更新車輛型號 ( AWS CLI ) .....    | 59 |
| 刪除車輛型號 .....                | 60 |
| 取得車輛型號資訊 (AWS CLI) .....    | 61 |
| 解碼器清單 .....                 | 61 |
| 設定網路介面和解碼器訊號 .....          | 63 |
| 創建解碼器清單 .....               | 66 |
| 更新解碼器清單 ( ) AWS CLI .....   | 73 |
| 刪除解碼器清單 .....               | 73 |
| 獲取解碼器清單信息 ( ) AWS CLI ..... | 74 |
| 車輛 .....                    | 76 |
| 提供車輛 .....                  | 77 |
| 驗證車輛 .....                  | 77 |
| 授權車輛 .....                  | 79 |
| 預留主題 .....                  | 80 |
| 建立車輛 .....                  | 82 |
| 創建車輛 ( 控制台 ) .....          | 82 |
| 建立車輛 (AWS CLI) .....        | 84 |
| 創建多個車輛 ( AWS CLI ) .....    | 86 |
| 更新車輛 ( AWS CLI ) .....      | 87 |
| 更新多輛車 ( AWS CLI ) .....     | 88 |
| 刪除車輛 .....                  | 89 |
| 刪除車輛 ( 主機 ) .....           | 89 |
| 刪除車輛 (AWS CLI) .....        | 89 |
| 取得車輛資訊 (AWS CLI) .....      | 90 |
| 機群 .....                    | 91 |
| 建立叢集 (AWS CLI) .....        | 92 |
| 將車輛與車隊建立關聯 (AWS CLI) .....  | 92 |
| 取消車輛與車隊的關聯 (AWS CLI) .....  | 93 |

|  |     |
|--|-----|
| 更新艦隊 (AWS CLI) .....   | 93  |
| 刪除機群群群 (AWS CLI) .....   | 94  |
| 取得車隊資訊 (AWS CLI) .....   | 94  |
| 行銷活動 .....   | 96  |
| 建立行銷活動 .....   | 100 |
| 建立行銷活動 (主控台) .....   | 101 |
| 建立廣告活動 (AWS CLI) .....   | 107 |
| 促銷活動的邏輯運算 .....  | 110 |
| 更新行銷活動 (AWS CLI) .....   | 111 |
| 刪除廣告活動 .....   | 112 |
| 刪除廣告活動 (主機) .....  | 112 |
| 刪除廣告活動 (AWS CLI) .....   | 112 |
| 取得行銷活動資訊 (AWS CLI) .....   | 112 |
| 處理和視覺化車輛資料 .....   | 114 |
| 在時間流中處理車輛數據 .....  | 114 |
| 可視化儲存在時間流中的車輛資料 .....  | 115 |
| 在 S3 中處理車輛資料 .....   | 115 |
| S3 物件格式 .....  | 116 |
| 分析 S3 中存放的車輛資料 .....   | 116 |
| AWS CLI和AWS開發套件 .....  | 119 |
| 故障診斷 .....   | 120 |
| 解碼器清單問題 .....  | 120 |
| AWSIoT FleetWise 軟體問題的邊緣代理程式 .....   | 123 |
| 問題：Edge 代理程式軟體無法啟動。 .....  | 123 |
| 問題:[錯誤] [IoT:FleetWiseEngine: 連接]: [無法初始化持久性庫] .....   | 124 |
| 問題：邊緣代理程式軟體不會收集內建診斷 (OBD) II PID 和診斷故障碼 (DTC)。 .....   | 125 |
| 問題：AWSIoT FleetWise 軟體的 Edge 代理程式無法從網路收集資料，或無法套用資料檢查規則。 .....  | 125 |
| 問題：[錯誤] [:AwsIotConnectivityModule: 連接]：[連接失敗，錯誤] 或 [警告] [:: 發送]AwsIotChannel：[沒有活動的 MQTT 連接。] ..... | 126 |
| 安全 .....   | 127 |
| 資料保護 .....   | 127 |
| 靜態加密 .....   | 128 |
| 傳輸中加密 .....  | 129 |
| 資料加密 .....   | 129 |
| 控制存取 .....   | 136 |

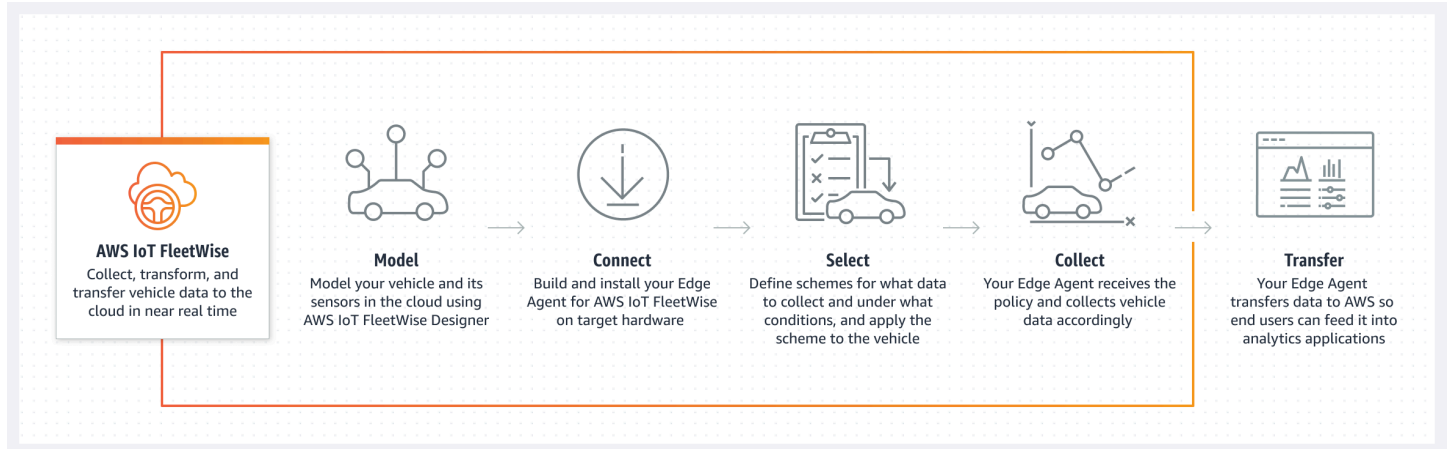
|  |        |
|--|--------|
| 授 AWS IoT FleetWise 予對 Amazon S3 目的地的存取權 .....         | 136    |
| 授 AWS IoT FleetWise 予對 Amazon Timestream 目的地的存取權 ..... | 139    |
| 身分和存取權管理 .....   | 142    |
| 物件 .....   | 142    |
| 使用身分驗證 .....   | 143    |
| 使用政策管理存取權 .....  | 145    |
| AWS IoT 如何與 IAM FleetWise 搭配運作 .....                   | 147    |
| 身分型政策範例 .....  | 155    |
| 故障診斷 .....   | 158    |
| 合規驗證 .....   | 159    |
| 恢復能力 .....   | 160    |
| 基礎架構安全 .....   | 161    |
| FleetWise 透過介面 VPC AWS 端點連線至 IoT .....                 | 161    |
| 組態與漏洞分析 .....  | 164    |
| 安全最佳實務 .....   | 164    |
| 盡可能授予最低的許可 .....                                       | 165    |
| 請勿記錄敏感資訊 .....   | 165    |
| 用 AWS CloudTrail 於查看 API 呼叫歷史記錄 .....                  | 165    |
| 讓裝置的時鐘保持同步 .....                                       | 165    |
| 監控 .....   | 166    |
| 使用監控 CloudWatch .....                                  | 166    |
| 使用 CloudWatch 記錄監控 .....                               | 170    |
| 在 CloudWatch 主控台中檢視 AWS IoT FleetWise 記錄 .....         | 170    |
| 設定 記錄 .....  | 175    |
| CloudTrail 日誌 .....                                    | 177    |
| AWS物聯網FleetWise中的資訊CloudTrail .....                    | 177    |
| 了解AWS物聯網FleetWise記錄檔項目 .....                           | 178    |
| 文件歷史記錄 .....   | 180    |
| .....  | clxxxi |

# 什麼是 AWS IoT FleetWise ？

AWS IoT FleetWise 是一種託管服務，您可以使用它來收集車輛數據並將其組織到雲中。您可以使用收集到的數據來提高車輛質量，性能和自主性。透過 AWS IoT FleetWise，您可以從使用不同通訊協定和資料格式的車輛收集和整理資料。AWS IoT FleetWise 有助於將低級消息轉換為人類可讀的值，並將雲中的數據格式標準化以進行數據分析。您也可以定義資料收集行銷活動，以控制要收集哪些車輛資料，以及何時將該資料傳輸到雲端。

當車輛資料存放在雲端時，您可以將其用於分析車隊健康狀況的應用程式。這些資料可協助您識別潛在的維護問題、讓車載資訊娛樂系統變得更智慧，並透過分析和機器學習 (ML) 改善自動駕駛和駕駛輔助系統等先進技術。

下圖顯示了 AWS IoT 的基本架構 FleetWise。



## 主題

- [優勢](#)
- [使用案例](#)
- [您是 AWS IoT 的新手 FleetWise 嗎？](#)
- [存取 AWS IoT FleetWise](#)
- [AWS IoT 定價 FleetWise](#)
- [AWS IoT 如何 FleetWise 運作](#)
- [相關服務](#)

## 優勢

AWS IoT 的主要優點 FleetWise 是：

### 更聰明地收集車輛資料

透過智慧型資料收集功能，僅將您需要的資料傳送至雲端進行分析，藉此改善資料相關性。

### 輕鬆分析整個叢集的標準化資料

無需開發自訂資料收集或記錄系統，即可分析來自車隊的標準化資料。

### 雲端中的自動資料同步

取得從標準感測器 (遙測資料) 和視覺系統 (來自攝影機、雷達和蓋板的資料) 收集到的資料的統一檢視，並將其自動保持在雲端中同步。AWS IoT FleetWise 會在雲端中自動同步結構化和非結構化視覺系統資料、中繼資料和標準感測器資料。這簡化了組合事件的完整視圖並獲得見解的流程。

#### Note

視覺系統資料正在預覽版中，可能會有所變更。

## 使用案例

您可以使用 AWS IoT 的案例 FleetWise 包括下列各項：

### 訓練 AI/ML 模型

透過從生產車輛收集資料，持續改善用於自動駕駛和先進駕駛輔助系統的機器學習模型。

### 提升數位客戶體驗

使用來自資訊娛樂系統的資料，讓車載視聽內容和應用程式內見解更具相關性。

### 維持車隊健康

利用車隊資料中的洞察來監控 EV 電池的健康狀況和充電量、管理維護時間表、分析油耗等。

## 您是 AWS IoT 的新手 FleetWise 嗎？

如果您是 AWS IoT 的新手 FleetWise，建議您先閱讀下列章節：



- [AWSIoT 如何 FleetWise 運作](#)
- [設定 AWS IoT FleetWise](#)
- [邊緣代理軟體示範](#)
- [將資料導入雲端](#)

## 存取 AWS IoT FleetWise

您可以使用 AWS IoT FleetWise 主控台或 API 來存取 AWS IoT FleetWise。

## AWSIoT 定價 FleetWise

車輛會透過 MQTT 訊息將資料傳送至雲端。您在每個月底為在 AWS IoT 中建立的車輛付費 FleetWise。您也需要為從車輛收集的訊息付費。如需有關定價的最新資訊，請參閱 [AWSIoT FleetWise 定價](#) 頁面。若要深入了解 MQTT 訊息通訊協定，請參閱開發人員指南中的 [MQTT](#)。AWS IoT Core

## AWSIoT 如何 FleetWise 運作

以下各節提供 AWS IoT FleetWise 服務元件及其互動方式的概觀。

閱讀本簡介後，請參閱 [設定 AWS IoT FleetWise](#) 章節以了解如何設定 AWS IoT FleetWise。

主題

- [重要概念](#)
- [AWSIoT 的特色 FleetWise](#)

### 重要概念

AWSIoT FleetWise 提供車輛建模架構，讓您在雲端建立車輛及其感測器和致動器的模型。為了確保車輛與雲端之間的安全通訊，AWSIoT FleetWise 還提供了參考實作，協助您開發 Edge Agent 軟體，供您安裝在車輛中。您可以在雲端中定義資料收集方案，並將其部署到您的車輛。車輛中執行的 Edge Agent 軟體會使用資料收集配置來控制要收集哪些資料，以及何時將其傳輸到雲端。

以下是 AWS IoT 的核心概念 FleetWise。

## Signal

信號是您定義用來包含車輛資料及其中繼資料的基本結構。信號可以是屬性，分支，傳感器或致動器。例如，您可以創建一個傳感器來接收車載溫度值，並存儲其元數據，包括傳感器名稱，數據類型和單位。如需詳細資訊，請參閱[建立和管理信號目錄](#)。

### 屬性

屬性代表一般不會變更的靜態資訊，例如製造商和製造日期。

### 分支

分支代表嵌套結構中的信號。分支演示信號層次結構。例如，分Vehicle支有一個子分支Powertrain。該Powertrain分支有一個子分支combustionEngine。若要尋找分combustionEngine支，請使用Vehicle.Powertrain.combustionEngine運算式。

### 感測器

傳感器數據報告車輛的當前狀態並隨著時間的推移而變化，例如車輛的狀態發生變化，例如液位，溫度，振動或電壓。

### 傳動器

致動器資料會報告車輛裝置的狀態，例如馬達、加熱器和門鎖。更改車輛設備的狀態可以更新致動器數據。例如，您可以定義致動器來表示加熱器。當您打開或關閉加熱器，致動器接收新的數據。

### 自定義結構

自訂結構 (也稱為結構) 代表複雜或高階資料結構。它有助於邏輯綁定或源自同一來源的數據分組。當數據被讀取或寫入一個原子操作，如表示一個複雜的數據類型或高階形狀的結構使用。

結構類型的信號是使用對結構數據類型的引用而不是原始數據類型在信號目錄中定義的。結構可用於所有類型的訊號，包括感測器、屬性、致動器和視覺系統資料類型。如果發送或接收結構類型的信號，AWS IoT FleetWise 預期所有包含的項目都具有有效值，因此所有項目都是強制性的。例如，如果一個結構包含的項目車輛相機圖像高度，車輛相機.寬度和車輛.相機。

#### Note

視覺系統資料正在預覽版中，可能會有所變更。

### 自訂屬性

自訂屬性代表複雜資料結構的成員。屬性的資料類型可以是原始或其他結構。

當使用 `struct` 和 `custom` 屬性表示高階形狀時，預期的高階形狀始終被定義並顯示為樹狀結構。自定義屬性用於定義所有葉節點，而該結構用於定義所有非葉節點。

## 信號目錄

信號目錄包含信號的集合。信號目錄中的信號可用於對使用不同協議和數據格式的車輛進行建模。例如，有兩輛汽車由不同的汽車製造商製造：一輛使用控制區域網絡 (CAN 總線) 協議；另一個使用板載診斷 (OBD) 協議。您可以在信號目錄中定義一個傳感器以接收車內溫度值。該傳感器可用於表示兩輛汽車中的熱電偶。如需詳細資訊，請參閱[建立和管理信號目錄](#)。

## 車輛型號 (模型清單)

車輛模型是聲明性結構，您可以使用它來標準化車輛的格式，並定義車輛中信號之間的關係。車輛模型會在多輛相同類型的車輛上執行一致的資訊。您可以加入訊號以建立車輛模型。如需詳細資訊，請參閱[創建和管理車輛模型](#)。

## 解碼器清單

解碼器清單包含車輛模型中每個信號的解碼信息。車輛中的傳感器和致動器傳輸低級信息 (二進制數據)。借助解碼器清單，AWS IoT 能夠將二進制數據轉換為人類可讀的 FleetWise 值。每個解碼器清單都與車輛模型相關聯。如需詳細資訊，請參閱[創建和管理解碼器清單](#)。

## 網路介面

包含車載網絡使用的協議的信息。AWS IoT FleetWise 支持以下協議。

### 控制器區域網路 (CAN 匯流排)

一種協議，用於定義數據如何在電子控制單元 (ECU) 之間進行通信。ECU 可以是引擎控制單元、安全氣囊或音訊系統。

### 車載診斷系統 (OBD) II

進一步開發的通訊協定，用於定義 ECU 之間如何通訊自我診斷資料。它提供了一些標準的診斷故障代碼 (DTC)，有助於識別您的車輛出了什麼問題。

### 車輛中間件

車輛中間件定義為一種類型的網絡接口。車輛中介軟體的範例包括機器人作業系統 (ROS 2) 和透過 IP (SOME/IP) 的可擴充服務導向中介軟體。

#### Note

AWS IoT FleetWise 支援視覺系統資料的 ROS 2 中介軟體。

## 解碼器信號

提供特定訊號的詳細解碼資訊。車輛型號中指定的每個信號都必須與解碼器信號配對。如果解碼器清單包含 CAN 網絡接口，則必須包含 CAN 解碼器信號。如果解碼器清單包含 OBD 網絡接口，則必須包含 OBD 解碼器信號。

如果解碼器清單還包含車輛中間件接口，則解碼器清單必須包含消息解碼器信號。

## 車輛

您的物理車輛的虛擬表示，例如汽車或卡車。車輛是車輛模型的實例。使用相同車型製造的車輛會繼承相同的信號組。每輛車對應一個AWS IoT東西。

## 機群

車隊代表一組車輛。您必須先將個別車輛與車隊建立關聯，才能輕鬆管理車隊。

## Campaign

包含資料收集配置。您可以在雲端中定義宣傳活動，並將其部署到車輛或車隊。行銷活動會提供 Edge Agent 軟體指示，說明如何選取、收集資料並將資料傳輸到雲端。

## 資料收集方案

資料收集配置會提供 Edge Agent 軟體指示，說明如何收集資料。目前，AWS IoT FleetWise 支持基於條件的收集方案和基於時間的收集方案。

### 基於條件的收集計劃

使用邏輯運算式識別要收集的資料。Edge 代理程式軟體會在符合條件時收集資料。例如，如果運算式為 `$variable.myVehicle.InVehicleTemperature >35.0`，Edge Agent 軟體會收集大於 35.0 的溫度值。

### 基於時間的收集計劃

指定以毫秒為單位的時間範圍，以定義收集資料的頻率。例如，如果時間週期為 10,000 毫秒，Edge Agent 軟體會每 10 秒收集一次資料。

## AWS IoT 的特色 FleetWise

以下是 AWS IoT 的主要功能 FleetWise。

### 車輛建模

為您的車輛建立虛擬表示，並應用通用格式來組織車輛信號。AWS IoT FleetWise 支援 [車輛訊號規格 \(VSS\)](#)，可用來標準化車輛訊號。

## 基於計劃的數據收集

定義僅將高價值車輛資料傳輸到雲端的方案。您可以定義以條件為基礎的配置來控制要收集的資料，例如大於 40 度的車內溫度值的資料。您也可以定義以時間為基礎的配置，以控制收集資料的頻率。

### AWS IoT FleetWise 軟體的邊緣代理程式

在車輛中執行的 Edge Agent 軟體可促進車輛與雲端之間的通訊。車輛連線至雲端時，Edge Agent 軟體會持續接收資料收集配置，並據此收集資料。

## 相關服務

AWS IoT 與下列 AWS 服務 FleetWise 整合，以提升雲端解決方案的可用性和可擴充性。

- AWS IoT Core— 註冊和控制將車輛數據上傳到 AWS IoT 的 AWS IoT 設備 FleetWise。如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 [什麼是 AWS IoT](#)。
- Amazon Timestream — 使用時間序列資料庫來存放和分析您的車輛資料。如需詳細資訊，請參閱 [Amazon 時間流開發人員指南中的 Amazon Timestream 是什麼](#)。
- Amazon S3 — 使用物件儲存服務來存放和管理您的車輛資料。如需詳細資訊，請參閱 [Amazon 簡單儲存服務使用者指南中的 Amazon S3 是什麼](#)。

# 設定 AWS IoT FleetWise

第一次使 FleetWise 用 AWS IoT 之前，請先完成以下各節中的步驟。

主題

- [設定您的 AWS 帳戶](#)
- [在主控台中開始使用](#)
- [進行設定](#)

## 設定您的 AWS 帳戶

完成下列工作以註冊 AWS 並建立系統管理使用者。

### 註冊一個 AWS 帳戶

如果您沒有 AWS 帳戶，請完成以下步驟來建立一個。

若要註冊成為 AWS 帳戶

1. 開啟 <https://portal.aws.amazon.com/billing/signup>。
2. 請遵循線上指示進行。

部分註冊程序需接收來電，並在電話鍵盤輸入驗證碼。

當您註冊一個時 AWS 帳戶，將創建 AWS 帳戶根使用者一個。根使用者有權存取該帳戶中的所有 AWS 服務 和資源。安全性最佳做法是將管理存取權指派給使用者，並僅使用 [root 使用者來執行需要 root 使用者存取權](#) 的工作。

AWS 註冊過程完成後，會向您發送確認電子郵件。您可以隨時登錄 <https://aws.amazon.com/> 並選擇我的帳戶，以檢視您目前的帳戶活動並管理帳戶。

### 建立具有管理權限的使用者

註冊後，請保護您的 AWS 帳戶 AWS 帳戶根使用者 AWS IAM Identity Center、啟用和建立系統管理使用者，這樣您就不會將 root 使用者用於日常工作。

## 保護您的 AWS 帳戶根使用者

1. 選擇 Root 使用者並輸入您的 AWS 帳戶 電子郵件地址，以帳戶擁有者身分登入。[AWS Management Console](#)在下一頁中，輸入您的密碼。

如需使用根使用者登入的說明，請參閱 AWS 登入 使用者指南中的[以根使用者身分登入](#)。

2. 若要在您的根使用者帳戶上啟用多重要素驗證 (MFA)。

如需指示，請參閱《IAM 使用者指南》中的[為 AWS 帳戶 根使用者啟用虛擬 MFA 裝置 \(主控台\)](#)。

## 建立具有管理權限的使用者

1. 啟用 IAM Identity Center。

如需指示，請參閱 AWS IAM Identity Center 使用者指南中的[啟用 AWS IAM Identity Center](#)。

2. 在 IAM 身分中心中，將管理存取權授予使用者。

[若要取得有關使用 IAM Identity Center 目錄 做為身分識別來源的自學課程，請參閱《使用指南》IAM Identity Center 目錄中的「以預設值設定使用AWS IAM Identity Center 者存取」。](#)

## 以具有管理權限的使用者身分登入

- 若要使用您的 IAM Identity Center 使用者簽署，請使用建立 IAM Identity Center 使用者時傳送至您電子郵件地址的簽署 URL。

如需使用 IAM 身分中心使用者[登入的說明](#)，請參閱[使用AWS 登入 者指南中的登入 AWS 存取入口網站](#)。

## 指派存取權給其他使用者

1. 在 IAM 身分中心中，建立遵循套用最低權限許可的最佳做法的權限集。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[建立權限集](#)」。

2. 將使用者指派給群組，然後將單一登入存取權指派給群組。

如需指示，請參閱《AWS IAM Identity Center 使用指南》中的「[新增群組](#)」。

### Note

您可以將服務連結角色與 AWS IoT FleetWise 搭配使用。服務連結角色由 AWS IoT 預先定義，FleetWise 並包含 AWS IoT 傳送指標至 Amazon CloudWatch 所 FleetWise 需的許可。如需詳細資訊，請參閱 [針對 AWS IoT 使用服務連結角色 FleetWise](#)。

## 在主控台中開始使用

如果您尚未登入，請登入 AWS 帳戶，然後開啟 [AWS IoT 主 FleetWise 控制台](#)。若要開始使用 AWS IoT FleetWise，請建立車輛模型。車輛模型將車輛的格式標準化。

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在 [開始使用] 中 AWS IoT FleetWise，選擇 [開始使用]。

如需建立車輛模型的更多資訊，請參閱 [創建車輛型號 \(控制台\)](#)。

## 進行設定

您可以使用 AWS IoT FleetWise 主控台或 API 來設定 Amazon CloudWatch 日誌指標、Amazon CloudWatch 日誌的設定，以及使用 AWS 受管金鑰。

透過 CloudWatch 指標，您可以監控 AWS IoT FleetWise 和其他 AWS 資源。您可以使用 CloudWatch 指標來收集和追蹤指標，例如判斷是否有超出服務限制。如需 CloudWatch 測量結果的詳細資訊，請參閱 [FleetWise 用 Amazon 監控 AWS IoT CloudWatch](#)。

使用 CloudWatch Logs，AWS IoT FleetWise 會將記錄資料傳送到記 CloudWatch 錄群組，您可以在其中使用它來識別和緩解任何問題。如需有關 CloudWatch 記錄檔的詳細資訊，請參閱 [設定 AWS IoT FleetWise 記錄](#)。

透過資料加密，AWS IoT AWS 受管金鑰可 FleetWise 用來加密資料。您也可以選擇使用建立和管理金鑰 AWS KMS。如需加密的詳細資訊，請參閱 [資料加密](#)。

## 進行設定 (主控台)

如果您尚未登入，請登入 AWS 帳戶，然後開啟 [AWS IoT 主 FleetWise 控制台](#)。

1. 導覽至 [AWS IoT FleetWise 主控台](#)。



2. 在左窗格中，選擇 [設定]。
3. 在量度中，選擇啟用。AWS IoT FleetWise 會自動將 CloudWatch 受管理的原則附加至服務連結角色，並啟用 CloudWatch 指標。
4. 在記錄中，選擇編輯。
  - a. 在「記CloudWatch 錄」區段中，輸入「記錄」群組。
  - b. 若要儲存變更，請選擇「提交」。
5. 在「加密」區段中，選擇「編輯」。
  - a. 選擇您要使用的金鑰類型。如需詳細資訊，請參閱 [金鑰管理](#)。
    - i. 使用 AWS 金鑰 — AWS IoT FleetWise 擁有並管理金鑰。
    - ii. 選擇不同的 AWS Key Management Service 金鑰 — 您可 AWS KMS keys 以管理您帳戶中的金鑰。
  - b. 若要儲存變更，請選擇「提交」。

## 配置設置 ( AWS CLI )

在中 AWS CLI，註冊帳戶以進行設定。

1. 若要進行設定，請執行下列命令。

```
aws iotfleetwise register-account
```

2. 若要驗證您的設定，請執行下列命令以擷取註冊狀態。

### Note

服務連結角色僅用於將 AWS IoT FleetWise 指標發佈至 CloudWatch。如需詳細資訊，請參閱 [針對 AWS IoT 使用服務連結角色 FleetWise](#)。

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

### Example response

```
{
```

```
"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",
}
}
```

註冊狀態可以是下列其中一種：

- REGISTRATION\_SUCCESS— 資 AWS 源已成功註冊。
- REGISTRATION\_PENDING— AWS IoT FleetWise 正在處理註冊請求。此過程大約需要五分鐘才能完成。
- REGISTRATION\_FAILURE— AWS IoT FleetWise 無法註冊資 AWS 源。請稍後再試。

# 開始使用 AWS IoT FleetWise

透過 AWS IoT FleetWise，您可以收集、轉換和傳輸車輛資料。使用本節中的教學課程開始使用 AWS IoT FleetWise。

若要深入了解 AWS IoT，請參閱下列主題 FleetWise：

- [將資料導入雲端](#)
- [車輛模型](#)
- [建立、佈建和管理車輛](#)
- [建立和管理車隊](#)
- [透過行銷活動收集和傳輸資料](#)

## 要求

您必須擁有一個 AWS 帳戶 才能開始使用 AWS IoT FleetWise。如果您沒有帳戶，請參閱 [設定 AWS IoT FleetWise](#)。

使用可使用 AWS IoT FleetWise 的區域。如需詳細資訊，請參閱 [AWS IoT FleetWise 端點和配額](#)。您可以使用中的「區域」選取器切換 AWS Management Console 至下列其中一個區域。

## 邊緣代理軟體示範

您可以使用探索 Edge 代理程式快速入門示範來探索 AWS IoT，FleetWise 並了解如何開發適用於 AWS IoT 的邊緣代理程式軟體 FleetWise。本演示使用 AWS CloudFormation 模板。它會引導您檢閱邊緣代理程式參考實作、開發邊緣代理程式，然後在 Amazon EC2 Graviton 上部署邊緣代理程式軟體，以及產生車輛範例資料。此示範也提供指令碼，您可以用來建立訊號目錄、車輛模型、解碼器資訊清單、車輛、車隊和行銷活動 — 全都在雲端中。如需快速入門示範的詳細資訊，請執行下列動作以下載 Edge Agent 軟體開發人員指南。

下載快速入門示範

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在服務首頁的 [開始使用 AWS IoT] FleetWise 區段中，選擇 [探索 Edge 代理程式]。

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](#)

## 教學課程：開始使用 AWS IoT FleetWise (主控台)

使用 AWS IoT FleetWise 以近乎即時的方式，從自動化車輛收集、轉換和傳輸獨特的資料格式至雲端。您可以訪問整個機隊的見解。這可以幫助您有效地檢測和緩解車輛健康問題，傳輸高價值數據信號以及遠程診斷問題，同時降低成本。

本教學課程說明如何開始使用 AWS IoT FleetWise。您將學習如何建立車輛模型（模型清單）、解碼器清單、車輛和戰役。

如需 AWS IoT 關鍵元件和概念的詳細資訊 FleetWise，請參閱[AWS IoT 如何 FleetWise 運作](#)。

預計時間：約 45 分鐘

### ⚠ Important

您需要支付此示範建立和使用的 AWS IoT FleetWise 資源費用。如需詳細資訊，請參閱[AWS IoT FleetWise 定價頁面](#) FleetWise 中的 IoT。

### 主題

- [必要條件](#)
- [步驟 1：設定 AWS IoT 的邊緣代理程式軟體 FleetWise](#)
- [步驟 2：建立車輛模型](#)
- [步驟 3：創建解碼器清單](#)

- [步驟 4：配置解碼器清單](#)
- [步驟 5：建立車輛](#)
- [步驟 6：建立廣告活動](#)
- [步驟 7：清除](#)
- [後續步驟](#)

## 必要條件

若要完成此入門教學課程，您首先需要下列項目：

- 一個 AWS 帳戶。如果您沒有 AWS 帳戶，請參閱 [AWS Account Management 參考指南](#) AWS 帳戶中的 [建立](#)。
- 存取支 AWS 區域 援 AWS IoT 的 FleetWise。目前，美國東部 (維吉尼亞北部) 和歐洲 (法蘭克福) 支援 AWS IoT FleetWise。
- Amazon Timestream 資源：
  - Amazon Timestream 數據庫。如需詳細資訊，請參閱 [Amazon Timestream 開發人員指南](#) 中的 [建立資料庫](#)。
  - Amazon Timestream 中創建的 Amazon Timestream 表將保存您的數據。如需詳細資訊，請參閱 [Amazon Timestream 開發人員指南](#) 中的 [建立表格](#)。

## 步驟 1：設定 AWS IoT 的邊緣代理程式軟體 FleetWise

### Note

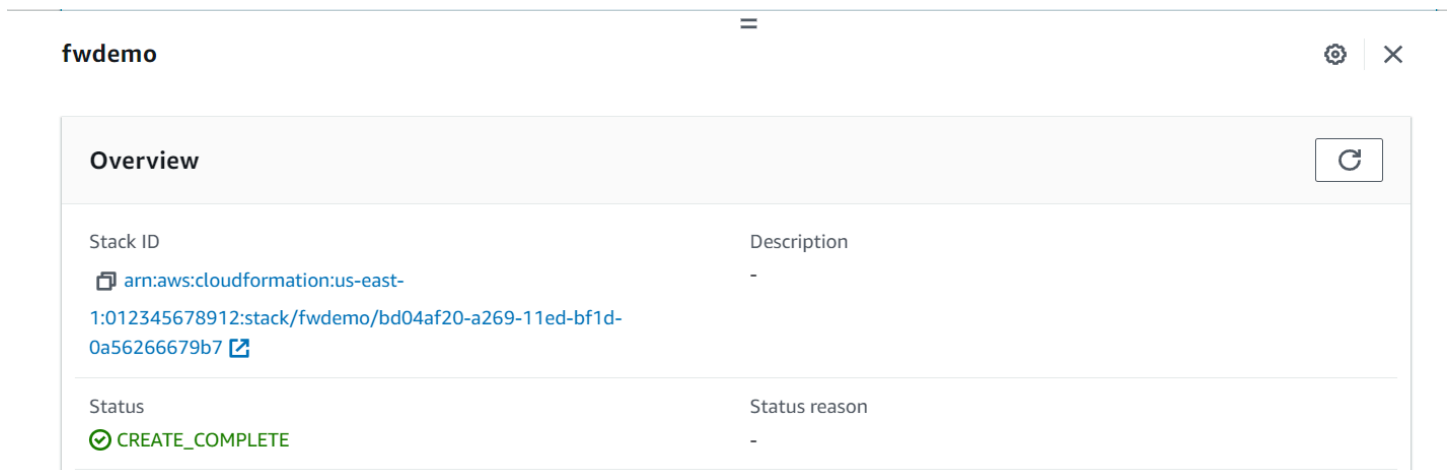
此步驟中的 CloudFormation 堆疊會使用遙測資料。您也可以使用視覺系統資料建立 CloudFormation 堆疊。如需詳細資訊，請參閱 [視覺系統資料開發人員指南](#)。視覺系統資料正在預覽版中，可能會有所變更。

適用 FleetWise 於 AWS IoT 的 Edge Agent 軟體可促進車輛與雲端之間的通訊。它會接收來自資料收集方案的指示，說明如何從雲端連線車輛收集資料。

若要設定 Edge Agent 軟體，請在一般資訊中執行下列動作：

1. 開啟 [啟動 CloudFormation 範本](#)。

- 在 [快速建立堆疊] 頁面上，對於 [堆疊名稱]，輸入 AWS IoT FleetWise 資源堆疊的名稱。堆疊是一個易記的名稱，在此 AWS CloudFormation 範本建立的資源名稱上顯示為前綴。
- 在「參數」下，為與堆疊相關的參數輸入自訂值。
  - 車隊大小-您可以通過更新車隊大小參數來增加車隊中的車輛數量。
  - IoT CoreRegion-您可以透過更新 IoT CoreRegion 參數來指定建立物 AWS IoT 件的區域。您必須使用您用來建立 AWS IoT FleetWise 車輛的相同區域。如需詳細資訊 AWS 區域，請參閱 [區域和區域-Amazon 彈性運算雲端](#)。
- 在 [功能] 區段中，選取方塊以確認 AWS CloudFormation 建立 IAM 資源。
- 選擇「建立堆疊」，然後等待大約 15 分鐘，讓堆疊的狀態顯示「建立 \_ 完成」。
- 若要確認堆疊已建立，請選擇 [堆疊資訊] 索引標籤，重新整理檢視，然後尋找 CREATE\_COMPLETE。



The screenshot shows the AWS CloudFormation console interface. At the top, the stack name 'fwdemo' is displayed. Below it, the 'Overview' tab is active, showing a table with the following information:

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

| Status   | Status reason |
|--|---------------|
| <span style="color: green;">✔</span> CREATE_COMPLETE | -             |

### Important

您需要支付此示範建立和使用的 AWS IoT FleetWise 資源費用。如需詳細資訊，請參閱 [AWS IoT FleetWise 定價頁面](#) FleetWise 中的 IoT。

## 步驟 2：建立車輛模型

### Important

您無法在 IoT FleetWise 主控台中建立具有視覺系統資料訊號的 AWS 車輛模型。相反地，請使用 AWS CLI。

您可以使用車輛模型來標準化車輛的格式，並協助定義所建立車輛中訊號之間的關係。建立車輛模型時，也會建立信號目錄。信號目錄是標準化信號的集合，可以重複使用以創建車輛模型。信號是您定義用來包含車輛資料及其中繼資料的基本結構。目前，AWS IoT FleetWise 服務每 AWS 區域 個帳戶僅支援一個訊號目錄。這有助於驗證從車隊處理的數據是否一致。

### 建立車輛模型

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 在「車輛型號」頁面上，選擇「建立車輛型號」。
4. 在「一般資訊」區段中，輸入車輛型號的名稱，例如「車輛 1」，以及選擇性描述。然後選擇下一步。
5. 從信號目錄中選擇一個或多個信號。您可以在搜尋目錄中依名稱篩選訊號，或從清單中選擇訊號。例如，您可以選擇輪胎壓力和煞車壓力的訊號，以便收集與這些訊號相關的資料。選擇下一步。
6. 選擇您的 .dbc 文件並從本地設備上傳它們。選擇下一步。

### Note

在本教學課程中，您可以下載[範例 .dbc 檔案](#)以進行此步驟上傳。

7. 為您的車輛型號新增屬性，然後選擇「下一步」。
  - a. 名稱-輸入運輸工具屬性的名稱，例如製造商名稱或製造日期。
  - b. 資料類型-在資料類型功能表上，選擇資料類型。
  - c. 單位-(選擇性) 輸入單位值，例如公里或攝氏度。
  - d. 路徑-(選擇性) 輸入信號路徑的名稱，如Vehicle.Engine.Light.點(.)表示該信號為子信號。
  - e. 預設值-(選擇性) 輸入預設值。

- f. 說明-(選擇性) 輸入屬性的說明。
8. 檢閱您的組態。當您準備好時，請選擇 **Create (建立)**。系統會顯示通知，說明您的車輛型號已成功建立。

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

## 步驟 3：創建解碼器清單

解碼器清單與您建立的車輛模型相關聯。它們包含的信息可幫助 AWS IoT FleetWise 解碼並將車輛數據從二進制格式轉換為可分析的人類可讀值。網絡接口和解碼器信號是幫助配置解碼器清單的組件。網絡接口包含有關車輛網絡使用的 CAN 或 OBD 協議的信息。解碼器信號為特定信號提供解碼信息。

若要建立解碼器資訊清單

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 在「車輛型號」部分中，選擇要用於創建解碼器清單的車輛型號。
4. 選擇創建解碼器清單。



## 步驟 4：配置解碼器清單

若要設定解碼器資訊清單

### Important

您無法使用 AWS IoT FleetWise 主控台在解碼器資訊清單中設定視覺系統資料訊號。相反地，請使用 AWS CLI。如需詳細資訊，請參閱 [創建解碼器清單 \( \) AWS CLI](#)。

1. 為了幫助您識別解碼器清單，請為其輸入名稱和可選描述。然後選擇下一步。
2. 若要新增一或多個網路介面卡，請選擇 CAN\_ 介面或 OBD\_ 介面類型。
  - 內建診斷 (OBD) 介面- 如果您想要定義自我診斷資料如何在電子控制單元 (ECU) 之間進行通訊的通訊協定，請選擇此介面類型。此通訊協定提供許多標準診斷故障代碼 (DTC)，可協助您排解車輛問題。
  - 控制器區域網路 (CAN 匯流排) 介面- 如果您想要定義 ECU 之間資料通訊方式的通訊協定，請選擇此介面類型。ECU 可以是引擎控制單元、安全氣囊或音訊系統。
3. 輸入網路介面名稱。
4. 若要將訊號新增至網路介面，請從清單中選擇一或多個訊號。
5. 為您在上一個步驟中新增加的訊號選擇解碼器訊號。若要提供解碼資訊，請上傳 .dbc 檔案。車輛型號中的每個信號都必須與解碼器信號配對，您可以從列表中選擇該信號。
6. 若要新增其他網路介面，請選擇 [新增網路介面]。當您完成新增網路介面時，請選擇 [下一步]。
7. 檢閱您的組態，然後選擇 [建立]。出現一條通知，說明您的解碼器清單已成功創建。

## 步驟 5：建立車輛

在 AWS IoT 中 FleetWise，車輛是您現實生活中的實體車輛的虛擬表現。所有從相同車輛型號建立的車輛都會繼承相同的訊號群組，而您建立的每輛車都會對應新建立的 IoT 物件。您必須將所有車輛與解碼器清單相關聯。

必要條件

1. 確認您已建立車輛型號和解碼器清單。另外，請確認車輛型號的狀態是否為「作用中」。
  - a. 若要確認車輛型號的狀態是否為作用中，請開啟 AWS IoT FleetWise 主控台。
  - b. 在導覽窗格中，選擇 [車輛型號]。

c. 在「摘要」區段的「狀態」下，查看車輛狀態。

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

## 建立車輛

1. 開啟 AWS FleetWise 主控台。
2. 在導覽窗格中，選擇 [車輛]。
3. 選擇「建立車輛」。
4. 要定義車輛屬性，請輸入車輛名稱，然後選擇模型清單 (車輛型號) 和解碼器清單。
5. (選擇性) 若要定義車輛屬性，請輸入鍵值配對，然後選擇「新增屬性」。
6. (選擇性) 若要為 AWS 資源加上標籤，請新增標籤，然後選擇 [新增標籤]。
7. 選擇下一步。
8. 要配置車輛證書，您可以上傳自己的證書或選擇自動生成新證書。我們建議您自動產生憑證以加快設定速度。如果您已經擁有憑證，則可以選擇改用憑證。
9. 下載公開金鑰和私密金鑰檔案，然後選擇 [下一步]。
10. 要將策略附加到車輛證書，您可以輸入現有策略名稱或創建新策略。若要建立新原則，請選擇 [建立原則]，然後選擇 [下一步]。
11. 檢閱您的組態。完成後，請選擇 [建立車輛]。

## 步驟 6：建立廣告活動

在 AWS IoT 中 FleetWise，營銷活動用於促進從車輛到雲端的數據選擇，收集和傳輸。促銷活動包含資料收集配置，可提供 Edge Agent 軟體指示，說明如何使用條件式收集配置或以時間為基礎的收集配置來收集資料。

### 建立行銷活動

1. 開啟 AWS IoT FleetWise 主控台。
2. 在導覽窗格中，選擇 [廣告活動]。
3. 選擇 Create campaign (建立活動)。
4. 輸入您的廣告活動名稱和可選描述。
5. 若要設定廣告活動的資料收集配置，您可以手動定義資料收集配置，或從本機裝置上傳 .json 檔案。上傳 .json 檔案會自動定義資料收集配置。
  - a. 若要手動定義資料收集配置，請選擇「定義資料收集配置」，然後選擇要用於促銷活動的資料收集配置類型。您可以選擇以條件為基礎的收集配置或以時間為基礎的收集配置。
  - b. 如果您選擇以時間為基礎的收集方案，您必須指定廣告活動收集車輛資料的時間長度。
  - c. 如果您選擇以條件為基礎的收集配置，則必須指定表示式以識別要收集的資料。請務必將訊號名稱指定為變數、比較運算子和比較值。
  - d. (選擇性) 選擇運算式的語言版本，或將其保留為預設值 1。
  - e. (選擇性) 指定兩個資料收集事件之間的觸發間隔。
  - f. 若要收集資料，請選擇 Edge Agent 軟體的觸發模式條件。依預設，AWS IoT FleetWise 軟體的 Edge 代理程式一律會在符合條件時收集資料。或者，它只能在第一次滿足條件時收集資料，開啟第一次觸發。
  - g. (可選) 您可以選擇更高級的方案選項。
6. 若要指定資料收集方案將從中收集資料的信號，請從功能表中搜尋訊號的名稱。
7. (可選) 您可以選擇最大樣本計數或最小採樣間隔。您還可以添加更多信號。
8. 選擇下一步。
9. 定義您希望促銷活動傳輸資料的儲存目的地。您可以將資料存放在 Amazon S3 或 Amazon Timestream 中。
  - a. Amazon S3 — 選擇 AWS IoT FleetWise 具有許可的 S3 儲存貯體。
  - b. Amazon Timestream — 選擇時間流資料庫和表格名稱。輸入允許 AWS IoT FleetWise 將資料傳送至「時間串流」的 IAM 角色。

10. 選擇下一步。
11. 從搜尋方塊中選擇車輛屬性或車輛名稱。
12. 輸入與您為車輛選擇的屬性或名稱相關的值。
13. 選擇您的廣告系列將從中收集資料的車輛。然後選擇下一步。
14. 檢閱廣告活動的設定，然後選擇 [建立廣告活動]。您或您的團隊必須將戰役部署到車輛上。

## 步驟 7：清除

若要避免您在本教學課程中使用的資源進一步收費，請刪除 AWS CloudFormation 堆疊和所有堆疊資源。

### 刪除 AWS CloudFormation 堆疊

1. 開啟 [AWS CloudFormation 主控台](#)。
2. 從「堆疊」清單中，選擇您在步驟 1 中建立的堆疊。
3. 選擇刪除。
4. 若要確認刪除，選擇 Delete (刪除)。堆疊大約需要 15 分鐘的時間來刪除。

## 後續步驟

1. 您可以處理並視覺化廣告活動收集的車輛資料。如需詳細資訊，請參閱 [處理和視覺化車輛資料](#)。
2. 您可以疑難排解並解決 AWS IoT 的問題 FleetWise。如需更多詳細資訊，請參閱 [AWS IoT 疑難排解 FleetWise](#)。

## 將資料導入雲端

適用於 AWS IoT FleetWise 軟體的 Edge Agent 在車輛中安裝並執行時，可促進車輛與雲端之間的安全通訊。

### Note

- AWS IoT FleetWise 不適用於任何可能導致嚴重身體傷害或死亡或造成環境或財產損害的危險環境或關鍵系統的操作，或與之相關聯。透過您使用 AWS IoT 收集的車輛資料 FleetWise 僅供參考，您不得使用 AWS IoT FleetWise 來控制或操作車輛功能。
- 通過使用 AWS IoT 收集的車輛數據 FleetWise 應根據您的用例進行適當的準確性進行評估，包括為了履行適用的車輛安全法規（例如安全監控和報告義務）下可能具有的任何合規義務。該評估應包括透過其他業界標準手段和來源（例如車輛駕駛人的報告）收集和審閱資訊。

若要將資料內嵌到雲端，請執行下列動作：

1. 在車輛中開發並安裝適用於 AWS IoT FleetWise 軟體的 Edge 代理程式。如需有關如何使用 Edge Agent 軟體的詳細資訊，請執行下列動作，下載適用於 [AWS IoT 的 Edge Agent FleetWise 軟體開發人員指南](#)。
  1. 導覽至 [AWS IoT FleetWise 主控台](#)。
  2. 在服務首頁的 [開始使用 AWS IoT] FleetWise 區段中，選擇 [探索 Edge 代理程式]。
2. 建立或匯入信號目錄，其中包含您將用來建立車輛模型的訊號。如需詳細資訊，請參閱 [建立信號目錄 \(AWS CLI\)](#) 及 [匯入信號目錄 \(AWS CLI\)](#)。

### Note

- 如果您使用 AWS IoT FleetWise 主控台建立第一個車輛型號，則不需要手動建立訊號目錄。當您建立第一個車輛型號時，AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱 [創建車輛型號 \(控制台\)](#)。
- AWS IoT FleetWise 目前支援每個 AWS 帳戶的訊號目錄 AWS 區域。

3. 使用信號目錄中的信號創建車輛模型。如需詳細資訊，請參閱 [建立車輛模型](#)。

**Note**

- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，您可以上傳 .dbc 檔案以匯入訊號。 .dbc 是控制器區域網路 (CAN 匯流排) 資料庫支援的檔案格式。建立車輛模型後，新訊號會自動新增至訊號目錄。如需詳細資訊，請參閱[創建車輛型號 \(控制台\)](#)。
- 如果您使用 CreateModelManifest API 操作來創建車輛模型，則必須使用 UpdateModelManifest API 操作來激活車輛模型。如需詳細資訊，請參閱[更新車輛型號 \(AWS CLI\)](#)。
- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，AWS IoT FleetWise 會自動為您啟動車輛型號。

4. 創建一個解碼器清單。解碼器清單包含您在上一步中創建的車輛型號中指定的每個信號的解碼信息。解碼器資訊清單與您建立的車輛型號相關聯。如需詳細資訊，請參閱[創建和管理解碼器清單](#)。

**Note**

- 如果您使用 CreateDecoderManifest API 作業建立解碼器資訊清單，您必須使用 UpdateDecoderManifest API 作業來啟動解碼器資訊清單。如需詳細資訊，請參閱[更新解碼器清單 \( \) AWS CLI](#)。
- 如果您使用 AWS IoT 主 FleetWise 控台建立解碼器資訊清單，AWS IoT FleetWise 會自動為您啟動解碼器資訊清單。

5. 從車輛模型創建車輛。使用相同車型製造的車輛會繼承相同的信號組。您必須先用 AWS IoT Core 來佈建車輛，才能將資料導入雲端。如需詳細資訊，請參閱[建立、佈建和管理車輛](#)。
6. (選擇性) 建立車隊來代表一組車輛，然後將個別車輛與車隊建立關聯。這有助於您同時管理多個車輛。如需詳細資訊，請參閱[建立和管理車隊](#)。
7. 建立行銷活動。活動被部署到車輛或車隊。行銷活動會提供 Edge Agent 軟體指示，說明如何選取、收集資料並將資料傳輸到雲端。如需詳細資訊，請參閱[透過行銷活動收集和傳輸資料](#)。

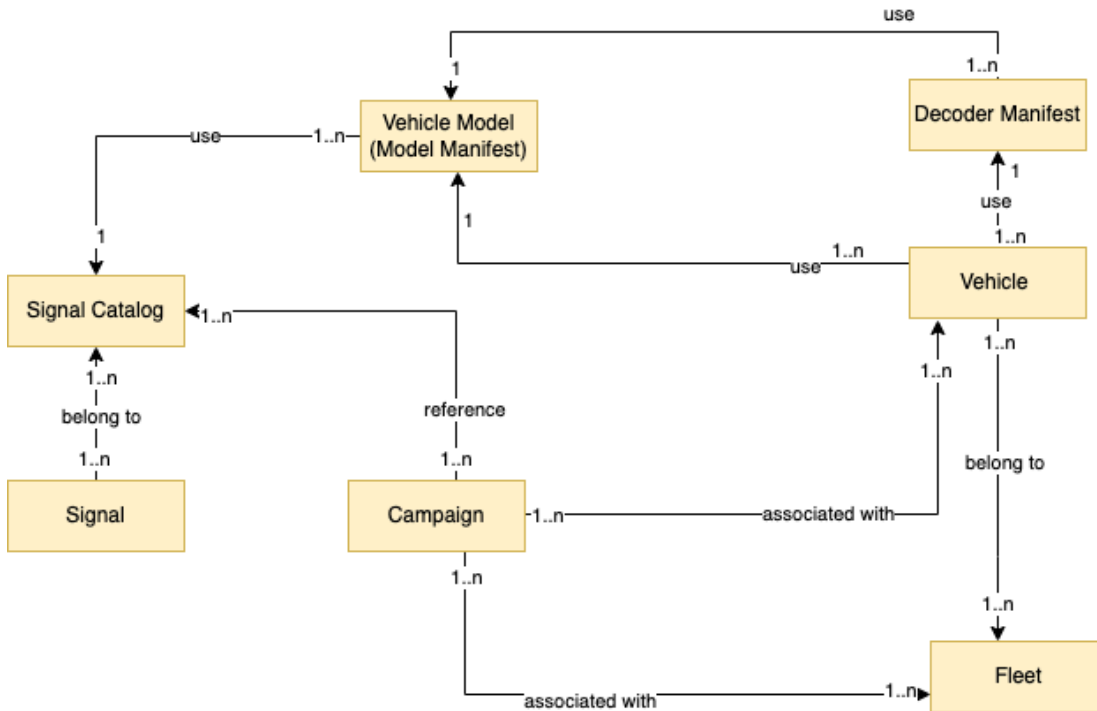
**Note**

您必須先使用 UpdateCampaign API 作業來核准行銷活動，AWS IoT FleetWise 才能將其部署到車輛或車隊。如需詳細資訊，請參閱[更新行銷活動 \(AWS CLI\)](#)。

Edge Agent 軟體會將車輛資料傳輸至AWS IoT Core使用保留的主題`aws/iotfleetwise/vehicles/vehicleName/signals`，將資料傳送至 AWS IoT FleetWise。AWS FleetWise 然後，IoT 會將資料傳送到時間流表格或 Amazon S3 儲存貯體。您可以使用時間流查詢資料，並使用 Amazon QuickSight 或 Grafana 將您的資料視覺化。如需詳細資訊，請參閱 [處理和視覺化車輛資料](#)。

## 車輛模型

AWS IoT FleetWise 提供了一個車輛建模架構，您可以使用它在雲端中建立車輛的虛擬表示。信號、訊號目錄、車輛模型和解碼器清單是您用來建模車輛的核心元件。



### Signal

信號是您定義用來包含車輛資料及其中繼資料的基本結構。信號可以是屬性，分支，傳感器或致動器。例如，您可以創建一個傳感器來接收車載溫度值，並存儲其元數據，包括傳感器名稱，數據類型和單位。如需詳細資訊，請參閱 [建立和管理信號目錄](#)。

### 信號目錄

信號目錄包含信號的集合。信號目錄中的信號可用於對使用不同協議和數據格式的車輛進行建模。例如，有兩輛汽車由不同的汽車製造商製造：一輛使用控制區域網絡（CAN 總線）協議；另一個使用板載診斷（OBD）協議。您可以在信號目錄中定義一個傳感器以接收車內溫度值。該傳感器可用於表示兩輛汽車中的熱電偶。如需詳細資訊，請參閱 [建立和管理信號目錄](#)。

### 車輛型號（模型清單）

車輛模型是聲明性結構，您可以使用它來標準化車輛的格式，並定義車輛中信號之間的關係。車輛模型會在多輛相同類型的車輛上執行一致的資訊。您可以加入訊號以建立車輛模型。如需詳細資訊，請參閱 [創建和管理車輛模型](#)。



## 解碼器清單

解碼器清單包含車輛模型中每個信號的解碼信息。車輛中的傳感器和致動器傳輸低級信息（二進制數據）。借助解碼器清單，AWS IoT 能夠將二進制數據轉換為人類可讀的 FleetWise 值。每個解碼器清單都與車輛模型相關聯。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。

您可以使用 AWS IoT FleetWise 主控台或 API 以下列方式對車輛進行模型。

1. 建立或匯入信號目錄，其中包含您將用來建立車輛模型的訊號。如需詳細資訊，請參閱 [建立信號目錄 \(AWS CLI\)](#) 及 [匯入信號目錄 \(AWS CLI\)](#)。

### Note

- 如果您使用 AWS IoT FleetWise 主控台建立第一個車輛型號，則不需要手動建立訊號目錄。當您建立第一個車輛型號時，AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱 [創建車輛型號 \(控制台\)](#)。
- AWS IoT FleetWise 目前支援每個 AWS 帳戶的訊號目錄 AWS 區域。

2. 使用信號目錄中的信號創建車輛模型。如需詳細資訊，請參閱 [建立車輛模型](#)。

### Note

- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，您可以上傳 .dbc 檔案以匯入訊號。dbc 是控制器區域網路 (CAN 匯流排) 資料庫支援的檔案格式。建立車輛模型後，新訊號會自動新增至訊號目錄。如需詳細資訊，請參閱 [創建車輛型號 \(控制台\)](#)。
- 如果您使用 CreateModelManifest API 操作來創建車輛模型，則必須使用 UpdateModelManifest API 操作來激活車輛模型。如需詳細資訊，請參閱 [更新車輛型號 \(AWS CLI\)](#)。
- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，AWS IoT FleetWise 會自動為您啟動車輛型號。

3. 創建一個解碼器清單。解碼器清單包含您在上一步中創建的車輛型號中指定的每個信號的解碼信息。解碼器資訊清單與您建立的車輛型號相關聯。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。

**Note**

- 如果您使用 CreateDecoderManifest API 作業建立解碼器資訊清單，您必須使用 UpdateDecoderManifest API 作業來啟動解碼器資訊清單。如需詳細資訊，請參閱 [更新解碼器清單 \( \) AWS CLI](#)。
- 如果您使用 AWS IoT 主 FleetWise 控制台建立解碼器資訊清單，AWS IoT FleetWise 會自動為您啟動解碼器資訊清單。

CAN 匯流排資料庫支援 .dbc 檔案格式。您可以上傳 .dbc 文件以導入信號和解碼器信號。若要取得範例 .dbc 檔案，請執行下列動作。

若要取得 .dbc 檔案

1. 下載 [EngineSignals.zip](#)。
2. 導覽至您下載 EngineSignals.zip 檔案的目錄。
3. 解壓縮檔案並將其儲存為 EngineSignals.dbc 本機。

**主題**

- [建立和管理信號目錄](#)
- [創建和管理車輛模型](#)
- [創建和管理解碼器清單](#)

## 建立和管理信號目錄

**Note**

您可以下載 [示範指令碼](#)，將 ROS 2 訊息轉換為與信號目錄相容的 VSS JSON 檔案。如需詳細資訊，請參閱 [視覺系統資料開發人員指南](#)。

信號目錄是標準化信號的集合，可以重複使用以創建車輛模型。AWS IoT FleetWise 支援 [車輛訊號規格 \(VSS\)](#)，您可以遵循這些規格來定義訊號。信號可以是以下任何一種類型。

## 屬性

屬性代表一般不會變更的靜態資訊，例如製造商和製造日期。

## 分支

分支代表嵌套結構中的信號。分支演示信號層次結構。例如，分Vehicle支有一個子分支Powertrain。該Powertrain分支有一個子分支combustionEngine。若要尋找分combustionEngine支，請使用Vehicle.Powertrain.combustionEngine運算式。

## 感測器

傳感器數據報告車輛的當前狀態並隨著時間的推移而變化，例如車輛狀態的變化，例如液位，溫度，振動或電壓。

## 傳動器

致動器資料會報告車輛裝置的狀態，例如馬達、加熱器和門鎖。更改車輛設備的狀態可以更新致動器數據。例如，您可以定義致動器來表示加熱器。當您打開或關閉加熱器，致動器接收新的數據。

## 自定義結構

自訂結構 (也稱為結構) 代表複雜或高階資料結構。它有助於邏輯綁定或源自同一來源的數據分組。當數據被讀取或寫入一個原子操作，如表示一個複雜的數據類型或高階形狀的結構使用。

結構類型的信號是使用對結構數據類型的引用而不是原始數據類型在信號目錄中定義的。結構可用於所有類型的訊號，包括感測器、屬性、致動器和視覺系統資料類型。如果發送或接收結構類型的信號，AWS IoT FleetWise 預期所有包含的項目都具有有效值，因此所有項目都是強制性的。例如，如果一個結構包含車輛的項目。相機高度，車輛攝像頭。寬度和車輛。相機。

### Note

視覺系統資料正在預覽版中，可能會有所變更。

## 自訂屬性

自訂屬性代表複雜資料結構的成員。屬性的資料類型可以是原始或其他結構。

當使用 struct 和 custom 屬性表示高階形狀時，預期的高階形狀始終被定義並顯示為樹狀結構。自定義屬性用於定義所有葉節點，而結構用於定義所有非葉節點。

### Note

- 如果您使用 AWS IoT FleetWise 主控台建立第一個車輛型號，則不需要手動建立訊號目錄。當您建立第一個車輛型號時，AWS IoT FleetWise 會自動為您建立訊號目錄。如需詳細資訊，請參閱 [創建車輛型號 \(控制台\)](#)。
- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，您可以上傳 .dbc 檔案以匯入訊號。 .dbc 是控制器區域網路 (CAN 匯流排) 資料庫支援的檔案格式。建立車輛模型後，新訊號會自動新增至訊號目錄。如需詳細資訊，請參閱 [創建車輛型號 \(控制台\)](#)。
- AWS IoT FleetWise 目前支援 AWS 帳戶 每個區域的訊號目錄。

AWS IoT FleetWise 提供下列 API 作業，您可以用來建立和管理訊號目錄。

- [CreateSignalCatalog](#)— 建立新的信號目錄。
- [ImportSignalCatalog](#)— 匯入訊號以透過上傳 JSON 檔案建立訊號目錄。信號必須遵循 VSS 定義，並以 JSON 格式保存。
- [UpdateSignalCatalog](#)— 透過更新、移除或新增信號來更新現有的信號目錄。
- [DeleteSignalCatalog](#)— 刪除既有的信號目錄。
- [ListSignalCatalogs](#)— 擷取所有訊號目錄摘要的分頁清單。
- [ListSignalCatalogNodes](#)— 擷取指定訊號目錄中所有訊號 (節點) 摘要的分頁清單。
- [GetSignalCatalog](#)— 擷取有關信號目錄的資訊。

### 教學課程

- [設定訊號](#)
- [建立信號目錄 \(AWS CLI\)](#)
- [匯入信號目錄](#)
- [更新信號目錄 \(AWS CLI\)](#)
- [刪除信號目錄 \(AWS CLI\)](#)
- [取得訊號目錄資訊 \(AWS CLI\)](#)

## 設定訊號

本節介紹如何配置分支，屬性，傳感器和致動器。

## 主題

- [配置分支](#)
- [配置屬性](#)
- [設定感測器或致動器](#)
- [設定複雜資料類型](#)

## 配置分支

若要設定分支，請指定下列資訊。

- `fullyQualifiedName`— 分支的完整名稱是分支的路徑加上分支的名稱。使用點 (.) 來指向子分支。例如，`Vehicle.Chassis.SteeringWheel`是分`SteeringWheel`支的完整名稱。`Vehicle.Chassis.`是這個分支的路徑。

完整名稱最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、冒號 (:) 和底線 (\_)。

- (選擇性) `Description` — 分支的說明。

說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_(底線) 和-(連字號)。

- (選用) `deprecationMessage` — 要移動或刪除之節點或分支的取代訊息。

取代訊息最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_(底線) 和-(連字號)。

- (選擇性) `comment` — 除了說明之外的註解。註解可用來提供有關分支的其他資訊，例如分支的基本原理或相關分支的參照。

註解最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_(底線) 和-(連字號)。

## 配置屬性

若要配置屬性，請指定下列資訊。

- `dataType`— 屬性的數據類型必須是以下其中之一:INT8, UINT8, INT16, UINT16, INT32, 搶, 肖, 布爾, 布爾, 浮動, 雙, 字符串, 時間戳, 數組, 陣列, 陣列, 陣列, 陣列, 雙陣列, 字串\_陣列、UNIX\_時間戳記\_陣列、未知或資料類型分支中定義的自訂結構。 `fullyQualifiedName` `UINT32` `INT64` `UINT64`
- `fullyQualifiedName`— 屬性的完整名稱是屬性的路徑加上屬性的名稱。使用點 (.) 來指向子信號。例如，`Vehicle.Chassis.SteeringWheel.Diameter`是屬性的完整名`Diameter`稱。`Vehicle.Chassis.SteeringWheel.`是此屬性的路徑。

完整名稱最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。

- (選擇性) Description — 屬性的描述。

說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選擇性) unit — 屬性的科學單位，例如 km 或攝氏度。
- (選擇性) min — 屬性的最小值。
- (選擇性) max — 屬性的最大值。
- (選擇性) defaultValue — 屬性的預設值。
- (選擇性) assignedValue — 指派給屬性的值。
- (選擇性) allowedValues — 屬性可接受的值清單。
- (選用) deprecationMessage — 要移動或刪除之節點或分支的取代訊息。

取代訊息最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選擇性) comment — 除了說明之外的註解。註解可用來提供有關屬性的其他資訊，例如屬性的基本原理或相關屬性的參照。

註解最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

## 設定感測器或致動器

要配置感測器或致動器，請指定以下資訊。

- dataType— 信號的數據類型必須是以下其中之一:INT8, UINT8, INT16, UINT16, INT32, 搶, 肖, 布爾, 布爾, 浮動, 雙, 字符串, 時間戳, 數組, 陣列, 陣列, 陣列, 浮點陣列, 陣列、雙陣列、字串\_陣列、UNIX\_時間戳記\_陣列、未知或資料類型分支中定義的自訂結構。 fullyQualifiedName UINT32 INT64 UINT64
- fullyQualifiedName— 信號的完全合格名稱是信號的路徑加上信號的名稱。使用點 (.) 來指向子信號。例  
如，Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState是HandsOffSteering的完整名稱。 Vehicle.Chassis.SteeringWheel.HandsOff.是通往此致動器的路徑。

完整名稱最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。

- (選擇性) Description — 訊號的描述。

說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選擇性) `unit` — 訊號的科學單位，例如 km 或攝氏度。
- (選擇性) `min` — 訊號的最小值。
- (選擇性) `max` — 訊號的最大值。
- (選擇性) `assignedValue` — 指派給訊號的值。
- (選擇性) `allowedValues` — 訊號接受的值清單。
- (選用) `deprecationMessage` — 要移動或刪除之節點或分支的取代訊息。

取代訊息最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_(底線)和-(連字號)。

- (選擇性) `comment` — 除了說明之外的註解。註解可用於提供有關感測器或致動器的其他資訊，例如其基本原理或對相關感測器或致動器的參考。

註解最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_(底線)和-(連字號)。

## 設定複雜資料類型

建模視覺系統時，會使用複雜的資料類型。除了分支之外，這些數據類型還由結構（也稱為結構）和屬性組成。結構體是由多個值（如圖像）描述的信號。屬性代表結構的成員，如原始數據類型（例如 UIN8）或另一個結構（例如時間戳）。例如，車輛相機。前代表一個分支，車輛攝像頭前。圖像表示一個結構，和車輛。相機時間戳代表一個屬性。

下列複雜資料類型範例示範如何將訊號和資料類型匯出至單一 JSON 檔案。

### Example 複雜資料類型

```
{
  "Vehicle": {
    "type": "branch"
    // Signal tree
  },
  "ComplexDataTypes": {
    "VehicleDataTypes": {
      // complex data type tree
      "children": {
        "branch": {
          "children": {
            "Struct": {
              "children": {
                "Property": {
                  "type": "property",
                  "datatype": "Data type",

```

```
        "description": "Description",
        //            ...
    }
},
"description": "Description",
"type": "struct"
}
}
"description": "Description",
"type": "branch"
}
}
}
}
```

### Note

您可以下載[示範指令碼](#)，將 ROS 2 訊息轉換為與訊號目錄相容的 VSS JSON 檔案。如需詳細資訊，請參閱[視覺系統資料開發人員指南](#)。  
視覺系統資料正在預覽版中，可能會有所變更。

## 配置結構

若要配置自訂結構 (或結構)，請指定下列資訊。

- `fullyQualifiedName`— 自訂結構的完整名稱。例如，自訂結構的完整名稱可能是 `ComplexDataTypes.VehicleDataTypes.SVMCamera`。  
完整名稱最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。
- (選擇性) `Description` — 訊號的描述。  
說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。
- (選用) `deprecationMessage` — 要移動或刪除之節點或分支的取代訊息。  
取代訊息最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。
- (選擇性) `comment` — 除了說明之外的註解。註解可用於提供有關感測器或致動器的其他資訊，例如其基本原理或對相關感測器或致動器的參考。  
註解最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。



## 配置屬性

若要規劃自訂屬性，請指定下列資訊。

- `dataType`— 信號的數據類型必須是以下其中之一:INT8, UINT8, INT16, UINT16, INT32, 搶, 肖, 布爾, 布爾, 浮動, 雙, 字符串, 時間戳, 數組, 陣列, 陣列, 陣列, 浮點陣列, 數組, 雙數組, 字符串數組, UNIX\_時間戳記\_數組, 結構, 結構\_數組, 或未知。UINT32 INT64 UINT64
- `fullyQualifiedName`— 自訂內容的完整名稱。例如, 自訂內容的完整名稱可能是 `ComplexDataTypes.VehicleDataTypes.SVMCamera.FPS`。

完整名稱最多可包含 150 個字元。有效字元: a—z、A—Z、0—9、:(冒號) 和 \_ (底線)

- (選擇性) `Description` — 訊號的描述。

說明最多可包含 2048 個字元。有效字元: a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選用) `deprecationMessage` — 要移動或刪除之節點或分支的取代訊息。

取代訊息最多可包含 2048 個字元。有效字元: a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選擇性) `comment` — 除了說明之外的註解。註解可用於提供有關感測器或致動器的其他資訊, 例如其基本原理或對相關感測器或致動器的參考。

註解最多可包含 2048 個字元。有效字元: a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

- (選擇性) `dataEncoding` — 指出內容是否為二進位資料。自訂屬性的資料編碼必須是下列其中一項: 二進位或類型化。

- (選用) `structFullyQualifiedName` — 自訂內容的資料類型為 Struct 或時, 自訂內容的結構 (struct) 節點的完整名稱。StructArray

完整名稱最多可包含 150 個字元。有效字元: a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。

## 建立信號目錄 (AWS CLI)

您可以使用 [CreateSignalCatalog](#) API 作業來建立信號目錄。下面的例子使用 AWS CLI。

若要建立信號目錄, 請執行以下指令。

替換 *signal-catalog-configuration* 為包含配置的 JSON 文件的名稱。

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

- 以您正在建立的訊號目錄名稱取 *signal-catalog-name* 代。
- (可選) 以 ## 取代描述，以協助您識別信號目錄。

如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱 [設定訊號](#)。

```
{
  "name": "signal-catalog-name",
  "description": "description",
  "nodes": [
    {
      "branch": {
        "fullyQualifiedName": "Types"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.std_msgs_Header"
      }
    },
    {
      "struct": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time"
      }
    },
    {
      "property": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time.sec",
        "dataType": "INT32",
        "dataEncoding": "TYPED"
      }
    },
    {
      "property": {
        "fullyQualifiedName": "Types.builtin_interfaces_Time.nanosec",
        "dataType": "UINT32",
        "dataEncoding": "TYPED"
      }
    }
  ]
}
```

```
},
{
  "property": {
    "fullyQualifiedName": "Types.std_msgs_Header.stamp",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.builtin_interfaces_Time"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.std_msgs_Header.frame_id",
    "dataType": "STRING",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.header",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.std_msgs_Header"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.format",
    "dataType": "STRING",
    "dataEncoding": "TYPED"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage.data",
    "dataType": "UINT8_ARRAY",
    "dataEncoding": "BINARY"
  }
},
{
  "branch": {
    "fullyQualifiedName": "Vehicle",
    "description": "Vehicle"
  }
},
{
  "branch": {
```

```
    "fullyQualifiedName": "Vehicle.Cameras"
  }
},
{
  "branch": {
    "fullyQualifiedName": "Vehicle.Cameras.Front"
  }
},
{
  "sensor": {
    "fullyQualifiedName": "Vehicle.Cameras.Front.Image",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.sensor_msgs_msg_CompressedImage"
  }
},
{
  "struct": {
    "fullyQualifiedName": "Types.std_msgs_msg_Float64"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.std_msgs_msg_Float64.data",
    "dataType": "DOUBLE",
    "dataEncoding": "TYPED"
  }
},
{
  "sensor": {
    "fullyQualifiedName": "Vehicle.Velocity",
    "dataType": "STRUCT",
    "structFullyQualifiedName": "Types.std_msgs_msg_Float64"
  }
},
{
  "struct": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
  }
},
{
  "property": {
    "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.x_offset",
    "dataType": "UINT32",
    "dataEncoding": "TYPED"
  }
}
```

```
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.y_offset",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.height",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.width",
      "dataType": "UINT32",
      "dataEncoding": "TYPED"
    }
  },
  {
    "property": {
      "fullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest.do_rectify",
      "dataType": "BOOLEAN",
      "dataEncoding": "TYPED"
    }
  },
  {
    "branch": {
      "fullyQualifiedName": "Vehicle.Perception"
    }
  },
  {
    "sensor": {
      "fullyQualifiedName": "Vehicle.Perception.Obstacle",
      "dataType": "STRUCT",
      "structFullyQualifiedName": "Types.sensor_msgs_msg_RegionOfInterest"
    }
  }
]
```

```
}
```

### Note

您可以下載[示範指令碼](#)，將 ROS 2 訊息轉換為與訊號目錄相容的 VSS JSON 檔案。如需詳細資訊，請參閱[視覺系統資料開發人員指南](#)。  
視覺系統資料正在預覽版中，可能會有所變更。

## 匯入信號目錄

您可以使用 AWS IoT FleetWise 主控台或 API 匯入訊號目錄。

### 主題

- [匯入訊號目錄 \(主控台\)](#)
- [匯入信號目錄 \(AWS CLI\)](#)

## 匯入訊號目錄 (主控台)

您可以使用 AWS IoT FleetWise 主控台匯入訊號目錄。

### Important

您最多可以有一個信號目錄。如果您已經有信號目錄，您將看不到在控制台中導入信號目錄的選項。

## 匯入信號目錄的步驟

1. 開啟 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [信號目錄]。
3. 在「信號目錄摘要」頁面上，選擇「匯入信號目錄」。
4. 匯入包含訊號的檔案。
  - 若要從 S3 儲存貯體上傳檔案：
    - a. 選擇 Import from S3 (從 S3 匯入)。
    - b. 選擇 Browse S3 (瀏覽 S3)。

- c. 在「值區」中，輸入值區名稱或物件，從清單中選擇它，然後從清單中選擇檔案。選擇「選擇檔案」按鈕。

或者，對於 S3 URI，請輸入 Amazon 簡單儲存服務 URI。如需詳細資訊，請參閱 Amazon S3 使用者指南中的[存取儲存貯體的方法](#)。

- 若要從您的電腦上傳檔案：
  - a. 選擇「從檔案匯入」。
  - b. 上傳車輛訊號規格 (VSS) 格式的 .json 檔案。

5. 確認信號目錄，然後選擇「匯入檔案」。

## 匯入信號目錄 (AWS CLI)

您可以使用 [ImportSignalCatalog](#) API 作業上傳有助於建立訊號目錄的 JSON 檔案。您必須遵循[車輛訊號規格 \(VSS\)](#)，才能將訊號儲存在 JSON 檔案中。下面的例子使用 AWS CLI。

若要匯入信號目錄，請執行以下指令。

- 以您正在建立的訊號目錄名稱取 *signal-catalog-name* 代。
- (可選) 以描述取代 *##*，以協助您識別信號目錄。
- 取代 *signal-catalog-configuration-vss* 為包含 VSS 中定義之訊號的 JSON 字串檔案名稱。

如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱[設定訊號](#)。

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

JSON 必須被字符串化並通過字段傳遞。vssJson 以下是 VSS 中定義的信號的示例。

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



```

\":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

您可以下載[示範指令碼](#)，將 ROS 2 訊息轉換為與訊號目錄相容的 VSS JSON 檔案。如需詳細資訊，請參閱[視覺系統資料開發人員指南](#)。  
視覺系統資料正在預覽版中，可能會有所變更。

## 更新信號目錄 (AWS CLI)

您可以使用 [UpdateSignalCatalog](#) API 作業來更新既有的信號目錄。下面的例子使用 AWS CLI。

若要更新既有的信號目錄，請執行以下指令。

替換 *signal-catalog-configuration* 為包含配置的 JSON 文件的名稱。

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

以您要更新的訊號目錄名稱取 *signal-catalog-name* 代。

如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱[設定訊號](#)。

### Important

自定義結構是不可變的。如果您需要重新排序或插入屬性至現有的自訂結構 (struct)，請刪除該結構並以所需的屬性順序建立全新的結構。

若要刪除自訂結構，請在中加入結構的完整名稱 `nodesToRemove`。一個結構不能被刪除，如果它是由任何信號引用。參照該結構的任何訊號 (其資料類型定義為目標結構) 必須在請求更新訊號目錄之前更新或刪除。

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

## 刪除信號目錄 (AWS CLI)

您可以使用 [DeleteSignalCatalog](#) API 操作刪除信號目錄。下面的例子使用 AWS CLI。

### Important

刪除訊號目錄之前，請確定其中沒有相關的車輛型號、解碼器清單、車輛、車隊或廣告活動。如需詳細說明，請參閱下列主題：

- [刪除車輛型號](#)
- [刪除解碼器清單](#)
- [刪除車輛](#)
- [刪除機群群群群 \(AWS CLI\)](#)
- [刪除廣告活動](#)

若要刪除既有的信號目錄，請執行以下指令。*signal-catalog-name* 替換為您要刪除的信號目錄的名稱。

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

### Note

此命令不會產生輸出。

## 取得訊號目錄資訊 (AWS CLI)

您可以使用 [ListSignalCatalogs](#) API 操作來驗證是否已刪除訊號目錄。下面的例子使用 AWS CLI。

若要擷取所有訊號目錄摘要的分頁清單，請執行下列命令。

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

您可以使用 [ListSignalCatalogNodes](#) API 操作來驗證訊號目錄是否已更新。下面的例子使用 AWS CLI。

若要擷取給定訊號目錄中所有訊號 (節點) 摘要的分頁清單，請執行下列命令。

以您要檢查的訊號目錄名稱取 *signal-catalog-name* 代。

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

您可以使用 [GetSignalCatalog](#) API 作業擷取訊號目錄資訊。下面的例子使用 AWS CLI。

若要擷取有關訊號目錄的資訊，請執行下列命令。

*signal-catalog-name* 以您要擷取的訊號目錄名稱取代。

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

### Note

此操作為 [最終一致](#)。換句話說，對訊號目錄的變更可能不會立即反映出來。

## 創建和管理車輛模型

您可以使用訊號來創建車輛模型，以幫助標準化車輛的格式。車輛模型會在多輛相同類型的車輛上執行一致的資訊，因此您可以處理來自車輛車隊的資料。使用相同車型製造的車輛會繼承相同的訊號組。如需詳細資訊，請參閱 [建立、佈建和管理車輛](#)。

每個車輛型號都有一個狀態欄位，其中包含車輛模型的狀態。狀態可以是下列其中一個值：

- ACTIVE— 車輛模型處於活動狀態。

- DRAFT— 保存了車輛模型的配置。

#### Important

- 如果要使用 CreateModelManifest API 操作來創建第一個車輛模型，則必須先創建一個信號目錄。如需詳細資訊，請參閱 [建立信號目錄 \(AWS CLI\)](#)。
- 如果您使用 AWS IoT FleetWise 主控台建立車輛型號，AWS IoT FleetWise 會自動為您啟動車輛型號。
- 如果您使用 CreateModelManifest API 操作來創建車輛模型，則車輛模型將保持在 DRAFT 狀態。
- 您無法從 DRAFT 處於該州的車輛模型建立車輛。使用 UpdateModelManifest API 操作將車輛模型更改為狀 ACTIVE 態。
- 您無法編輯處於此 ACTIVE 狀態的車輛模型。

#### 主題

- [建立車輛模型](#)
- [更新車輛型號 \( AWS CLI \)](#)
- [刪除車輛型號](#)
- [取得車輛型號資訊 \(AWS CLI\)](#)

## 建立車輛模型

您可以使用 AWS IoT FleetWise 主控台或 API 建立車輛模型。

#### Important

您必須具有信號目錄，然後才能使用 CreateModelManifest API 操作來創建車輛模型。

#### 主題

- [創建車輛型號 \( 控制台 \)](#)
- [建立車輛模型 \(AWS CLI\)](#)

## 創建車輛型號 ( 控制台 )

在 AWS IoT 主 FleetWise 控制台中，您可以透過下列方式建立車輛模型：

- [使用提供的範本 AWS](#)
- [手動建立車輛模型](#)
- [複製車輛模型](#)

### 使用提供的範本 AWS

AWS IoT FleetWise 提供車載診斷 (OBD) II、J1979 範本，可自動為您建立訊號目錄、車輛型號和解碼器清單。該模板還將 OBD 網路接口添加到解碼器清單。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。

### 使用範本建立車輛模型

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 在「車輛型號」頁面上，選擇「新增提供的範本」。
4. 選擇板載診斷 ( OBD ) II。
5. 輸入 AWS IoT FleetWise 正在建立的 OBD 網路介面名稱。
6. 選擇新增。

### 手動建立車輛模型

您可以透過上傳一個或多個 .dbc 檔案來新增訊號目錄中的訊號或匯入訊號。A .dbc 文件是控制器區域網路 ( CAN 總線 ) 數據庫支持的文件格式。

#### Important

您無法使用 AWS IoT FleetWise 主控台建立具有視覺系統資料訊號的車輛模型。而是使用 AWS CLI 建立車輛模型。

視覺系統資料正在預覽版中，可能會有所變更。

### 手動建立車輛模型

1. 導覽至 [AWS IoT FleetWise 主控台](#)。

2. 在導覽窗格中，選擇 [車輛型號]。
3. 在「車輛型號」頁面上，選擇「建立車輛型號」，然後執行下列動作。

## 主題

- [步驟 1：設定車型](#)
- [步驟 2：添加信號](#)
- [步驟 3：導入信號](#)
- [\(選擇性\) 步驟 4：新增屬性](#)
- [步驟 4：檢閱和建立](#)

### 步驟 1：設定車型

在 [一般資訊] 中，執行下列動作。

1. 輸入車輛型號的名稱。
2. (選用) 輸入描述。
3. 選擇下一步。

### 步驟 2：添加信號

#### Note

- 如果這是您第一次使用 AWS IoT FleetWise，則此步驟在您擁有信號目錄之前無法使用。建立第一輛車型時，AWS IoT FleetWise 會自動建立訊號目錄，並將訊號新增至第一輛車型。
- 如果您對 AWS IoT 有經驗 FleetWise，可以從信號目錄中選擇信號或上傳 .dbc 文件以導入信號，將信號添加到車輛型號中。
- 您必須至少有一個訊號才能建立車輛模型。

### 若要新增信號

1. 從您要新增至車輛型號的訊號目錄中選擇一或多個訊號。您可以在右窗格中查看選定的信號。

**Note**

只有選定的訊號才會新增至車輛型號。

**2. 選擇下一步。****步驟 3：導入信號****Note**

- 如果這是您第一次使用 AWS IoT FleetWise，您必須上傳至少一個 .dbc 檔案才能匯入訊號。
- 如果您對 AWS IoT 有經驗 FleetWise，可以從信號目錄中選擇信號或上傳 .dbc 文件以導入信號，將信號添加到車輛型號中。
- 您必須至少有一個訊號才能建立車輛模型。

**匯入信號**

1. 選擇「選擇檔案」。
2. 在對話方塊中，選擇包含信號的 .dbc 檔案。您可以上傳多個 .dbc 檔案。
3. AWS IoT FleetWise 會剖析您的 .dbc 檔案以擷取訊號。

在「信號」區段中，為每個訊號指定下列中繼資料。

- 名稱 — 訊號的名稱。

信號名稱必須是唯一的。訊號名稱加上路徑最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。

- 數據類型 — 信號的數據類型必須是以下其中之一：  
INT8，UINT8，INT16，UINT16，INT32，，夯，屏，布爾，浮動，雙，字符串，時間戳，數組，陣列，陣列，陣列，AT\_陣列、雙陣列、字串\_陣列、UNIX 時間戳記\_陣列或未知。UINT32 INT64 UINT64
- 信號類型 — 信號的類型，可以是傳感器或致動器。
- (選擇性) 單位 — 訊號的科學單位，例如公里或攝氏度。

- (選擇性) 路徑 — 訊號的路徑。類似於 JSONPath，使用一個點 ( . ) 來指一個子信號。例如 **Vehicle.Engine.Light**。

訊號名稱加上路徑最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)。

- (選擇性) 最小值 — 訊號的最小值。
- (選擇性) Max — 訊號的最大值。
- (選擇性) 說明 — 訊號的描述。

說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

#### 4. 選擇下一步。

##### (選擇性) 步驟 4：新增屬性

您最多可以加入 100 個屬性，包括信號目錄中的既有屬性。

##### 加入屬性的步驟

#### 1. 在新增屬性中，為每個屬性指定下列中繼資料。

- 名稱 — 屬性的名稱。

信號名稱必須是唯一的。訊號名稱和路徑最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)

- 數據類型-屬性的數據類型必須是以下其中之一：INT8，UINT8，INT16，UINT16，INT32，欺騙，贊，布爾，浮動，雙，字符串，時間戳，數組，數組，陣列，AT\_陣列、雙陣列、字串 \_陣列、UNIX\_時間戳記 \_陣列或未知 UIN32 INT64 UIN64
- (選擇性) 單位 — 屬性的科學單位，例如 km 或攝氏度。
- (選擇性) 路徑 — 訊號的路徑。類似於 JSONPath，使用一個點 ( . ) 來指一個子信號。例如 **Vehicle.Engine.Light**。

訊號名稱加上路徑最多可包含 150 個字元。有效字元：a—z、A—Z、0—9、:(冒號) 和 \_ (底線)

- (選擇性) 最小值 — 屬性的最小值。
- (選擇性) Max — 屬性的最大值。
- (選擇性) 說明 — 屬性的描述。

說明最多可包含 2048 個字元。有效字元：a—z、A—Z、0—9、:(冒號)、\_ (底線) 和-(連字號)。

## 2. 選擇下一步。

### 步驟 4：檢閱和建立

確認車輛模型的組態，然後選擇 [建立]。

### 複製車輛模型

AWS IoT FleetWise 可以複製現有車輛模型的配置以創建新模型。所選車輛型號中指定的信號會複製到新車型。

### 複製車輛模型

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 從車輛型號清單中選擇車型，然後選擇「複製模型」。

要配置車輛模型，請按照[手動建立車輛模型](#)教程進行操作。

AWS IoT 可能需要幾分鐘的時間 FleetWise 來處理您建立車輛模型的要求。成功建立車輛模型後，在「車輛型號」頁面上，「狀態」欄會顯示「作用中」。當車輛模型變為作用中時，您無法對其進行編輯。

### 建立車輛模型 (AWS CLI)

您可以使用 [CreateModelManifest](#) API 操作來創建車輛模型（模型清單）。下列為使用 AWS CLI 的範例。

#### Important

如果要使用 AWS IoT FleetWise API 建立第一個車輛模型，則必須先建立訊號目錄。若要取得有關如何建立訊號目錄的更多資訊，請參閱 [〈〉 建立訊號目錄 \(AWS CLI\)](#)。

若要建立車輛模型，請執行下列指令。

替換 *vehicle-model-configuration* 為包含配置的 JSON 文件的名稱。

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



- 以您正在建立的車輛型號名稱取 *vehicle-model-name* 代。
- 用 *##### Amazon #####ARN#####- ARN*。
- (選擇性) 以 *##* 取代說明，以協助您識別車輛型號。

如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱 [設定訊號](#)。

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

## 更新車輛型號 ( AWS CLI )

您可以使用 [UpdateModelManifestAPI](#) 操作來更新現有的車輛模型 ( 模型清單 )。下列為使用 AWS CLI 的範例。

若要更新現有的車輛型號，請執行以下指令。

替換 *update-vehicle-model-configuration* 為包含配置的 JSON 文件的名稱。

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

- 以您要更新的車輛型號名稱取 *vehicle-model-name* 代。
- (選擇性) 若要啟動車輛型號，請取代 *vehicle-model-status* 為 ACTIVE。

### Important

啟動車輛模型後，您將無法變更車輛型號。

- (選擇性) 以更新後的 *##* 取代說明，以協助您識別車輛型號。

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

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

## 刪除車輛型號

您可以使用 AWS IoT FleetWise 主控台或 API 刪除車輛型號。

### Important

必須先刪除與車輛型號相關聯的車輛和解碼器清單。如需詳細資訊，請參閱 [刪除車輛](#) 及 [刪除解碼器清單](#)。

## 刪除車輛型號 ( 控制台 )

若要刪除車輛型號，請使用 AWS IoT FleetWise 主控台。

### 刪除車輛型號

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 在「車輛型號」頁面上，選擇目標車輛型號。
4. 選擇刪除。
5. 在刪除 **vehicle-model-name**？ 中，輸入要刪除的車輛型號名稱，然後選擇 [確認]。

## 刪除車輛型號 (AWS CLI)

您可以使用 [DeleteModelManifest](#) API 操作刪除現有的車輛模型 ( 模型清單 )。下列為使用 AWS CLI 的範例。

若要刪除車輛型號，請執行下列指令。

以您要刪除的車輛型號名稱取 *model-manifest-name* 代。

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

**Note**

此命令不會產生輸出。

## 取得車輛型號資訊 (AWS CLI)

您可以使用 [ListModelManifests](#) API 操作來驗證車輛型號是否已被刪除。下面的例子使用 AWS CLI。

若要擷取所有車輛型號摘要的分頁清單，請執行以下指令。

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

您可以使用 [ListModelManifestNodes](#) API 操作來驗證車輛型號是否已更新。下面的例子使用 AWS CLI。

要檢索給定車輛型號中所有信號 ( 節點 ) 摘要的分頁列表，請運行以下命令。

以您要檢查的車輛型號名稱取 *vehicle-model-name* 代。

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

若要擷取有關車輛型號的資訊，請執行下列指令。

用您要檢索的車 ## 號名稱替換車輛型號。

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

**Note**

此操作為 [最終一致](#)。換句話說，車輛型號的變更可能不會立即反映出來。

## 創建和管理解碼器清單

解碼器清單包含 AWS IoT FleetWise 用於將車輛數據 ( 二進制數據 ) 轉換為人類可讀的值的解碼信息，並準備數據以進行數據分析。網路介面和解碼器訊號是您用來設定解碼器資訊清單的核心元件。

## 網路介面

包含車載網路使用的協議的信息。AWS IoT FleetWise 支持以下協議。

### 控制器區域網路 (CAN 匯流排)

一種協議，用於定義數據如何在電子控制單元 ( ECU ) 之間進行通信。ECU 可以是引擎控制單元、安全氣囊或音訊系統。

### 車載診斷系統 (OBD) II

進一步開發的通訊協定，用於定義 ECU 之間如何通訊自我診斷資料。它提供了一些標準的診斷故障代碼 ( DTC ) ，有助於識別您的車輛出了什麼問題。

### 車輛中間件

車輛中間件定義為一種類型的網路接口。車輛中介軟體的範例包括機器人作業系統 (ROS 2) 和 IP 上的可擴充服務導向中介軟體 (SOME/IP)。

#### Note

AWS IoT FleetWise 支援視覺系統資料的 ROS 2 中介軟體。

## 解碼器信號

提供特定訊號的詳細解碼資訊。車輛型號中指定的每個信號都必須與解碼器信號配對。如果解碼器清單包含 CAN 網路接口，則必須包含 CAN 解碼器信號。如果解碼器清單包含 OBD 網路接口，則必須包含 OBD 解碼器信號。

如果解碼器清單還包含車輛中間件接口，則解碼器清單必須包含消息解碼器信號。

每個解碼器清單必須與車輛型號相關聯。AWS IoT FleetWise 使用相關的解碼器清單來解碼基於車輛型號創建的車輛的數據。

每個解碼器資訊清單都有一個狀態欄位，其中包含解碼器資訊清單的狀態。狀態可以是下列其中一個值：

- ACTIVE— 解碼器清單處於活動狀態。
- DRAFT— 解碼器清單的配置未保存。
- VALIDATING— 解碼器清單正在驗證其資格。這只適用於至少包含一個視覺系統資料訊號的解碼器清單。

- INVALID— 解碼器清單驗證失敗，無法激活尚未。這只適用於至少包含一個視覺系統資料訊號的解碼器清單。您可以使用 ListDecoderManifests 和 GetDecoderManifest API 來檢查驗證失敗的原因。

### Important

- 如果您使用 AWS IoT 主 FleetWise 控制台建立解碼器資訊清單，AWS IoT FleetWise 會自動為您啟動解碼器資訊清單。
- 如果您使用 CreateDecoderManifest API 作業建立解碼器資訊清單，解碼器資訊清單會保持在 DRAFT 狀態。
- 您無法從與 DRAFT 解碼器清單相關聯的車輛型號創建車輛。使用 UpdateDecoderManifest API 操作將解碼器清單更改為 ACTIVE 狀態。
- 您無法編輯處於 ACTIVE 於該狀態的解碼器清單。

## 主題

- [設定網路介面和解碼器訊號](#)
- [創建解碼器清單](#)
- [更新解碼器清單 \( \) AWS CLI](#)
- [刪除解碼器清單](#)
- [獲取解碼器清單信息 \( \) AWS CLI](#)

## 設定網路介面和解碼器訊號

每個解碼器清單至少具有與相關車輛型號中指定的信號配對的網路接口和解碼器信號。

如果解碼器清單包含 CAN 網路接口，則必須包含 CAN 解碼器信號。如果解碼器清單包含 OBD 網路接口，則必須包含 OBD 解碼器信號。

## 主題

- [設定網路介面](#)
- [設定解碼器訊號](#)

## 設定網路介面

若要設定 CAN 網路介面，請指定下列資訊。

- `name`— CAN 介面的名稱。

介面名稱必須是唯一的，且可包含 1—100 個字元。

- (選擇性) `protocolName` — 通訊協定的名稱。

有效值：CAN-FD和 CAN

- (可選) `protocolVersion`— AWS IoT FleetWise 目前支持 CAN-FD 和 CAN 2.0。

有效值：1.0和 2.0b

若要設定 OBD 網路介面，請指定下列資訊。

- `name`— OBD 介面的名稱。

介面名稱必須是唯一的，且可包含 1—100 個字元。

- `requestMessageId`— 要求資料之訊息的 ID。

- (可選) `dtcRequestIntervalSeconds`— 在幾秒鐘內從車輛請求診斷故障代碼 (DTC) 的頻率。例如，如果指定的值為 120，Edge 代理程式軟體會每 2 分鐘收集一次儲存的 DTC。

- (可選) `hasTransmissionEcu`— 車輛是否具有變速箱控制模塊 (TCM)。

有效值：true和 false

- (選用) `obdStandard` — AWS IoT FleetWise 支援的 OBD 標準。AWS IoT FleetWise 目前支援全球協調車載診斷 (WWH OBD) ISO15765-4 標準。

- (可選) `pidRequestIntervalSeconds`— 從車輛請求 OBD II PID 的頻率。例如，如果指定的值為 120，邊緣代理程式軟體會每 2 分鐘收集一次 OBD II PID。

- (選用) `useExtendedIds` — 是否要在訊息中使用延伸 ID。

有效值：true和 false

若要設定車輛中介軟體網路介面，請指定下列資訊。

- `name`— 車輛中間件接口的名稱。

介面名稱必須是唯一的，且可包含 1—100 個字元。

- `protocolName`— 通訊協定的名稱。

有效值：R0S\_2

## 設定解碼器訊號

若要設定 CAN 解碼器訊號，請指定下列資訊。

- `factor`— 用於解碼消息的乘數。
- `isBigEndian`— 訊息的位元組順序是否為大端。如果它是大端，序列中最重要值會先儲存在最低的儲存位址。
- `isSigned`— 郵件是否已簽署。如果已簽署，郵件可以代表正數和負數。
- `length`— 訊息的長度 (以位元組為單位)。
- `messageId`— 訊息的識別碼。
- `offset`— 用於計算信號值的偏移量。與係數相結合，計算為  $value = raw\_value * factor + offset$ 。
- `startBit`— 指示訊息第一個位元的位置。
- (選擇性) `name` — 訊號的名稱。

若要設定 OBD 解碼器訊號，請指定下列資訊。

- `byteLength`— 訊息的長度 (以位元組為單位)。
- `offset`— 用於計算信號值的偏移量。結合縮放比例，計算為  $value = raw\_value * scaling + offset$ 。
- `pid`— 用於向車輛請求消息的診斷代碼以獲取此信號。
- `pidResponseLength`— 要求訊息的長度。
- `scaling`— 用於解碼消息的乘數。
- `serviceMode`— 消息中的操作模式 (診斷服務)。
- `startByte`— 指示訊息的開頭。
- (選擇性) `bitMaskLength` — 訊息中遮罩的位元數目。
- (選擇性) `bitRightShift` — 向右移動的位置數。

若要設定訊息解碼器訊號，請指定下列資訊。

- `topicName`— 訊息訊號的主題名稱。它對應於 ROS 2 中的主題。如需結構化訊息物件的詳細資訊，請參閱 [StructuredMessage](#)。
- `structuredMessage`— 訊息訊號的結構化訊息。它可以使用 a `primitiveMessageDefinition`，定義或 `structuredMessageDefinition` 遞歸來定 `structuredMessageList` 義。

## 創建解碼器清單

您可以使用 AWS IoT FleetWise 主控台或 API 為您的車輛型號建立解碼器清單。

### Important

您必須擁有車輛型號，然後才能創建解碼器清單。每個解碼器清單必須與車輛型號相關聯。如需詳細資訊，請參閱 [創建和管理車輛模型](#)。

### 主題

- [創建解碼器清單 \(控制台\)](#)
- [創建解碼器清單 \( \) AWS CLI](#)

## 創建解碼器清單 (控制台)

您可以使用 AWS IoT FleetWise 主控台建立與車輛型號相關聯的解碼器資訊清單。

### Important

您無法使用 AWS IoT FleetWise 主控台在解碼器資訊清單中設定視覺系統資料訊號。而是使用 AWS CLI。視覺系統資料正在預覽版中，可能會有所變更。

### 若要建立解碼器資訊清單

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 選擇目標車型。
4. 在車輛型號摘要頁面上，選擇 [建立解碼器資訊清單]，然後執行下列動作。



## 主題

- [步驟 1：配置解碼器清單](#)
- [步驟 2：新增網路介面](#)
- [步驟 3：檢閱和建立](#)

### 步驟 1：配置解碼器清單

在 [一般資訊] 中，執行下列動作。

1. 輸入解碼器資訊清單的唯一名稱。
2. (選用) 輸入描述。
3. 選擇下一步。

### 步驟 2：新增網路介面

每個解碼器資訊清單必須至少有一個網路介面。您可以將多個網路介面新增至解碼器資訊清單。

#### 新增網路介面

- 在 [網路介面] 中，執行下列動作。
  - a. 對於「網路介面類型」，請選擇「CAN\_ 介面」或「OBD\_ 介面」。
  - b. 輸入網路介面的唯一名稱。
  - c. 輸入唯一的網路介面識別碼。您可以使用 AWS IoT 產生的識別碼 FleetWise。
  - d. 選擇車輛型號中指定的一個或多個信號以與解碼器信號配對。
  - e. 若要提供解碼資訊，請上傳 .dbc 檔案。AWS IoT 會 FleetWise 剖析 .dbc 檔案以擷取解碼器訊號。
  - f. 在配對信號部分，確保每個信號都與解碼器信號配對。
  - g. 選擇下一步。

#### Note

- 您只能為每個網路介面上傳一個 .dbc 檔案。
- 確保您車輛型號中指定的每個訊號都與解碼器訊號配對。

- 選擇新增其他網路介面後，就無法編輯您正在編輯的介面。您可以刪除任何現有的網路介面。

### 步驟 3：檢閱和建立

驗證解碼器資訊清單的組態，然後選擇 [建立]。

### 創建解碼器清單 ( ) AWS CLI

您可以使用 [CreateDecoderManifest](#) API 操作來創建解碼器清單。下列為使用 AWS CLI 的範例。

#### Important

在創建解碼器清單之前，請先創建一個車輛模型。如需詳細資訊，請參閱 [建立車輛模型](#)。

若要建立解碼器資訊清單，請執行下列命令。

替換 *decoder-manifest-configuration* 為包含配置的 JSON 文件的名稱。

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

- 替換 *decoder-manifest-name* 為您正在創建的解碼器清單的名稱。
- 用 `##### Amazon #####ARN` 替換車輛模型 ARN。
- (選擇性) 以 `##` 取代說明，以協助您識別解碼器資訊清單。

如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱 [設定網路介面和解碼器訊號](#)。

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

```

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

```

- 替換 *decoder-manifest-name* 為您正在創建的解碼器清單的名稱。
- 用 `##### Amazon #####ARN` ) 替換車輛模型 ARN。
- (選擇性) 以 `##` 取代說明，以協助您識別解碼器資訊清單。

結構 ( 結構 ) 中屬性節點的順序必須保持與信號目錄和車輛模型 ( 模型清單 ) 中定義的一致。如需如何設定分支、屬性、感測器和致動器的詳細資訊，請參閱[設定網路介面和解碼器訊號](#)。

```
{
  "name": "decoder-manifest-name",
  "modelManifestArn": "vehicle-model-arn",
  "description": "description",
  "networkInterfaces": [{
    "canInterface": {
      "name": "myNetworkInterface",
      "protocolName": "CAN",
      "protocolVersion": "2.0b"
    },
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_INTERFACE"
  }, {
    "type": "VEHICLE_MIDDLEWARE",
    "interfaceId": "G1KzxkdnmV5Hn7wkV3ZL9",
    "vehicleMiddleware": {
      "name": "ROS2_test",
      "protocolName": "ROS_2"
    }
  }
  ]],
  "signalDecoders": [{
    "canSignal": {
      "name": "Engine_Idle_Time",
      "factor": 1,
      "isBigEndian": true,
      "isSigned": false,
      "length": 24,
      "messageId": 271343712,
      "offset": 0,
      "startBit": 16
    },
    "fullyQualifiedName": "Vehicle.EngineIdleTime",
    "interfaceId": "Qq1acaenBy0B3sSM39SYm",
    "type": "CAN_SIGNAL"
  },
  {
    "canSignal": {
      "name": "Engine_Run_Time",
      "factor": 1,
      "isBigEndian": true,
```

```

    "isSigned": false,
    "length": 24,
    "messageId": 271343712,
    "offset": 0,
    "startBit": 40
  },
  "fullyQualifiedName": "Vehicle.EngineRunTime",
  "interfaceId": "Qq1acaenBy0B3sSM39SYm",
  "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

您可以下載[演示腳本](#)以創建帶有視覺系統信號的解碼器清單。如需詳細資訊，請參閱[視覺系統資料開發人員指南](#)。

視覺系統資料正在預覽版中，可能會有所變更。

## 更新解碼器清單 ( ) AWS CLI

您可以使用 [UpdateDecoderManifest](#) API 操作來更新解碼器清單。您可以新增、移除和更新網路介面和訊號解碼器。您也可以變更解碼器資訊清單的狀態。下列為使用 AWS CLI 的範例。

若要更新解碼器資訊清單，請執行下列命令。

替換 *decoder-manifest-name* 為您正在更新的解碼器清單的名稱。

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

### Important

激活解碼器清單後，您無法對其進行編輯。

## 刪除解碼器清單

您可以使用 AWS IoT FleetWise 主控台或 API 刪除解碼器資訊清單。

### Important

必須先刪除與解碼器清單相關聯的車輛。如需詳細資訊，請參閱 [刪除車輛](#)。

## 主題

- [刪除解碼器清單 \(控制台\)](#)
- [刪除解碼器清單 \( \) AWS CLI](#)

## 刪除解碼器清單 (控制台)

您可以使用 AWS IoT 主 FleetWise 控制台刪除解碼器資訊清單。

若要刪除解碼器資訊清單

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛型號]。
3. 選擇目標車型。
4. 在車輛型號摘要頁面中，選擇「解碼器清單」分頁。
5. 選擇目標解碼器資訊清單，然後選擇 [刪除]。
6. 在刪除 **decoder-manifest-name** ? ，輸入要刪除的解碼器資訊清單名稱，然後選擇 [確認]。

## 刪除解碼器清單 ( ) AWS CLI

您可以使用 [DeleteDecoderManifest](#) API 操作來刪除解碼器清單。下面的例子使用 AWS CLI。

### Important

刪除解碼器資訊清單之前，請先刪除相關車輛。如需詳細資訊，請參閱 [刪除車輛](#)。

若要刪除解碼器資訊清單，請執行下列命令。

*decoder-manifest-name* 替換為您要刪除的解碼器清單的名稱。

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

## 獲取解碼器清單信息 ( ) AWS CLI

您可以使用 [ListDecoderManifests](#) API 操作來驗證解碼器清單是否已被刪除。下面的例子使用 AWS CLI。

若要擷取所有解碼器資訊清單摘要的分頁清單，請執行下列命令。



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

您可以使用 [ListDecoderManifestSignals](#) API 操作來驗證解碼器清單中的解碼器信號是否已更新。下面的例子使用 AWS CLI。

要檢索給定解碼器清單中所有解碼器信號 ( 節點 ) 的摘要的分頁列表，請運行以下命令。

替換 *decoder-manifest-name* 為您正在檢查的解碼器清單的名稱。

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

您可以使用 [ListDecoderManifestNetworkInterfaces](#) API 操作來驗證解碼器清單中的網絡接口是否已更新。以下範例使用 AWS CLI。

若要擷取指定解碼器資訊清單中所有網路介面之摘要的分頁清單，請執行下列命令。

替換 *decoder-manifest-name* 為您正在檢查的解碼器清單的名稱。

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

您可以使用 [GetDecoderManifest](#) API 操作來驗證解碼器清單中的網絡接口和解碼器信號是否已更新。下面的例子使用 AWS CLI。

若要擷取有關解碼器資訊清單的資訊，請執行下列命令。

用您要檢索的 ##### 的名稱替換解碼器清單。

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

#### Note

此操作為 [最終一致](#)。換句話說，對解碼器清單的更改可能不會立即反映出來。

## 建立、佈建和管理車輛

車輛是車輛模型的實例。車輛必須從車輛模型中創建並與解碼器清單相關聯。車輛會將一或多個資料串流上傳至雲端。例如，車輛可以將里程、引擎溫度和加熱器資料狀態傳送至雲端。每輛車都包含以下信息：

### vehicleName

識別車輛的 ID。

請勿在您的車輛名稱中添加個人身份信息 (PII) 或其他機密或敏感信息。車輛名稱可以通過其他 AWS 服務 (包括 Amazon) 訪問 CloudWatch。車輛名稱不適用於私人或敏感資料。

### modelManifestARN

車輛模型 (模型清單) 的 Amazon 資源名稱 (ARN)。每輛車都是從車輛模型創建的。使用相同車型製造的車輛包含從車輛模型繼承的同一組信號。這些信號在信號目錄中定義和標準化。

### decoderManifestArn

解碼器的 ARN 清單。解碼器資訊清單提供解碼資訊，AWS IoT FleetWise 可用來將原始訊號資料 (二進位資料) 轉換為人類可讀的值。解碼器清單必須與車輛型號相關聯。AWS IoT FleetWise 使用相同的解碼器清單來解碼基於相同車輛型號創建的車輛的原始數據。

### attributes

屬性是包含靜態資訊的鍵值配對。車輛可以包含繼承自車輛模型的屬性。您可以加入其他屬性，以區分個別車輛與從同一車輛模型建立的其他車輛。例如，如果您有一輛黑色的汽車，則可以為屬性指定以下值：{"color": "black"}

#### Important

屬性必須先在關聯的車輛模型中定義，才能將其新增至個別車輛。

有關車輛型號、解碼器清單和屬性的更多資訊，請參閱 [車輛模型](#)

AWS IoT FleetWise 提供下列 API 作業，您可以使用這些作業來建立和管理車輛。

- [CreateVehicle](#)— 創建一個新的車輛。
- [BatchCreateVehicle](#)— 創建一個或多個新的車輛。
- [UpdateVehicle](#)— 更新現有車輛。

- [BatchUpdateVehicle](#)— 更新一個或多個現有車輛。
- [DeleteVehicle](#)— 刪除現有車輛。
- [ListVehicles](#)-檢索所有車輛摘要的分頁列表。
- [GetVehicle](#)— 檢索有關車輛的信息。

## 教學課程

- [提供車輛](#)
- [預留主題](#)
- [建立車輛](#)
- [更新車輛 \( AWS CLI \)](#)
- [更新多輛車 \( AWS CLI \)](#)
- [刪除車輛](#)
- [取得車輛資訊 \(AWS CLI\)](#)

## 提供車輛

在您車輛中執行的 AWS IoT FleetWise 軟體邊緣代理程式會收集資料並將其傳輸到雲端。AWS 與 IoT FleetWise 整合，AWS IoT Core 以支援邊緣代理程式軟體與透過 MQTT 雲端之間的安全通訊。每輛車都對應一個 AWS IoT 東西。您可以使用現有物 AWS IoT 件來建立車輛，或 FleetWise 將 AWS IoT 設定為您的車輛自動建立物 AWS IoT 件。如需詳細資訊，請參閱 [建立車輛 \(AWS CLI\)](#)。

AWS IoT Core 支援[驗證](#)和[授權](#)，協助安全地控制 AWS IoT 資 FleetWise 源的存取。車輛可以使用 X.509 憑證取得驗證 (登入)，以使用 AWS IoT FleetWise 和 AWS IoT Core 政策取得授權 (具有權限) 來執行指定動作。

## 驗證車輛

您可以建立 AWS IoT Core 政策來驗證車輛。

### 驗證您的車輛

- 若要建立 AWS IoT Core 原則，請執行下列命令。
  - 將####取代為您要建立的原則名稱。
  - 將檔#####包含 AWS IoT Core 原則的 JSON 檔案名稱。

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

使用範例原則之前，請先執行下列動作：

- 將##取代為您建立 AWS IoT FleetWise 資源的 AWS 區域。
- 將 **AWS #####** AWS 戶 ID。

此範例包括 AWS IoT 保留的主題 FleetWise。您必須將主題新增至策略。如需詳細資訊，請參閱 [預留主題](#)。

```
{
  "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"
    ]
}
]
}

```

## 授權車輛

您可以建立 X.509 憑證來授權您的車輛。

授權您的車輛

### Important

我們建議您為每輛車建立新的憑證。

- 若要建立 RSA key pair 並發出 X.509 憑證，請執行下列命令。
  - 將證#替換為保存證書 PEM 命令輸出內容的文件的名称。
  - 將##替換為保存密鑰對的命令輸出內容的文件的名称。 PublicKey。
  - 將#####對的命令輸出內容的文件的名称。 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. 從輸出複製憑證的 Amazon 資源名稱 (ARN)。
3. 若要將原則附加至憑證，請執行下列命令。
  - 將#####立的原 AWS IoT Core 則名稱。
  - 將## *arn* #####的憑證的 ARN。

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

4. 若要將憑證附加至物件，請執行下列命令。
  - 將##### *AWS IoT* #####輛的 ID。
  - 將## *arn* #####的憑證的 ARN。

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

## 預留主題

AWS IoT FleetWise 保留下列主題的使用。如果保留主題允許，您可以訂閱或發佈至該主題。不過，您無法建立以美元符號 (\$) 開頭的新主題。如果您對保留主題使用不支援的發佈或訂閱作業，可能會導致連線結束。

| 主題  | 允許用戶端作業 | 描述  |
|---|---------|---|
| \$aws/iotfleetwise/vehicles/ <i>vehicleName</i> / checkins          | 發佈      | Edge Agent 軟體會將車輛狀態資訊發佈到本主題中。<br><br>車輛狀態信息以協議緩衝器 ( protobuf ) 格式交換。如需詳細資訊，請參閱 <a href="#">AWS IoT FleetWise 軟體 Edge 代理程式開發人員指南</a> 。 |
| \$aws/iotfleetwise/vehicles/ <i>vehicleName</i> / signals           | 發佈      | Edge 代理程式軟體會將訊號發佈至此主題。<br><br>信號信息以協議緩衝器 ( 原生值 ) 格式進行交換。如需詳細資訊，請參閱 <a href="#">AWS IoT FleetWise 軟體 Edge 代理程式開發人員指南</a> 。             |
| \$aws/iotfleetwise/vehicles/ <i>vehicleName</i> /collection_schemes | 訂閱      | AWS IoT FleetWise 會針對此主題發佈資料收集方案。車輛消耗這些數據收集方案。  |
| \$aws/iotfleetwise/vehicles/ <i>vehicleName</i>                     | 訂閱      | AWS IoT FleetWise 發布解碼器清單到這一主題。車輛消耗這些解碼器清單。   |

| 主題                                 | 允許用戶端作業 | 描述 |  |
|------------------------------------|---------|----|--|
| <code>me /decoder_manifests</code> |         |    |  |

## 建立車輛

您可以使用 AWS IoT FleetWise 主控台或 API 建立車輛。

### Important

在開始之前，請檢查以下內容：

- 您必須擁有車輛型號，並且車輛型號的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理車輛模型](#)。
- 您的車輛型號必須與解碼器清單相關聯，並且解碼器清單的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。

### 主題

- [創建車輛 \(控制台\)](#)
- [建立車輛 \(AWS CLI\)](#)
- [創建多個車輛 \(AWS CLI\)](#)

## 創建車輛 (控制台)

您可以使用 AWS IoT FleetWise 主控台建立車輛。

### Important

在開始之前，請檢查以下內容：

- 您必須擁有車輛型號，並且車輛型號的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理車輛模型](#)。
- 您的車輛型號必須與解碼器清單相關聯，並且解碼器清單的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。



## 建立車輛

1. 開啟 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛]。
3. 在車輛摘要頁面上，選擇「建立車輛」，然後執行下列步驟。

### 主題

- [步驟 1：定義車輛屬性](#)
- [步驟 2：設定車輛憑證](#)
- [步驟 3：將原則附加至憑證](#)
- [步驟 4：檢閱和建立](#)

## 步驟 1：定義車輛屬性

在此步驟中，您可以命名車輛並將其與模型清單和解碼器清單相關聯。

1. 輸入車輛的唯一名稱。

### Important

車輛對應於一個 AWS IoT 東西。如果已有該名稱的物件存在，請選擇「將車輛與 IoT 物件建立關聯」來更新物件與車輛。或者，選擇不同的車輛名稱，AWS IoT 就 FleetWise 會自動為車輛建立新物件。

2. 從列表中選擇車輛型號（型號清單）。
3. 從列表中選擇一個解碼器清單。解碼器清單與車輛模型相關聯。
4. (選擇性) 若要關聯車輛屬性，請選擇「新增屬性」。如果跳過此步驟，您必須在建立車輛之後新增屬性，才能將其部署至行銷活動。
5. (選擇性) 若要將標籤與車輛產生關聯，請選擇「新增標籤」。您也可以在建建立車輛後新增標籤。
6. 選擇下一步。

## 步驟 2：設定車輛憑證

要將您的車輛用作 AWS IoT 物品，您必須配置帶有附加策略的車輛證書。如果跳過此步驟，您必須在建立車輛後設定憑證，才能將其部署至行銷活動。

1. 選擇「自動產生新憑證」(建議選項)。
2. 選擇下一步。

### 步驟 3：將原則附加至憑證

將原則附加至您在上一個步驟中設定的憑證。

1. 針對策略，輸入現有的策略名稱。若要建立新策略，請選擇 [建立策略]。
2. 選擇下一步。

### 步驟 4：檢閱和建立

確認車輛的設定，然後選擇 [建立車輛]。

#### Important

建立車輛後，您必須下載憑證和金鑰。您將使用憑證和私密金鑰來連接 AWS IoT FleetWise 軟體邊緣代理程式中的車輛。

## 建立車輛 (AWS CLI)

建立車輛時，您必須使用與解碼器清單相關聯的車輛型號。您可以使用 [CreateVehicle](#) API 操作來創建車輛。下列為使用 AWS CLI 的範例。

#### Important

在開始之前，請檢查以下內容：

- 您必須擁有車輛型號，並且車輛型號的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理車輛模型](#)。
- 您的車輛型號必須與解碼器清單相關聯，並且解碼器清單的狀態必須是ACTIVE。如需詳細資訊，請參閱 [創建和管理解碼器清單](#)。

若要建立車輛，請執行下列指令。

將文####包含車輛配置的 JSON 文件的名稱。

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

## Example 車輛配置

- (選擇性) `associationBehavior` 值可以是下列其中一項：
  - `CreateIotThing`— 創建車輛後，AWS IoT FleetWise 會自動為您的車輛創建帶有車輛 ID 名稱的 AWS IoT 東西。
  - `ValidateIotThingExists`— 使用現有的 AWS IoT 東西來創建車輛。

若要建立 AWS IoT 物件，請執行下列命令。將物件#####之物件的名稱。

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

如果未指定，AWS IoT FleetWise 會自動為您的車輛建立物 AWS IoT 件。

### Important

確保在車輛創建後配置了該 AWS IoT 物件。如需詳細資訊，請參閱 [提供車輛](#)。

- 以下列其中一項取代#####。
  - AWS IoT 物件的名稱 (若設定`associationBehavior`為 `ValidateIotThingExists`)。
  - 如果設定為 `ValidateIotThingExists`，則`associationBehavior`要建立的車輛 ID `CreateIotThing`。

車輛識別碼可以有 1-100 個字元。有效字元：a—z、A—Z、0—9、破折號 (-)、底線 (\_) 和冒號 (:)。

- 將#####-ARN 替換為車輛型號的 ARN ( 車型清單 )。
- 用與指定車##### ARN #####- ARN。
- ( 可選 ) 您可以添加其他屬性，以區分此車輛與從同一車輛型號創建的其他車輛。例如，如果您有一輛電動汽車，則可以為屬性指定以下值：{"fuelType": "electric"}

### Important

屬性必須先在關聯的車輛模型中定義，才能將其新增至個別車輛。

```
{
```

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

## 創建多個車輛 ( AWS CLI )

您可以使用 [BatchCreateVehicle](#) API 操作一次創建多輛車。下列為使用 AWS CLI 的範例。

若要建立多個車輛，請執行下列指令。

將檔案#####包含多輛車配置的 JSON 檔案名稱。

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

### Example 車輛配置

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

```
}
```

每個批次作業最多可建立 10 輛車輛。如需車輛配置的更多資訊，請參閱[建立車輛 \(AWS CLI\)](#)。

## 更新車輛 ( AWS CLI )

您可以使用 [UpdateVehicle](#) API 操作來更新現有車輛。下列為使用 AWS CLI 的範例。

若要更新車輛，請執行下列指令。

將檔案#####包含車輛設定的 JSON 檔案名稱。

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

### Example 車輛配置

- 將車###替換為您要更新的車輛的 ID。
- ( 可選 ) 將##-##-ARN 替換為您用來替換使用中車輛型號的車輛型號 ( 車型清單 ) 的 ARN。
- ( 可選 ) 將#####-ARN 替換為與您指定的新車輛型號相關聯的解碼器清單的 ARN。
- (選擇性) *attribute-update-mode* 以車輛屬性取代。
  - Merge— 透過使用新值更新現有屬性並新增屬性 (如果不存在)，將新屬性合併到現有屬性中。

例如，如果車輛具有下列屬性：`{"color": "black", "fuelType": "electric"}`，且您使用下列屬性更新車輛：`{"color": "", "fuelType": "gasoline", "model": "x"}`，則更新後的車輛具有下列屬性：`{"fuelType": "gasoline", "model": "x"}`

- Overwrite— 以新屬性取代現有屬性。

例如，如果一輛車具有以下屬性：`{"color": "black", "fuelType": "electric"}`，並且您使用該`{"model": "x"}`屬性更新車輛，則更新後的車輛具有該`{"model": "x"}`屬性。

如果輸入中存在屬性，則這是必需的。

- (選擇性) 若要新增屬性或使用新值更新現有屬性，請進行配置attributes。例如，如果您有一輛電動汽車，則可以為屬性指定以下值：`{"fuelType": "electric"}`

若要刪除屬性，attributeUpdateMode請將設定為Merge。

**⚠ Important**

屬性必須先在關聯的車輛模型中定義，才能將其新增至個別車輛。

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

## 更新多輛車 ( AWS CLI )

您可以使用 [BatchUpdateVehicle](#) API 操作一次更新多輛現有車輛。下列為使用 AWS CLI 的範例。

若要更新多輛車，請執行以下指令。

將檔案#####包含多輛車配置的 JSON 檔案名稱。

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

### Example 車輛配置

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

每批作業最多可更新 10 輛車輛。若要取得有關每輛車配置的更多資訊，請參閱 [〈〉 更新車輛 \(AWS CLI\)](#)。

## 刪除車輛

您可以使用 AWS IoT FleetWise 主控台或 API 刪除車輛。

### Important

車輛刪除後，AWS IoT FleetWise 會自動從相關的車隊和行銷活動中移除該車輛。如需詳細資訊，請參閱 [建立和管理車隊](#) 及 [透過行銷活動收集和傳輸資料](#)。但是，車輛仍然存在於物件或仍與中的物件相關聯 AWS IoT Core。如需刪除物件的相關指示，請參閱 AWS IoT Core 開發人員指南中的 [刪除物件](#)。

## 刪除車輛 (主機)

您可以使用 AWS IoT FleetWise 主控台刪除車輛。

### 刪除車輛

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [車輛]。
3. 在「車輛」頁面上，選擇您要刪除的車輛旁邊的按鈕。
4. 選擇刪除。
5. 在 [刪除] 中 **vehicle-name**，輸入車輛名稱，然後選擇 [刪除]。

## 刪除車輛 (AWS CLI)

您可以使用 [DeleteVehicle](#) API 操作刪除車輛。下面的例子使用 AWS CLI。

若要刪除車輛，請執行下列指令。

將車###替換為您要刪除的車輛的 ID。

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

## 取得車輛資訊 (AWS CLI)

您可以使用 [ListVehicles](#) API 操作來驗證車輛是否已被刪除。下列為使用 AWS CLI 的範例。

若要擷取所有車輛摘要的分頁清單，請執行以下指令。

```
aws iotfleetwise list-vehicles
```

您可以使用 [GetVehicle](#) API 操作來檢索車輛信息。下列為使用 AWS CLI 的範例。

要檢索車輛的元數據，請運行以下命令。

將車###替換為您要檢索的車輛的 ID。

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

### Note

此操作為**最終一致**。換句話說，車輛的更改可能不會立即反映出來。



# 建立和管理車隊

機群代表一組車輛。沒有關聯車輛的車隊是空實體。在您可以使用車隊同時管理多輛車輛之前，您必須先將車輛與車隊建立關聯。一輛車輛可屬於多個機群。您可以透過部署行銷活動，控制要從車隊收集哪些資料，以及何時收集資料。如需詳細資訊，請參閱[透過行銷活動收集和傳輸資料](#)。

機群群包含以下資訊。

fleetId

機群群的 ID。

(選用) description

可協助您找到艦隊的說明。

signalCatalogArn

訊號目錄的 Amazon Resource Name (ARN)。

AWS IoT FleetWise 提供下列 API 作業，您可以用來建立和管理叢集。

- [CreateFleet](#)— 創建一組包含相同信號組的車輛。
- [AssociateVehicleFleet](#)— 將車輛與車隊聯繫在一起。
- [DisassociateVehicleFleet](#)— 取消車輛與車隊的關聯。
- [UpdateFleet](#)— 更新現有機群的描述。
- [DeleteFleet](#)— 刪除現有的叢集。
- [ListFleets](#)— 擷取所有艦隊摘要的分頁清單。
- [ListFleetsForVehicle](#)— 檢索該車輛所屬的所有車隊 ID 的分頁列表。
- [ListVehiclesInFleet](#)— 檢索車隊中所有車輛摘要的分頁列表。
- [GetFleet](#)— 擷取有關叢集的資訊。

## 主題

- [建立叢集 \(AWS CLI\)](#)
- [將車輛與車隊建立關聯 \(AWS CLI\)](#)
- [取消車輛與車隊的關聯 \(AWS CLI\)](#)
- [更新艦隊 \(AWS CLI\)](#)

- [刪除機群群群 \(AWS CLI\)](#)
- [取得車隊資訊 \(AWS CLI\)](#)

## 建立叢集 (AWS CLI)

您可以使用 [CreateFleet](#) API 操作來創建車隊。以下範例使用 AWS CLI。

### Important

您必須有訊號目錄，才能建立機群群群。如需詳細資訊，請參閱[建立信號目錄 \(AWS CLI\)](#)。

若要建立機群群群，請執行以下命令。

- 將 **## ID** 取代為您正在建立的叢集的 ID。

叢集識別碼必須是唯一且具有 1-100 個字元。有效字元：字母 (A-Z)、數字 (0-9)、冒號 (:)、破折號 (-)、底線 (\_)、底線 ( )。

- (選擇性) 以 **##** 取代說明。

說明中的字元可以包含 1 至 2048 個字元。

- 取 **signal-catalog-arn** 代訊號目錄 ARN。

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

## 將車輛與車隊建立關聯 (AWS CLI)

您可以使用 [AssociateVehicleFleet](#) API 操作將車輛與車隊建立關聯。以下範例使用 AWS CLI。

### Important

- 您必須擁有一輛車輛和一支車隊，才能將車輛與車隊建立關聯。如需詳細資訊，請參閱[建立、佈建和管理車輛](#)。

- 如果您將車輛與行銷活動鎖定的車隊建立關聯，IoT FleetWise 會自動將行銷活動部署到車輛。

若要建立車輛與機群關聯，請執行以下命令。

- 用車## *ID* 替換車隊 ID。
- 用車#####。

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

## 取消車輛與車隊的關聯 (AWS CLI)

您可以使用 [DisassociateVehicleFleet](#) API 操作取消車輛與車隊的關聯。以下範例使用 AWS CLI。

若要取消車輛與機群的關聯，請執行以下命令。

- 用車## *ID* 替換車隊 ID。
- 用車#####。

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

## 更新艦隊 (AWS CLI)

您可以使用 [UpdateFleet](#) API 作業來更新叢集的描述。以下範例使用 AWS CLI。

若要更新機群群群群群，請執行以下命令。

- 將## *ID* 取代之為您要更新之叢集 ID。
- 以新##取代說明。

說明中的字元可以包含 1 至 2048 個字元。

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

## 刪除機群群群群 (AWS CLI)

您可以使用 [DeleteFleet](#) API 作業刪除叢集。以下範例使用 AWS CLI。

### Important

刪除車隊之前，請確定該車隊沒有相關聯的車輛。如需如何取消車輛與機群關聯的說明，請參閱「[取消車輛與車隊的關聯 \(AWS CLI\)](#)」。

若要刪除機群群群群群群群，請執行以下命令。

將 **## ID** 取代為您要刪除的叢集 ID。

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

## 取得車隊資訊 (AWS CLI)

您可以使用 [ListFleets](#) API 作業來驗證叢集是否已刪除。下列為使用 AWS CLI 的範例。

若要擷取所有機群的摘要清單，請執行以下命令。

```
aws iotfleetwise list-fleets
```

您可以使用 [ListFleetsForVehicle](#) API 操作來檢索該車輛所屬的所有車隊 ID 的分頁列表。下列為使用 AWS CLI 的範例。

若要擷取車輛所屬的所有機群的 ID 清單，請執行以下命令。

用車 **#####**。

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

您可以使用 [ListVehiclesInFleet](#) API 操作來擷取車隊中所有車輛摘要的分頁清單。下列為使用 AWS CLI 的範例。

若要擷取機群中所有車輛的摘要清單，請執行以下命令。

用車## *ID* 替換車隊 ID。


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

您可以使用 [GetFleetAPI](#) 作業擷取叢集資訊。下列為使用 AWS CLI 的範例。

若要擷取機群群的中繼資料，請執行以下命令。

用車## *ID* 替換車隊 ID。

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

 Note

此操作為最終一致。換言之，可能不會立即反映機群群的變更。

## 透過行銷活動收集和傳輸資料

促銷活動是資料收集規則的協調流程。行銷活動提供 AWS IoT Edge Agent FleetWise 軟體指示，說明如何選取、收集資料並將資料傳輸到雲端。

您可以在雲端中建立活動。在您或您的團隊核准行銷活動之後，AWS IoT FleetWise 會自動將其部署到車輛上。您可以選擇將戰役部署到車輛或車隊。在將執行中的行銷活動部署到車輛之前，Edge Agent 軟體不會開始收集資料。

### Note

在您擁有以下情況之前，廣告系列將無法運作。

- 邊緣代理程式軟體正在您的車輛中執行。如需有關如何開發、安裝和使用 Edge Agent 軟體的詳細資訊，請執行下列動作。
  1. 導覽至 [AWS IoT FleetWise 主控台](#)。
  2. 在服務首頁的 [開始使用 AWS IoT] FleetWise 區段中，選擇 [探索 Edge 代理程式]。
- 您已設定 AWS IoT Core 為配置您的車輛。如需詳細資訊，請參閱 [提供車輛](#)。

每個廣告系列都包含以下信息。

`signalCatalogArn`

與活動相關聯的信號目錄的 Amazon 資源名稱 (ARN)。

(選用) `tags`

標記是可用來管理促銷活動的中繼資料。您可以將相同標籤指派給來自不同服務的資源，以指出資源是相關的。

`TargetArn`

部署活動之車輛或機群的 ARN。

`name`

有助於識別促銷活動的唯一名稱。

## collectionScheme

資料收集配置會針對要收集的資料或何時收集資料提供 Edge Agent 軟體指示。AWS IoT FleetWise 目前支持基於條件的收集方案和基於時間的收集方案。

## conditionBasedCollectionScheme

以條件為基礎的收集配置使用邏輯運算式來識別要收集的資料。Edge 代理程式軟體會在符合條件時收集資料。

## expression

用來識別要收集哪些資料的邏輯運算式。例如，如果指定了 `$variable.`myVehicle.InVehicleTemperature` > 50.0` 運算式，Edge Agent 軟體會收集大於 50.0 的溫度值。如需如何撰寫表示式的指示，請參閱[促銷活動的邏輯運算](#)。

(選擇性) `triggerMode` 可以是下列其中一個值。

- `RISING_EDGE`— 只有在第一次符合條件時，Edge Agent 軟體才會收集資料。例如 `$variable.`myVehicle.AirBagDeployed` == true`。
- `ALWAYS`— Edge 代理程式軟體會在符合條件時收集資料。

## (選用) `minimumTriggerIntervalMs`

兩個資料收集事件之間的最短持續時間，以毫秒為單位。如果訊號經常變更，您可能會以較慢的速率收集資料。

## (選用) `conditionLanguageVersion`

條件運算式語言的版本。

## timeBasedCollectionScheme

當您定義以時間為基礎的收集配置時，請指定以毫秒為單位的時間週期。Edge 代理程式軟體會使用期間來決定收集資料的頻率。例如，如果時間週期為 120,000 毫秒，Edge Agent 軟體會每兩分鐘收集一次資料。

## (選用) `compression`

若要節省無線頻寬並減少網路流量，您可以指定 [SNAPPY](#) 來壓縮車輛中的資料。

依預設 (OFF)，Edge 代理程式軟體不會壓縮資料。

## dataDestinationConfigs

選擇廣告活動要傳送車輛資料的目的地。您可以選擇將資料存放在 Amazon S3 或 Amazon Timestream 中。

S3 是符合成本效益的資料儲存機制，可提供持久的資料管理功能和下游資料服務。您可以將 S3 用於與駕駛行為或分析長期維護相關的資料。

Timestream 是一種數據持久性機制，可以幫助您以近乎實時的方式識別趨勢和模式。您可以使用 Timestream 處理時間序列資料，例如分析車速或煞車的歷史趨勢。

#### (選用) dataExtraDimensions

您可以加入一個或多個屬性，以提供信號的其他資訊。

#### (選用) description

您可以新增說明以協助識別廣告活動的用途。

#### (選用) diagnosticsMode

將診斷模式設定為時SEND\_ACTIVE\_DTCS，促銷活動會傳送儲存的標準診斷故障代碼 (DTC)，以協助識別您的車輛出現問題。例如，P0097 表示引擎控制模組 (ECM) 已判定進氣溫度感測器 2 (IAT2) 輸入低於正常感測器範圍。

依預設 (OFF)，Edge 代理程式軟體不會傳送診斷代碼。

#### (選用) expiryTime

您可以定義廣告活動的到期日。促銷活動到期時，Edge Agent 軟體會停止收集此促銷活動中指定的資料。如果將多個行銷活動部署至車輛，Edge Agent 軟體會使用其他行銷活動來收集資料。

預設值：253402243200(一九九九九年十二月三十一日，世界標準時間 00:00:00)

#### (選用) postTriggerCollectionDuration

您可以定義觸發後收集持續時間，以便 Edge Agent 軟體在呼叫配置之後，持續收集指定期間的資料。例如，如果呼叫具有下列運算式的條件式收集配置：`$variable.`myVehicle.Engine.RPM` > 7000.0`，Edge Agent 軟體會繼續收集引擎的每分鐘轉數 (RPM) 值。即使 RPM 只高於 7000 一次，也可能表明存在機械問題。在此情況下，您可能希望 Edge Agent 軟體繼續收集資料，以協助監視狀況。

預設值：0

#### (選用) priority

您可以指定整數來表示促銷活動的優先順序層級。數量較小的促銷活動是較高的優先順序。如果您在車輛上部署多個行銷活動，則會先啟動優先順序較高的廣告活動。

預設值：0



## (選用) signalsToCollect

呼叫資料收集方案時，從中收集資料的訊號清單。

### Important

必須在此欄位中指定以條件為基礎的收集配置的運算式中使用的信號。

## name

呼叫資料收集方案時，從中收集資料的訊號名稱。

## (選用) maxSampleCount

呼叫資料收集配置時，Edge Agent 軟體收集並傳輸至雲端的資料範例數目上限。

## (選用) minimumSamplingIntervalMs

兩個資料樣本收集事件之間的最短持續時間，以毫秒為單位。如果訊號經常變更，您可以使用此參數以較慢的速率收集資料。

有效範圍：

## (選用) spoolingMode

如果設定spoolingMode為TO\_DISK，則當車輛未連線至雲端時，Edge Agent 軟體會暫時在本機儲存資料。重新建立連線後，儲存在本機的資料會自動傳輸到雲端。

預設值：OFF

## (選用) startTime

核准的行銷活動會在開始時間啟用。

預設值：0

廣告活動的狀態可以是下列其中一個值。

- CREATING— AWS IoT FleetWise 正在處理您的請求以創建廣告系列。
- WAITING\_FOR\_APPROVAL— 建立行銷活動後，它會進入WAITING\_FOR\_APPROVAL狀態。若要核准行銷活動，請使用 UpdateCampaign API 作業。行銷活動核准後，AWS IoT FleetWise 會自動將行銷活動部署到目標車輛或車隊。如需詳細資訊，請參閱[更新行銷活動 \(AWS CLI\)](#)。

- **RUNNING** — 該活動是活動。
- **SUSPENDED**— 活動已暫停。若要繼續宣傳活動，請使用 UpdateCampaign API 作業。

AWS IoT FleetWise 提供下列 API 作業，您可以用來建立和管理行銷活動。

- [CreateCampaign](#)-創建一個新的廣告系列。
- [UpdateCampaign](#)— 更新現有的行銷活動。建立宣傳活動後，您必須使用此 API 作業來核准行銷活動。
- [DeleteCampaign](#)— 刪除現有的行銷活動。
- [ListCampaigns](#)— 擷取所有行銷活動摘要的分頁清單。
- [GetCampaign](#)— 擷取有關行銷活動的資訊。

## 教學課程

- [建立行銷活動](#)
- [更新行銷活動 \(AWS CLI\)](#)
- [刪除廣告活動](#)
- [取得行銷活動資訊 \(AWS CLI\)](#)

## 建立行銷活動

您可以使用 AWS IoT 主 FleetWise 控制台或 API 建立行銷活動以收集車輛資料。

### Important

您必須具備以下條件，行銷活動才能正常運作：

- 邊緣代理程式軟體正在您的車輛中執行。如需有關如何開發、安裝和使用 Edge Agent 軟體的詳細資訊，請執行下列動作：
  1. 導覽至 [AWS IoT FleetWise 主控台](#)。
  2. 在服務首頁的 [開始使用 AWS IoT] FleetWise 區段中，選擇 [探索 Edge 代理程式]。
- 您已設定 AWS IoT Core 為配置您的車輛。如需詳細資訊，請參閱 [提供車輛](#)。

## 主題

- [建立行銷活動 \(主控台\)](#)
- [建立廣告活動 \(AWS CLI\)](#)
- [促銷活動的邏輯運算](#)

## 建立行銷活動 (主控台)

您可以使用 AWS IoT FleetWise 主控台建立行銷活動，以選取、收集車輛資料並將其傳輸到雲端。

### 建立行銷活動

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [廣告活動]。
3. 在 [行銷活動] 頁面上，選擇 [建立廣告活動]，然後完成下列主題中的步驟。

### 主題

- [步驟 1：設定行銷活動](#)
- [步驟 2：定義儲存目的地](#)
- [步驟 3：新增車輛](#)
- [步驟 4：檢閱和建立](#)
- [步驟 5：部署行銷活動](#)

#### Important

- 在建立廣告活動之前，您必須擁有信號目錄和車輛。如需詳細資訊，請參閱 [建立和管理信號目錄](#) 及 [建立、佈建和管理車輛](#)。
- 建立行銷活動之後，您必須核准行銷活動。如需詳細資訊，請參閱 [步驟 5：部署行銷活動](#)。

### 步驟 1：設定行銷活動

在 [一般資訊] 中，執行下列動作：

1. 輸入廣告活動的名稱。
2. (選用) 輸入描述。

設定促銷活動的資料收集配置。資料收集配置會針對要收集的資料或何時收集資料提供 Edge Agent 軟體指示。在 AWS IoT 主 FleetWise 控台中，您可以透過下列方式設定資料收集配置：

- 手動定義資料收集規則。
- 上載檔案以自動定義資料收集規則。

在「組態」選項中，選擇下列其中一項：

- 若要手動指定資料收集配置的類型並定義自訂配置的選項，請選擇「定義資料收集配置」。

手動指定資料收集規則的類型，並定義自訂配置的選項。

1. 在「資料收集方案詳細資料」區段中，選擇您要此行銷活動使用的資料收集配置類型。若要使用邏輯運算式辨識要收集的車輛資料，請選擇「條件式」。若要使用特定時段來決定收集車輛資料的頻率，請選擇「以時間為基礎」。
2. 定義促銷活動收集資料的持續時間。

#### Note

根據預設，核准的廣告活動會立即啟動，而且沒有設定的結束時間。為避免額外收費，您必須指定時間範圍。

3. 如果您指定了以條件為基礎的資料收集 Scheme，則必須定義邏輯運算式以識別要收集的資料。AWS IoT FleetWise 使用邏輯運算式來識別要為條件式架構收集哪些資料。運算式必須將信號的完整名稱指定為變數、比較運算子和比較值。

例如，如果您指定 `$variable.`myVehicle.InVehicleTemperature` > 50.0` 運算式，AWS IoT FleetWise 會收集大於 50.0 的溫度值。如需如何撰寫表示式的指示，請參閱 [促銷活動的邏輯運算](#)。


輸入用於識別要收集哪些資料的邏輯運算式。

4. (選擇性) 您可以指定條件運算式的語言版本。預設值為 1。
5. (選擇性) 您可以指定最小觸發間隔，也就是兩個資料收集事件之間的最短持續時間。例如，如果訊號經常變更，您可能想要以較慢的速率收集資料。
6. 指定 Edge Agent 軟體收集資料的觸發模式條件。依預設，AWS IoT FleetWise 軟體的 Edge 代理程式一律會在符合條件時收集資料。或者，它只能在第一次滿足條件時收集資料，開啟第一次觸發。

7. 如果您指定以時間為基礎的資料收集配置，則必須以毫秒為單位指定 10,000 至 60,000 毫秒的時間週期。Edge 代理程式軟體會使用期間來決定收集資料的頻率。
8. (選擇性) 您可以編輯配置的進階配置選項。
  - a. 若要透過壓縮資料節省無線頻寬並減少網路流量，請選擇 Snappy。
  - b. (選擇性) 若要定義在資料收集事件之後繼續收集資料的時間長度 (以毫秒為單位)，您可以指定 Post 觸發程序收集持續時間。
  - c. (選擇性) 若要指定廣告活動的優先順序層級，您可以指定廣告活動優先順序。優先順序較小的促銷活動會先部署，並視為具有較高的優先順序。
  - d. 當車輛未連線至雲端時，Edge Agent 軟體可以暫時在本機儲存資料。重新建立連線後，儲存在本機的資料會自動傳輸到雲端。指定是否要讓 Edge 代理程式在中斷連線期間將資料儲存在本機。
  - e. (選擇性) 若要提供訊號的其他資訊，請將最多五個屬性新增為「額外」資料維度。
- 若要上傳檔案以定義資料收集配置，請選取 [從本機裝置上傳 .json 檔案]。AWS IoT FleetWise 會自動定義您可以在檔案中定義的選項。您可以檢閱和更新選取的選項。

上傳包含資料收集配置詳細資訊的 .json 檔案。

1. 若要匯入有關資料收集規則的資訊，請選擇「選擇檔案」。如需所需檔案格式的詳細資訊，請參閱 [CreateCampaignAPI](#) 文件。


 Note

AWS IoT FleetWise 目前支援 .json 檔案格式副檔名。

2. AWS IoT FleetWise 會根據檔案中的資訊自動定義資料收集方案。檢閱 AWS IoT 為您 FleetWise 選取的選項。如有需要，您可以更新選項。

## 指定信號

您可以指定在呼叫資料收集配置時收集資料的訊號。

 Important

必須在此欄位中指定以條件為基礎的收集配置的運算式中使用的信號。

## 若要指定收集資料的訊號

1. 搜索信號的完全合格名稱。

### Note

信號的完整名稱是信號的路徑加上信號的名稱。使用點 (.) 來指向子信號。

例

如，`Vehicle.Chassis.SteeringWheel.HandsOff.HandsOffSteeringState` 是 HandsOff 致動器的完整名稱。`Vehicle.Chassis.SteeringWheel.HandsOff.` 是通往此致動器的路徑。

2. (選擇性) 針對最大樣本計數，輸入呼叫資料收集配置時 Edge Agent 軟體收集並傳輸至雲端的資料樣本數目上限。
3. (選擇性) 對於最小取樣間隔，請輸入兩個資料樣本收集事件之間的最短持續時間 (以毫秒為單位)。如果訊號經常變更，您可以使用此參數以較慢的速率收集資料。
4. 若要新增其他訊號，請選擇 [新增更多訊號]。您最多可以添加 999 個信號。
5. 選擇 下一步。

## 步驟 2：定義儲存目的地

### Note

只有在行銷活動包含視覺系統資料訊號時，才能將車輛資料傳輸到 Amazon S3。視覺系統資料正在預覽版中，可能會有所變更。

選擇您要儲存廣告活動所收集資料的目的地。您可以將車輛資料傳輸到 Amazon S3 或 Amazon Timestream。

在目的地設定中，執行下列動作：

- 從下拉式清單中選擇 S3 或時間流。

若要將車輛資料存放在 S3 儲存貯體中，請選擇 Amazon S3。S3 是一種物件儲存服務，可將資料作為物件存放在儲存貯體中。如需詳細資訊，請參閱 [Amazon 簡單儲存服務使用者指南中的建立、設定和使用 Amazon S3 儲存貯體](#)。

S3 可最佳化資料儲存成本，並提供使用車輛資料的其他機制，例如資料湖、集中式資料儲存、資料處理管道和分析。您可以使用 S3 存放資料以進行批次處理和分析。例如，您可以針對機器學習 (ML) 模型建立嚴格煞車事件的報告。傳入的車輛資料會在交付前緩衝 10 分鐘。

## Amazon S3

### Important

只有當 AWS IoT FleetWise 具有寫入 S3 儲存貯體的權限時，您才能將資料傳輸到 S3。如需授與存取權的詳細資訊，請參閱[使用 AWS IoT 控制存取權 FleetWise](#)。

在 S3 目的地設定中，執行下列動作：

1. 對於 S3 儲存貯體，請選擇 AWS IoT FleetWise 具有權限的儲存貯體。
2. (選擇性) 輸入可用於組織 S3 儲存貯體中存放的資料的自訂前置詞。
3. 選擇輸出格式，這是儲存在 S3 儲存貯體中的格式檔案。
4. 選擇是否要將存放在 S3 儲存貯體中的資料壓縮為 .gzip 檔案。建議您壓縮資料，因為這樣可將儲存成本降至最低。
5. 您在 S3 目標設定中選取的選項會變更範例 S3 物件 URI。這是 S3 中保存哪些文件的示例。

若要將車輛資料存放在「時間流」表格中，請選擇「Amazon Time stream」。您可以使用 Timestream 查詢車輛數據，以便識別趨勢和模式。例如，您可以使用 Timestream 來建立車輛燃油液位的警報。傳入的車輛資料會以近乎即時的速度傳輸至「時間流」。如需詳細資訊，請參閱[什麼是 Amazon Timestream](#)？在 Amazon Timestream 開發人員指南中。

## Amazon Timestream

### Important

僅當 AWS IoT FleetWise 具有將資料寫入時間流的權限時，您才能將資料傳輸到資料表。如需授與存取權的詳細資訊，請參閱[使用 AWS IoT 控制存取權 FleetWise](#)。

在「時間流」表格設定中，執行下列動作：

1. 對於「時間流」資料庫名稱，請從下拉式清單中選擇您的時間流資料庫名稱。
2. 對於「時間流」表格名稱，請從下拉式清單中選擇「時間串流」表格的名稱。

在時間串流的服務存取中，執行下列動作：

- 從下拉式清單中選擇 IAM 角色。
- 選擇 下一步。

### 步驟 3：新增車輛

若要選擇要部署行銷活動的車輛，請在車輛清單中選取它們。通過搜索您在創建車輛時添加的屬性和它們的值或按車輛名稱來過濾車輛。

在「篩選車輛」中，執行下列操作：

1. 在搜索框中，找到屬性或車輛名稱，然後從列表中選擇它。

#### Note

每個屬性只能使用一次。

2. 輸入要部署行銷活動的屬性值或車輛名稱。例如，如果屬性的完整名稱為fuelType，請輸入gasoline作為其值。
3. 若要搜尋其他車輛屬性，請重複上述步驟。您最多可以搜尋五種車輛屬性和不限數量的車輛名稱。
4. 符合搜尋條件的車輛會列在車輛名稱之下。選擇您想要部署行銷活動的車輛。

#### Note

搜尋結果中最多可顯示 100 輛車。選擇全選，將所有車輛新增至廣告活動。

5. 選擇 下一步。

### 步驟 4：檢閱和建立

確認廣告活動的設定，然後選擇 [建立廣告活動]。

#### Note

建立行銷活動後，您或您的團隊必須將戰役部署到車輛上。



## 步驟 5：部署行銷活動

建立行銷活動後，您或您的團隊必須將戰役部署到車輛上。

若要部署行銷活動

1. 在促銷活動摘要頁面上，選擇部署。
2. 檢閱並確認您要開始部署，並開始從連接到行銷活動的車輛收集資料。
3. 選擇部署。

如果您想要暫停從連接到廣告活動的車輛收集資料，請在「廣告活動摘要」頁面上選擇「暫停」。若要繼續從連線到行銷活動的車輛收集資料，請選擇 [繼續]。

## 建立廣告活動 (AWS CLI)

您可以使用 [CreateCampaign](#) API 作業建立行銷活動。下列為使用 AWS CLI 的範例。

建立行銷活動時，從車輛收集到的資料可以存放在 Amazon S3 (S3) 或 Amazon Timestream 中。選擇 Timestream 以取得快速、可擴充且無伺服器的时间序列資料庫，例如儲存需要接近即時處理的資料。選擇 S3 以獲得具有領先業界可擴展性、資料可用性、安全性和效能的物件儲存。

### Important

僅當 AWS IoT FleetWise 具有將資料寫入 S3 或時間流的權限時，您才能傳輸車輛資料。如需授與存取權的詳細資訊，請參閱[使用 AWS IoT 控制存取權 FleetWise](#)。

## 建立活動

### Important

- 在建立行銷活動之前，您必須擁有信號目錄和車輛或車隊。如需詳細資訊，請參閱 [建立和管理信號目錄](#)、[建立、佈建和管理車輛](#) 及 [建立和管理車隊](#)。
- 建立宣傳活動後，您必須使用 UpdateCampaign API 作業來核准行銷活動。如需更多資訊，請參閱 [更新行銷活動 \(AWS CLI\)](#)

若要建立行銷活動，請執行下列命令。

將檔#####包含行銷活動設定的 JSON 檔案名稱。

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

- 將廣告#####立的廣告活動名稱。
- *signal-catalog-arn*用信號目錄的 Amazon 資源名稱 ( ARN ) 替換。
- 以您建立##### ARN ###標 arn。
- 將儲存## *arn* 取代為 S3 儲存貯體的 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"
    }
  }
]
}

```

- 將廣告#####立的廣告活動名稱。
- *signal-catalog-arn*用信號目錄的 Amazon 資源名稱 ( ARN ) 替換。
- 以您建立##### ARN ###標 arn。
- 將 role *arn* 取代為任務執行角色的 ARN，該角色會授與 AWS IoT FleetWise 權限，以將資料傳遞至「時間串流」表格。
- 用時間流## ARN 替換表 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"
    }
  }
]
}

```

## 促銷活動的邏輯運算

AWS IoT FleetWise 使用邏輯表達式來識別哪些數據作為活動的一部分收集。如需運算式的詳細資訊，請參閱AWS IoT Events開發人員指南中的[運算式](#)。

運算式變數應該建構為符合所收集之資料類型的規則。對於遙測系統資料，運算式變數應該是訊號的完整名稱。對於視覺系統資料，運算式會結合訊號的完整名稱與從訊號資料類型到其中一個屬性的路徑。

例如，如果信號目錄包含以下節點：

```

{
  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"
}

```

如果節點遵循 ROS 2 定義：

```

{
  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運算式變數：

```

{
  ...
  $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`
}

```

## 更新行銷活動 (AWS CLI)

您可以使用 [UpdateCampaign](#) API 作業來更新現有的行銷活動。下面的命令使用 AWS CLI。

- 將廣告 **####** 取代為您要更新的廣告活動名稱。
- 以下列其中一項取代 **##** :
  - APPROVE— 批准活動以允許 AWS IoT FleetWise 將其部署到車輛或車隊。
  - SUSPEND— 暫停活動。行銷活動會從車輛中刪除，暫停的行銷活動中的所有車輛都將停止傳送資料。
  - RESUME— 重新激活 SUSPEND 活動。該戰役將重新部署到所有車輛，車輛將繼續發送數據。
  - UPDATE— 通過定義屬性並將其與信號相關聯來更新廣告系列。

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

## 刪除廣告活動

您可以使用 AWS IoT FleetWise 主控台或 API 刪除行銷活動。

### 刪除廣告活動 (主機)

若要刪除行銷活動，請使用 AWS IoT FleetWise 主控台。

若要刪除廣告活動

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在導覽窗格中，選擇 [廣告活動]。
3. 在 [行銷活動] 頁面上，選擇目標廣告活動。
4. 選擇 刪除。
5. 在刪除 **campaign-name** ? ，輸入要刪除的廣告活動名稱，然後選擇 [確認]。

### 刪除廣告活動 (AWS CLI)

您可以使用 [DeleteCampaign](#) API 操作刪除廣告活動。以下範例使用 AWS CLI。

若要刪除行銷活動，請執行下列命令。

將 ##### 取代為您要刪除的車輛名稱。

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

### 取得行銷活動資訊 (AWS CLI)

您可以使用 [ListCampaigns](#) API 操作來確認廣告活動是否已刪除。下列為使用 AWS CLI 的範例。

若要擷取所有行銷活動的摘要分頁清單，請執行下列命令。


```
aws iotfleetwise list-campaigns
```

您可以使用 [GetCampaign](#) API 操作來檢索車輛信息。下列為使用 AWS CLI 的範例。

若要擷取促銷活動的中繼資料，請執行下列命令。

將廣告####取代為您要擷取的廣告活動名稱。

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

 Note

此操作為最終一致。換句話說，廣告活動的更改可能不會立即反映出來。

## 處理和視覺化車輛資料

AWS IoT FleetWise 軟體的邊緣代理程式會將選取的車輛資料傳輸到 Amazon Timestream 或亞馬遜簡單儲存服務 (Amazon S3)。資料送達資料目標後，您可以使用其他AWS服務來視覺化和共用資料。

### 在時間流中處理車輛數據

Timestream 是一個完全受控的時間序列資料庫，每天可以儲存和分析數兆個時間序列資料點。您的資料會儲存在客戶管理的「時間流」表格中。您可以使用 Timestream 查詢車輛數據，以便獲得有關車輛的見解。如需詳細資訊，請參閱[什麼是 Amazon Timestream](#)？

傳輸至 Timestream 的資料預設結構描述包含下列欄位。

| 欄位名稱                   | 資料類型      | 描述                        |
|------------------------|-----------|---------------------------|
| eventId                | varchar   | 資料收集事件的識別碼。               |
| vehicleName            | varchar   | 從中收集資料的車輛 ID。             |
| name                   | varchar   | Edge 代理程式軟體用來收集資料的行銷活動名稱。 |
| time                   | timestamp | 資料點的時間戳記。                 |
| measure_name           | varchar   | 訊號的名稱。                    |
| measure_value::bigint  | bigint    | 整數類型的信號值。                 |
| measure_value::double  | double    | 雙倍類型的信號值。                 |
| measure_value::boolean | boolean   | 布爾類型的信號值。                 |



## 可視化儲存在時間流中的車輛資料

將車輛數據傳輸到 Timestream 後，您可以使用以下AWS服務來視覺化、監控、分析和共享您的數據。

- 使用 [Grafana 或 Amazon 受管的 Grafana](#)，以視覺化方式呈現和監控儀表板中的資料。您可以使用單一 Grafana 儀表板將來自多個AWS來源 (例如 Amazon CloudWatch 和時間流) 和其他資料來源的資料視覺化。
- 使用 [Amazon](#) 分析儀表板中的資料並將其視覺化 QuickSight。

## 在 S3 中處理車輛資料

Amazon S3 是一種物件儲存服務，可存放和保護任何數量的資料。您可以將 S3 用於各種使用案例，例如資料湖、備份和還原、存檔、企業應用程式、AWS IoT裝置和大數據分析。您的資料會以物件形式存放在 S3 儲存貯體中。如需詳細資訊，請參閱[什麼是 Amazon S3 ?](#)

傳輸到 Amazon S3 的預設資料結構描述包含下列欄位。

| 欄位名稱                 | 資料類型      | 描述                        |
|----------------------|-----------|---------------------------|
| eventId              | varchar   | 資料收集事件的識別碼。               |
| vehicleName          | varchar   | 從中收集資料的車輛 ID。             |
| name                 | varchar   | Edge 代理程式軟體用來收集資料的行銷活動名稱。 |
| time                 | timestamp | 資料點的時間戳記。                 |
| measure_name         | varchar   | 訊號的名稱。                    |
| measure_value_BIGINT | bigint    | 整數類型的信號值。                 |

| 欄位名稱                  | 資料類型    | 描述        |
|-----------------------|---------|-----------|
| measure_value_DOUBLE  | double  | 雙倍類型的信號值。 |
| measure_value_BOOLEAN | boolean | 布爾類型的信號值。 |
| measure_value_STRUCT  | struct  | 結構類型的信號值。 |

## S3 物件格式

AWS IoT 會將車輛資料 FleetWise 傳輸到 S3，並將其儲存為物件。您可以使用唯一識別資料的物件 URI，從促銷活動中尋找資料。S3 物件 URI 格式取決於收集的資料是非結構化資料還是已處理的資料。

### 非結構化資料

非結構化資料以非預先定義的方式存放在 S3 中。它可以是各種格式，例如圖像或視頻。

FleetWise 使用來自 Amazon Ion 檔案的訊號資料傳送至 AWS IoT 的車輛訊息會解碼並以物件的形式傳輸到 S3。S3 對象表示每個信號，並進行二進制編碼。

非結構化資料 S3 物件 URI 使用下列格式：

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

### 已處理資料

已處理的資料存放在 S3 中，並執行驗證、擴充和轉換訊息的處理步驟。物件清單和速度是處理資料的範例。

傳輸到 S3 的資料會以物件形式存放，這些物件代表緩衝約 10 分鐘的記錄。依預設，AWS IoT 會在 FleetWise 將物件寫入 S3 之前，以格式新增 UTC 時間 year=YYYY/month=MM/date=DD/hour=HH 前置詞。此前置詞會在值區中建立邏輯階層，其中每個正斜線 (/) 都會在階層中建立一個層級。處理的資料也包含非結構化資料的 S3 物件 URI。

處理過的資料 S3 物件 URI 使用下列格式：

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

## 原始資料

原始資料也稱為主要資料，是從 Amazon Ion 檔案收集的資料。您可以使用原始資料來解決任何問題或根本原因錯誤。

原始資料 S3 物件 URI 使用下列格式：

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

## 分析 S3 中存放的車輛資料

將車輛資料傳輸到 S3 後，您可以使用下列AWS服務來監控、分析和共用您的資料。

使用 Amazon 擷取和分析資料，以 SageMaker 進行下游標籤和機器學習 (ML) 工作流程。

如需詳細資訊，請參閱 Amazon SageMaker 開發人員指南中的下列主題：

- [處理資料](#)
- [訓練機器學習模型](#)
- [標籤圖片](#)

在 Amazon Athena 使用資料編目AWS Glue 編目程式並進行分析。根據預設，寫入 S3 的物件具有 Apache Hive 樣式的時間分區，其中包含以等號連接的索引鍵值對的資料路徑。

如需詳細資訊，請參閱 Amazon Athena 使用者指南中的下列主題：

- [在 Athena 分割資料](#)
- [用於連接AWS Glue到 Amazon S3 中的資料來源](#)
- [搭配使用 Athena 時的最佳做法 AWS Glue](#)

直接閱讀您的 Athena 表或 S3 儲存貯體，使用 Amazon QuickSight 將資料視覺化。

**i** Tip

如果您直接從 S3 閱讀，請確認您的車輛數據是 JSON 格式，因為 Amazon QuickSight 不支持 Apache 拼花格式。

如需詳細資訊，請參閱 Amazon QuickSight 使用者指南中的以下主題：

- [支援的資料來源](#)
- [建立資料來源](#)

# AWS CLI和AWS開發套件

本節提供製作相關資訊AWS物聯網 (IoT) FleetWise API 請求。如需有關的詳細資訊AWS物聯網 (IoT) FleetWise [作業和資料型別](#)，請參閱AWS物聯網 (IoT) FleetWise API 參考。

若為使用AWS物聯網 (IoT) FleetWise 使用各種程式語言，請使用[AWS開發套件](#)，其中包含下列自動功能：

- 加密簽署服務請求
- 重試請求
- 處理錯誤回應

對於命令行訪問，請使用AWS物聯網 (IoT) FleetWise 與[AWS CLI](#)。您可以控制AWS物聯網 (IoT) FleetWise，以及您的其他服務，從命令列，並透過指令碼將服務自動化。

# AWS IoT 疑難排解 FleetWise

使用本節中的疑難排解資訊和解決方案來協助解決 AWS IoT 的問題 FleetWise。

下列資訊可協助您疑難排解 AWS IoT 的常見問題 FleetWise。

## 主題

- [解碼器清單問題](#)
- [AWS IoT FleetWise 軟體問題的邊緣代理程式](#)

## 解碼器清單問題

解碼器清單問題疑難排解。

### 診斷解碼器清單 API 調用

| 錯誤  | 準則疑難排解   |
|---|--|
| UpdateOperationFailure.ConflictingDecoderUpdate   | 相同的解碼器資訊清單有多個更新要求。請稍候，然後再試一次。                          |
| UpdateOperationFailure.InternalFailure  | InternalFailure 作為封裝異常啟動。問題本身取決於封裝的異常。                 |
| UpdateOperationFailure.ActiveDecoderUpdate  | 解碼器清單處於狀Active態，無法更新。將解碼器資訊清單狀態變更為DRAFT，然後再試一次。        |
| UpdateOperationFailure.ConflictingModelUpdate   | AWS IoT FleetWise 正在嘗試驗證其他人正在修改的車輛模型（模型清單）。請稍候，然後再試一次。 |
| UpdateOperationFailure.ModelManifestValidationResponse : FailureReason.MODEL_DATA_ENTRIES_NOT_FOUND | 車輛型號沒有任何與之相關的信號。將信號添加到車輛模型中，並驗證信號是否可以在關聯的信號目錄中找到。      |

| 錯誤   | 準則疑難排解  |
|--|---|
| <code>UpdateOperationFailure.Mode<br/>lManifestValidationResponse :<br/>FailureReason.MODEL_NOT_ACTIVE</code>                    | 請更新車輛型號，使其ACTIVE處於狀態，然後再試一次。  |
| <code>UpdateOperationFailure.Mode<br/>lManifestValidationResponse :<br/>FailureReason.MODEL_NOT_FOUND</code>                     | AWSIoT FleetWise 無法找到與解碼器清單關聯的車輛型號。驗證車輛型號的 Amazon 資源名稱 (ARN)，然後再試一次。                      |
| <code>UpdateOperationFailure.Mode<br/>lManifestValidationResponse<br/>(FailureReason.MODEL_DATA_E<br/>NTRIES_READ_FAILURE</code> | 車輛模型驗證失敗，因為在信號目錄中找不到來自車型的信號名稱。驗證車輛模型中的信號全部包含在關聯的信號目錄中。                                    |
| <code>UpdateOperationFailure.Vali<br/>dationFailure</code>   | 在更新解碼器資訊清單的請求中找到無效的訊號或網路介面。驗證例外狀況傳回的所有訊號和網路介面是否存在，使用的所有訊號都與可用的介面相關聯，而且您不會移除具有與其相關聯之訊號的介面。 |
| <code>UpdateOperationFailure.KmsK<br/>eyAccessDenied</code>  | 用於作業的 AWS Key Management Service (AWS KMS) 金鑰存在權限問題。確認您所使用的角色可存取金鑰，然後再試一次。                |
| <code>UpdateOperationFailure.Deco<br/>derDoesNotExist</code>   | 解碼器清單不存在。請驗證解碼器資訊清單名稱，然後再試一次。   |

包含SIGNAL\_DECODER\_INCOMPATIBLE\_WITH\_SIGNAL\_CATALOG原因的 Vision 系統資料錯誤訊息會在回應中包含提示，提供要求失敗原因的相關資訊。您可以使用提示來決定要遵循的疑難排解準則。

#### Note

視覺系統資料正在預覽版中，可能會有所變更。

## 診斷解碼器清單視覺系統資料驗證

| 錯誤   | 準則疑難排解   |
|--|--|
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.NO_SIGNAL_IN_CATALOG_FOR_DECODER_SIGNAL)</pre>                   | <p>AWSIoT FleetWise 沒有使用信號目錄找到信號解碼器中使用的根信號結構。驗證結構的根信號是否在信號目錄中正確定義。</p>   |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_TYPE_INCOMPATIBLE_WITH_MESSAGE_SIGNAL_TYPE)</pre> | <p>在解碼器清單更新請求中，信號目錄中的原始消息未使用相同的數據類型定義。驗證請求中定義的原始消息是否與其對應的信號目錄定義匹配。</p>   |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.STRUCT_SIZE_MISMATCH)</pre>                                      | <p>信號目錄中結構中定義的屬性數量與您嘗試在解碼器清單中解碼的屬性數量不匹配。通過將信號與信號目錄中定義的信號進行比較，以確認您具有要解碼的信號數量是否正確。</p>   |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>           | <p>AWSIoT 在信號目錄中 FleetWise 找到了一個定義為 STRUCT 的信號，而沒有在解碼器清單請求中 structuredMessageDefinition 定義。確保每個結構都定義為解碼器清單更新請求 structuredMessageDefinition 中的 a。</p> |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>           | <p>解碼器清單中使用的結構的根信號未正確定義為信號目錄中的結構。解碼器清單中使用的根信號結構必須定義其字段 structFullyQualified Name。它還需要一個結構節點。 fullyQualifiedName</p>                                  |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>           | <p>解碼器清單請求中使用的葉消息之一未被定義為原始消息。確認要求中的所有分葉物件都定義為原始訊息。</p>   |
| <pre>InvalidSignalDecoder.withReason(SignalDecoderFailureReason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG)</pre>           | <p>信號目錄中的數組對像未定義為解碼器清單更新請求中的定義。 structuredMessageList確認所</p>   |



| 錯誤  | 準則疑難排解   |
|---|--|
| ason.SIGNAL_DECODER_INCOMPATIBLE_WITH_SIGNAL_CATALOG) | 有陣列屬性都定 structuredMessageList 義為解碼器資訊清單更新要求中的定義。 |

## AWS IoT FleetWise 軟體問題的邊緣代理程式

疑難排解 Edge 代理軟體問題。

### 問題

- [問題：Edge 代理程式軟體無法啟動。](#)
- [問題:\[錯誤\] \[IoT:FleetWiseEngine: 連接\]: \[無法初始化持久性庫\]](#)
- [問題：邊緣代理程式軟體不會收集內建診斷 \(OBD\) II PID 和診斷故障碼 \(DTC\)。](#)
- [問題：AWS IoT FleetWise 軟體的 Edge 代理程式無法從網路收集資料，或無法套用資料檢查規則。](#)
- [問題：\[錯誤\] \[:AwsIotConnectivityModule: 連接\]：\[連接失敗，錯誤\] 或 \[警告\] \[: 發送\]AwsIotChannel：\[沒有活動的 MQTT 連接。\]](#)

### 問題：Edge 代理程式軟體無法啟動。

Edge 代理程式軟體無法啟動時，您可能會看到下列錯誤。

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

解決方案：確定 AWS IoT FleetWise 軟體設定檔的邊緣代理程式使用有效的 JSON 格式。例如，確保已正確使用逗號。如需有關設定檔的詳細資訊，請執行下列動作，下載 AWS IoT 適用的 Edge Agent FleetWise 軟體開發人員指南。

1. 導覽至 [AWS IoT FleetWise 主控台](#)。
2. 在服務首頁的 [開始使用 AWS IoT] FleetWise 區段中，選擇 [探索 Edge 代理程式]。

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

解決方案：當 Edge Agent 軟體無法與組態檔中定義的網路介面建立通訊端通訊時，您可能會看到此錯誤。

若要檢查組態中定義的每個網路介面是否可用，請執行下列命令。

```
ip link show
```

若要使網路介面上線，請執行下列命令。*network-interface-id*以網路介面的 ID 取代。

```
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]
```

解決方案：當 Edge Agent 軟體無法建立 MQTT 連線時，您可能會看到此錯誤。AWS IoT Core 檢查下列項目是否正確設定，然後重新啟動 Edge Agent 軟體。

- `mqttnConnection::endpointUrl`— AWS 帳戶的 IoT 裝置端點。
- `mqttnConnection::clientId`— 執行 Edge Agent 軟體的車輛識別碼。
- `mqttnConnection::certificateFilename`— 車輛證書文件的路徑。
- `mqttnConnection::privateKeyFilename`— 車輛私鑰文件的路徑。
- 你曾經用 AWS IoT Core 來提供車輛。如需詳細資訊，請參閱[提供車輛](#)。

如需更多疑難排解資訊，請參閱[適用於 C++ 的 AWS IoT Device SDK 常見問題集](#)。

## 問題:[錯誤] [IoT:FleetWiseEngine: 連接]: [無法初始化持久性庫]

解決方案：當 Edge Agent 軟體找不到持續性儲存區時，您可能會看到此錯誤。檢查下列項目是否已正確設定，然後重新啟動 Edge Agent 軟體。

`persistency:persistencyPath`— 用於持續收集配置、解碼器資訊清單和資料快照集的本機路徑。

**問題：**邊緣代理程式軟體不會收集內建診斷 (OBD) II PID 和診斷故障碼 (DTC)。

**解決方案：**如

果 `obdInterface:pidRequestIntervalSeconds` 或 `obdInterface:dtcRequestIntervalSeconds` 設定為 0，您可能會看到此錯誤。

如果 Edge Agent 軟體在自動變速箱載具中執行，請確定 `obdInterface:hasTransmissionEcu` 已設定為 `true`。

如果您的車輛支援延伸控制器區域網路 (CAN 匯流排) 仲裁 ID，請確定 `obdInterface:useExtendedIds` 已設定為 `true`。

**問題：**AWS IoT FleetWise 軟體的 Edge 代理程式無法從網路收集資料，或無法套用資料檢查規則。

**解決方案：**違反預設配額時，您可能會看到這個錯誤。

| 資源             | 配額                  | 可調整 | 注意  |
|----------------|---------------------|-----|---|
| 訊號識別碼的值        | 訊號識別碼必須小於或等於 50,000 | 是   | 邊緣代理程式軟體不會從識別碼大於 50,000 的訊號收集資料。我們建議您在變更此配額之前，先檢查訊號目錄包含多少個訊號。 |
| 每輛車的有效數據收集計劃數目 | 256                 | 是   | 我們建議您在變更此配額之前，先檢查您在雲端中建立的廣告活動數量，以及每個廣告活動包含多少配置。               |
| 訊號歷史緩衝區的大小     | 20 MB               | 是   | 如果超出配額，Edge Agent 軟體會停止收集新資料。                                 |

問題：[錯誤] [:AwsIotConnectivityModule: 連接]：[連接失敗，錯誤] 或 [警告] [:: 發送]AwsIotChannel：[沒有活動的 MQTT 連接。]

解決方案：Edge Agent 軟體未連線至雲端時，您可能會看到此錯誤。根據預設，Edge Agent 軟體會 AWS IoT Core 每分鐘傳送偵測要求，並等待三分鐘。如果沒有回應，Edge Agent 軟體會自動重新建立與雲端的連線。

# AWS IoT 安全 FleetWise

雲安全 AWS 是最高的優先級。身為 AWS 客戶，您可以從資料中心和網路架構中獲益，該架構專為滿足對安全性最敏感的組織的需求而打造。

安全是 AWS 與您之間共同承擔的責任。[共同責任模型](#)將其描述為雲端的安全性和雲端中的安全性：

- 雲端的安全性 — AWS 負責保護在 AWS 雲端中執行 AWS 服務的基礎架構。AWS 還為您提供可以安全使用的服務。若要了解適用於 AWS IoT 的合規計劃 FleetWise，請參閱合規計劃的[AWS 服務範圍內的合規計](#)。
- 雲端中的安全性 — 您的責任取決於您使用的 AWS 服務。您也必須對其他因素負責，包括資料的機密性、您的公司的要求和適用法律和法規

本文件可協助您瞭解如何在使用 AWS IoT 時應用共同的責任模型 FleetWise。它說明如何設定 AWS IoT FleetWise 以符合安全性和合規性目標。您也會學到如何使用其他可 AWS 協助您監控和保護 AWS IoT FleetWise 資源的服務。

## 目錄

- [AWS IoT 中的資料保護 FleetWise](#)
- [控制存取 AWS IoT FleetWise](#)
- [AWS IoT 的 Identity and Access Management FleetWise](#)
- [AWS IoT 的合規性驗證 FleetWise](#)
- [AWS IoT 的韌性 FleetWise](#)
- [AWS IoT 中的基礎架構安全 FleetWise](#)
- [AWS IoT 中的組態和漏洞分析 FleetWise](#)
- [AWS IoT 的安全性最佳做法 FleetWise](#)

## AWS IoT 中的資料保護 FleetWise

AWS [共同責任模型](#)應用於 AWS IoT 中的數據保護 FleetWise。如此模型中所述，AWS 負責保護執行所有 AWS 雲端。您負責維護在此基礎設施上託管內容的控制權。您也同時負責所使用 AWS 服務的安全組態和管理任務。如需資料隱私權的詳細資訊，請參閱[資料隱私權常見問答集](#)。如需有關歐洲資料保護的相關資訊，請參閱 AWS 安全性部落格上的 [AWS 共同的責任模型和 GDPR](#) 部落格文章。

基於資料保護目的，我們建議您使用 AWS IAM Identity Center 或 AWS Identity and Access Management (IAM) 保護 AWS 帳戶 登入資料並設定個別使用者。如此一來，每個使用者都只會獲得授與完成其任務所必須的許可。我們也建議您採用下列方式保護資料：

- 每個帳戶均要使用多重要素驗證 (MFA)。
- 使用 SSL/TLS 與 AWS 資源進行通訊。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 使用設定 API 和使用者活動記錄 AWS CloudTrail。
- 使用 AWS 加密解決方案以及其中的所有默認安全控制 AWS 服務。
- 使用進階的受管安全服務 (例如 Amazon Macie)，協助探索和保護儲存在 Amazon S3 的敏感資料。
- 如果您在透過命令列介面或 API 存取時需要經 AWS 過 FIPS 140-2 驗證的加密模組，請使用 FIPS 端點。如需有關 FIPS 和 FIPS 端點的更多相關資訊，請參閱[聯邦資訊處理標準 \(FIPS\) 140-2 概觀](#)。

我們強烈建議您絕對不要將客戶的電子郵件地址等機密或敏感資訊，放在標籤或自由格式的文字欄位中，例如名稱欄位。這包括當您使用主控台、API FleetWise 或 AWS SDK AWS 服務 使用 AWS IoT 或其他物聯網時。AWS CLI您在標籤或自由格式文字欄位中輸入的任何資料都可能用於計費或診斷日誌。如果您提供外部伺服器的 URL，我們強烈建議請勿在驗證您對該伺服器請求的 URL 中包含憑證資訊。

AWS IoT FleetWise 旨在與您在支援的車輛硬體上開發並安裝的 Edge Agent 搭配使用，以便將車輛資料傳輸到 AWS 雲端。從車輛中提取數據可能會受到某些司法管轄區的數據隱私法規的約束。在使用 AWS IoT FleetWise 和安裝 Edge 代理程式之前，我們強烈建議您根據適用法律評估合規義務。這包括任何適用的法律要求，以提供合法適當的隱私權通知，並取得擷取車輛資料的任何必要同意。

## 靜態加密

從車輛收集的數據通過帶有 MQTT 消息協議的 AWS IoT Core 消息傳輸到雲中。AWS IoT 將資料 FleetWise 傳送到您的 Amazon Timestream 資料庫。在「時間流」中，您的數據已加密。默認情況下，所有靜態數據都會 AWS 服務 加密。

靜態加密與 AWS Key Management Service (AWS KMS) 整合以管理用於加密資料的加密金鑰。您可以選擇使用客戶受管金鑰來加密 AWS IoT 收集的資料 FleetWise。您可以透過建立、管理和檢視您的加密金鑰 AWS KMS。如需詳細資訊，請參閱[什麼是 AWS Key Management Service ?](#) 在 AWS Key Management Service 開發人員指南中。

## 傳輸中加密

與 AWS IoT 服務交換的所有資料都會在傳輸過程中使用傳輸層安全性 (TLS) 進行加密。如需詳細資訊，請參閱 AWS IoT 開發人員指南中的 [傳輸安全性](#)。

此外，還 AWS IoT Core 支援 [驗證](#) 和 [授權](#)，以協助安全地控制 AWS IoT 資 FleetWise 源的存取。車輛可以使用 X.509 憑證取得驗證 (登入) 以使用 AWS IoT，FleetWise 並使用 AWS IoT Core 原則取得授權 (具有權限) 來執行指定動作。如需詳細資訊，請參閱 [the section called “提供車輛”](#)。

## 資料加密

資料加密是指在傳輸中 (往返 AWS IoT FleetWise、閘道和伺服器之間傳輸時) 以及靜態 (儲存在本機裝置或中 AWS 服務) 時保護資料。您可以使用用戶端加密來保護靜態資料。

### Note

AWS IoT FleetWise 邊緣處理會公開託管在 AWS IoT FleetWise 閘道內的 API，並可透過本機網路存取。這些 API 會透過由 AWS IoT FleetWise Edge 連接器擁有的伺服器憑證所支援的 TLS 連線公開。對於用戶端驗證，這些 API 使用存取控制密碼。伺服器憑證私密金鑰和存取控制密碼都儲存在磁碟上。AWS IoT FleetWise 邊緣處理依賴檔案系統加密，以確保這些認證的靜態安全性。

如需伺服器端加密和用戶端加密的相關資訊，請檢閱下列主題。

### 目錄

- [靜態加密](#)
- [金鑰管理](#)

## 靜態加密

AWS IoT 會將您的資料 FleetWise 儲存在 AWS 雲端和閘道上。

### AWS 雲端中的靜態資料

AWS IoT AWS 服務 會將資料 FleetWise 儲存在預設情況下加密靜態資料的其他資料。靜態加密與 [AWS Key Management Service \(AWS KMS\)](#) 整合，用於管理用於加密 AWS IoT 中的資產屬性值和彙



總值的加密金鑰 FleetWise。您可以選擇使用客戶受管金鑰來加密 AWS IoT 中的資產屬性值和彙總值 FleetWise。您可以透過建立、管理和檢視您的加密金鑰 AWS KMS。

您可以選擇 AWS 擁有的金鑰 或客戶管理的金鑰來加密資料。

## 運作方式

靜態加密與 AWS KMS 整合，用於管理用於加密資料的加密金鑰。

- **AWS 擁有的金鑰** — 預設加密金鑰。AWS IoT FleetWise 擁有這個密鑰。您無法在 中檢視、管理或使用此金鑰 AWS 帳戶。您也無法在 AWS CloudTrail 記錄檔中看到金鑰的作業。您可以使用此鑰匙，無需額外付費。
- **客戶管理金鑰** — 金鑰會儲存在您建立、擁有及管理的帳戶中。您可以完全控制 KMS 金鑰。需 AWS KMS 支付額外費用。

## AWS 擁有的金鑰

AWS 擁有的金鑰 不會儲存在您的帳戶中。它們是 AWS 擁有和管理以供多個使用的 KMS 金鑰集合的一部分 AWS 帳戶。AWS 服務 可用 AWS 擁有的金鑰 來保護您的資料。

您無法檢視、管理或使用 AWS 擁有的金鑰，或稽核其使用情況。不過，您不需要採取任何動作或變更任何程式來保護資料加密的金鑰。

如果您使用，則不會向您收取費用 AWS 擁有的金鑰，且不會計入您帳戶的 AWS KMS 配額。

## 客戶受管金鑰

客戶受管金鑰為您在 帳戶中建立、擁有和管理的 KMS 金鑰。您可以完全控制這些 KMS 金鑰，例如：

- 建立和維護其關鍵政策、IAM 政策和撥款
- 啟用和禁用它們
- 旋轉他們的密碼材料
- 新增標籤
- 創建引用它們的別名
- 排定要刪除的程式

您也可以使用 CloudTrail 和 Amazon CloudWatch 日誌來追蹤 AWS IoT 代表您 FleetWise 傳送 AWS KMS 的請求。



如果您使用客戶受管金鑰，則必須授與 AWS IoT FleetWise 存取存放在帳戶中的 KMS 金鑰。AWS IoT FleetWise 使用信封加密和金鑰階層來加密資料。您的 AWS KMS 加密金鑰是用來加密此金鑰階層的金鑰。如需詳細資訊，請參閱《AWS Key Management Service 開發人員指南》中的[封套加密](#)。

下列範例原則授與 AWS IoT FleetWise 權限，以代表您建立客戶管理的金鑰。

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

#### Important

當您將新區段新增至 KMS 金鑰原則時，請勿變更原則中的任何現有區段。AWS 如果 IoT FleetWise 已啟用加密，FleetWise 且下列任一條件成立，AWS IoT 就無法對您的資料執行作業：

- KMS 金鑰已停用或刪除。
- KMS 金鑰原則未正確設定服務。

使用具有靜態加密功能的視覺系統資料

#### Note

視覺系統資料正在預覽版中，可能會有所變更。

如果您的 AWS IoT FleetWise 帳戶已啟用 AWS KMS 金鑰的客戶管理加密，並且想要使用視覺系統資料，請重設您的加密設定，使其與複雜的資料類型相容。這可讓 AWS IoT 建 FleetWise 立視覺系統資料所需的其他權限。

### Note

如果您尚未重設視覺系統資料的加密設定，您的解碼器資訊清單可能會卡在驗證狀態。

1. 使用 [GetEncryptionConfiguration](#) API 作業檢查是否已啟用 AWS KMS 加密。如果加密類型為 `FLEETWISE_DEFAULT_ENCRYPTION`，則不需要進一步處理。
2. 如果加密類型為 `KMS_BASED_ENCRYPTION`，請使用 [PutEncryptionConfiguration](#) API 作業將加密類型重設為 `FLEETWISE_DEFAULT_ENCRYPTION`。

```
{
  aws iotfleetwise put-encryption-configuration --encryption-type
  FLEETWISE_DEFAULT_ENCRYPTION
}
```

3. 使用 [PutEncryptionConfiguration](#) API 作業重新啟用的加密類型 `KMS_BASED_ENCRYPTION`

```
{
  aws iotfleetwise put-encryption-configuration \
  --encryption-type "KMS_BASED_ENCRYPTION"
  --kms-key-id kms_key_id
}
```

如需啟用加密的詳細資訊，請參閱[金鑰管理](#)。

## 金鑰管理

### AWS IoT FleetWise 雲端金鑰管理

依預設，AWS IoT FleetWise 會用 AWS 受管金鑰來保護您中的資料 AWS 雲端。您可以更新設定，以使用客戶受管金鑰來加密 AWS IoT 中的資料 FleetWise。您可以透過 AWS Key Management Service (AWS KMS) 建立、管理和檢視您的加密金鑰。

AWS IoT FleetWise 支援伺服器端加密，並將客戶管理金鑰儲存在其中，AWS KMS 以加密下列資源的資料。

| AWS IoT FleetWise 資源 | 資料類型                       | 使用客戶管理的金鑰靜態加密的欄位   |
|----------------------|----------------------------|--|
| 信號目錄                 |                            | description  |
|                      | 屬性                         | 說明、allowedValues、defaultValue、最大值  |
|                      | 傳動器                        | 說明，allowedValues，最小值，最大  |
|                      | 感測器                        | 說明，allowedValues，最小值，最大  |
| 車輛型號 ( 模型清單 )        |                            | description  |
| 解碼器清單                |                            | description  |
|                      | CanInterface               | 通訊協定名稱，通訊協定版本  |
|                      | ObdInterface               | requestMessageId, dtcRequestInterval 秒, hasTransmissionEcu, OBD 標準, 秒, pidRequestInterval useExtendedIds |
|                      | CanSignal                  | 係數、未簽署 isBigEndian、長度、messageId、偏移量、起始位元   |
|                      | ObdSignal                  | 依照長度、偏移量、PID、縮放、服務模式 pidResponseLength、起始位元組、bitMaskLength bitRightShift                                 |
| 車輛                   |                            | 屬性   |
| Campaign             |                            | description  |
|                      | conditionBasedCollection計劃 | 表達式, conditionLanguageVersion, minimumTriggerInterval 女士, 觸發模式   |
|                      | TimeBasedCollectionScheme  | 時期   |

**Note**

其他數據和資源使用由 AWS IoT 管理的密鑰使用默認加密進行加密 FleetWise。此金鑰會建立並儲存在 AWS IoT FleetWise 帳戶中。

如需詳細資訊，請參閱[什麼是 AWS Key Management Service ?](#) 在 AWS Key Management Service 開發人員指南中。

啟用使用 KMS 金鑰加密 (主控台)

若要將客戶受管金鑰與 AWS IoT 搭配使用 FleetWise，您必須更新 AWS IoT FleetWise 設定。

啟用使用 KMS 金鑰加密 (主控台)

1. 開啟 [AWS IoT FleetWise 主控台](#)。
2. 導覽至 [設定]。
3. 在加密中，選擇編輯以開啟編輯加密頁面。
4. 針對 [加密金鑰類型]，選擇 [選擇其他 AWS KMS 金鑰]。這可以使用存儲在中的客戶託管金鑰進行加密 AWS KMS。

**Note**

您只能針對 AWS IoT FleetWise 資源使用客戶受管金鑰加密。這包括信號目錄，車輛模型（模型清單），解碼器清單，車輛，車隊和戰役。

5. 使用下列其中一個選項選擇您的 KMS 金鑰：
  - 若要使用現有的 KMS 金鑰，請從清單中選擇您的 KMS 金鑰別名。
  - 若要建立新的 KMS 金鑰 — 選擇 [建立金 AWS KMS 鑰]。

**Note**

這將打開控 AWS KMS 制台。如需有關建立 KMS 金鑰的詳細資訊，請參閱 AWS Key Management Service 開發人員指南中的[建立金鑰](#)。

6. 選擇 [儲存] 以更新您的設定。

## 啟用使用 KMS 金鑰加密 (AWS CLI)

您可以使用 [PutEncryptionConfiguration](#) API 操作為您的 AWS IoT FleetWise 帳戶啟用加密。下面的例子使用 AWS CLI。

若要啟用加密，請執行下列命令。

- 將 **KMS #####** 取代為 KMS 金鑰的識別碼。

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

### Example response

```
{
  "kmsKeyId": "customer_kms_key_id",
  "encryptionStatus": "PENDING",
  "encryptionType": "KMS_BASED_ENCRYPTION"
}
```

## KMS 金鑰政策

建立 KMS 金鑰之後，您至少必須將下列陳述式新增至 KMS 金鑰原則，才能與 AWS IoT 搭配使用 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": "*"
}
```

```
}
```

如需有關編輯與 AWS IoT 搭配使用的 KMS 金鑰原則的詳細資訊 FleetWise，請參閱 AWS Key Management Service 開發人員指南中的 [變更金鑰原則](#)。

### Important

當您將新區段新增至 KMS 金鑰原則時，請勿變更原則中的任何現有區段。AWS 如果 IoT FleetWise 已啟用加密，FleetWise 且下列任一條件成立，AWS IoT 就無法對您的資料執行作業：

- KMS 金鑰已停用或刪除。
- KMS 金鑰原則未正確設定服務。

## 控制存取 AWS IoT FleetWise

以下各節說明如何控制 AWS IoT FleetWise 資源的存取權限。它們涵蓋的資訊包括如何授予應用程式存取權限，以便 AWS IoT FleetWise 可以在行銷活動期間傳輸車輛資料。他們也說明如何授與 Amazon S3 (S3) 儲 AWS IoT FleetWise 存貯體或 Amazon Timestream 資料庫和表格的存取權以存放資料。

管理所有這些存取形式的技術是 AWS Identity and Access Management (IAM)。如需 IAM 的相關資訊，請參閱 [什麼是 IAM ?](#)。

### 目錄

- [授 AWS IoT FleetWise 予對 Amazon S3 目的地的存取權](#)
- [授 AWS IoT FleetWise 予對 Amazon Timestream 目的地的存取權](#)

## 授 AWS IoT FleetWise 予對 Amazon S3 目的地的存取權

當您使用 Amazon S3 目的地時，會將車輛資料 AWS IoT FleetWise 交付到 S3 儲存貯體，並且可以選擇性地使用您擁有的 AWS KMS 金鑰進行資料加密。如果啟用了錯誤記錄，AWS IoT FleetWise 也會將資料傳遞錯誤傳送到您的記 CloudWatch 錄群組和串流。建立交付串流時，您必須具有 IAM 角色。

AWS IoT FleetWise 使用具有 S3 目標服務主體的儲存貯體政策。如需新增儲存貯體 [政策的詳細資訊](#)，請參閱 [Amazon 簡單儲存服務使用者指南中的使用 Amazon S3 主控台新增儲存貯體政策](#)。

使用下列存取政策啟用存 AWS IoT FleetWise 取 S3 儲存貯體。如果您未擁有 S3 儲存貯體，請新增 `s3:PutObjectAcl` 至 Amazon S3 動作清單。這會授與值區擁有者對由傳送之物件的完整存取權 AWS IoT FleetWise。如需有關如何保護值區中物件存取安全的詳細資訊，請參閱 Amazon 簡單儲存服務使用者指南中的儲存貯體政策範例。

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

下列儲存貯體政策適用於某個 AWS 區域中帳戶中的所有廣告活動。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:ListBucket"
      ],
      "Resource": "arn:aws:s3:::bucket-name"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "iotfleetwise.amazonaws.com"
        ]
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::bucket-name/*",
      "Condition": {
        "StringLike": {
          "aws:SourceArn": "arn:aws:iotfleetwise:region:account-id:campaign/*",
          "aws:SourceAccount": "account-id"
        }
      }
    }
  ]
}
```

如果您的 S3 儲存貯體已附加 KMS 金鑰，則該金鑰將需要下列政策。如需金鑰 [AWS Key Management Service](#) 鑰管理的相關資訊，請參閱 [Amazon 簡單儲存服務使用者指南](#) 中的 [使用伺服器端加密 \(SSE-KMS\) 保護資料](#)。



```
{
  "Version": "2012-10-17",
  "Effect": "Allow",
  "Principal": {
    "Service": "iotfleetwise.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey",
    "kms:Decrypt"
  ],
  "Resource": "key-arn"
}
```

### Important

建立儲存貯體時，S3 會建立預設存取控制清單 (ACL)，授與資源擁有者對資源的完整控制權。如果 AWS IoT FleetWise 無法將資料傳遞到 S3，請確定您停用 S3 儲存貯體上的 ACL。如需詳細資訊，請參閱 Amazon 簡單儲存體服務[使用者指南中的停用所有新值區的 ACL 和強制執行物件擁有權](#)。

## 授 AWS IoT FleetWise 予對 Amazon Timestream 目的地的存取權

當您使用「時間流」目的地時，會將車輛資料傳 AWS IoT FleetWise 送至「時間流」表格。您必須將政策附加到 IAM 角色，才能將資料傳送 AWS IoT FleetWise 至「時間串流」。

如果您使用主控台建[立行銷活動](#)，AWS IoT FleetWise 會自動將必要的原則附加至角色。

在開始之前，請檢查以下內容：

### Important

- 當您為 AWS IoT FleetWise 建立時間流資源時，您必須使用相同的 AWS 區域。如果您切換 AWS 區域，存取時間串流資源可能會發生問題。
- AWS IoT 可 FleetWise 在美國東部 (維吉尼亞北部) 和歐洲 (法蘭克福) 使用。
- [如需支援的區域清單，請參閱 AWS 一般參考](#)

- 您必須擁有時間流資料庫。如需教學課程，請參閱 Amazon Timestream 開發人員指南中的[建立資料庫](#)。
- 您必須在指定的時間流資料庫中建立資料表。如需教學課程，請參閱 Amazon Timestream 開發人員指南中的[建立表格](#)。

您可以使用 AWS CLI 建立具有時間流信任政策的 IAM 角色。若要建立 IAM 角色，請執行下列命令。

#### 使用信任政策建立 IAM 角色

- 以您正在建立的角色名稱取 *TimestreamExecutionRole* 代。
- 以包含 ##### 的 JSON 檔案取代信任原則。

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

建立權限原則以授與 AWS IoT FleetWise 權限，以便將資料寫入時間流。若要建立權限原則，請執行下列命令。

若要建立權限原則

- 以您正在建立的原則名稱取 `AWSIoT Fleetwise Access Timestream Permissions Policy` 代。
- 用包含 `####` 的 JSON 文件的名稱替換權限策略。

```
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": "*"   
    }  
  ]  
}
```

將許可政策附加到您的 IAM 角色

1. 從輸出中，複製許可政策的 Amazon 資源名稱 (ARN)。
2. 若要將 IAM 許可政策附加到您的 IAM 角色，請執行下列命令。

- 以您在上一個步驟中複製的 ARN 取 `permissions-policy-arn` 代。
- 以您建立的 IAM 角色名稱取 `TimestreamExecutionRole` 代。

```
aws iam attach-role-policy --policy-arn permissions-policy-arn --role-name TimestreamExecutionRole
```

如需詳細資訊，請參閱 [IAM 使用者指南中的 AWS 資源存取管理](#)。

## AWS IoT 的 Identity and Access Management FleetWise

AWS Identity and Access Management (IAM) 可協助管理員安全地控制 AWS 資源存取權。AWS 服務 IAM 管理員控制哪些人可以驗證 (登入) 和授權 (具有權限) 以使用 AWS IoT FleetWise 資源。IAM 是您可以使用的 AWS 服務，無需額外付費。

### 主題

- [物件](#)
- [使用身分驗證](#)
- [使用政策管理存取權](#)
- [AWS IoT 如何與 IAM FleetWise 搭配運作](#)
- [IoT 的身分識別原則範例 AWS FleetWise](#)
- [疑難排解 AWS IoT FleetWise 身分與存取](#)

### 物件

您使用 AWS Identity and Access Management (IAM) 的方式會因您在 AWS IoT 中所做的工作而有所不同 FleetWise。

服務使用者 — 如果您使用 AWS IoT FleetWise 服務執行工作，則管理員會為您提供所需的認證和權限。當您使用更多 AWS IoT FleetWise 功能來完成工作時，您可能需要額外的權限。了解存取許可的管理方式可協助您向管理員請求正確的許可。如果您無法存取 AWS IoT 中的功能 FleetWise，請參閱 [疑難排解 AWS IoT FleetWise 身分與存取](#)。

**服務管理員** — 如果您負責公司的 AWS IoT FleetWise 資源，您可能擁有完整的 AWS IoT 存取權 FleetWise。判斷服務使用者應存取哪些 AWS IoT FleetWise 功能和資源是您的工作。接著，您必須將請求提交給您的 IAM 管理員，來變更您服務使用者的許可。檢閱此頁面上的資訊，了解 IAM 的基本概念。若要進一步了解貴公司如何將 IAM 與 AWS IoT 搭配使用 FleetWise，請參閱[AWS IoT 如何與 IAM FleetWise 搭配運作](#)。

**IAM 管理員** — 如果您是 IAM 管理員，您可能想要瞭解如何撰寫政策來管理 AWS IoT 存取權的詳細資訊 FleetWise。若要檢視可在 IAM 中使用的 AWS IoT FleetWise 身分型政策範例，請參閱。[IoT 的身分識別原則範例 AWS FleetWise](#)

## 使用身分驗證

驗證是您 AWS 使用身分認證登入的方式。您必須以 IAM 使用者身分或假設 IAM 角色進行驗證 (登入 AWS)。AWS 帳戶根使用者

您可以使用透過 AWS 身分識別來源提供的認證，以聯合身分識別身分登入。AWS IAM Identity Center (IAM 身分中心) 使用者、貴公司的單一登入身分驗證，以及您的 Google 或 Facebook 登入資料都是聯合身分識別的範例。您以聯合身分登入時，您的管理員先前已設定使用 IAM 角色的聯合身分。當您使 AWS 用同盟存取時，您會間接擔任角色。

根據您的使用者類型，您可以登入 AWS Management Console 或 AWS 存取入口網站。如需有關登入的詳細資訊 AWS，請參閱《AWS 登入 使用指南》AWS 帳戶中的[如何登入](#)您的。

如果您 AWS 以程式設計方式存取，請 AWS 提供軟體開發套件 (SDK) 和命令列介面 (CLI)，以使用您的認證以加密方式簽署您的要求。如果您不使用 AWS 工具，則必須自行簽署要求。如需使用建議的方法自行簽署請求的詳細資訊，請參閱 IAM 使用者指南中的[簽署 AWS API 請求](#)。

無論您使用何種身分驗證方法，您可能都需要提供額外的安全性資訊。例如，AWS 建議您使用多重要素驗證 (MFA) 來增加帳戶的安全性。如需更多資訊，請參閱 AWS IAM Identity Center 使用者指南中的[多重要素驗證](#)和 IAM 使用者指南中的[在 AWS 中使用多重要素驗證 \(MFA\)](#)。

## AWS 帳戶 根使用者

當您建立時 AWS 帳戶，您會從一個登入身分開始，該身分可完整存取該帳戶中的所有資源 AWS 服務和資源。此身分稱為 AWS 帳戶 root 使用者，可透過使用您用來建立帳戶的電子郵件地址和密碼登入來存取。強烈建議您不要以根使用者處理日常任務。保護您的根使用者憑證，並將其用來執行只能由根使用者執行的任務。如需這些任務的完整清單，了解需以根使用者登入的任務，請參閱 IAM 使用者指南中的[需要根使用者憑證的任務](#)。

## 聯合身分

最佳作法是要求人類使用者 (包括需要系統管理員存取權的使用者) 使用與身分識別提供者的同盟，才能使用臨時登入資料進行存取 AWS 服務。

聯合身分識別是來自企業使用者目錄的使用者、Web 身分識別提供者、Identity Center 目錄，或使用透過身分識別來源提供的認證進行存取 AWS 服務的任何使用者。AWS Directory Service 同盟身分存取時 AWS 帳戶，他們會假設角色，而角色則提供臨時認證。

對於集中式存取權管理，我們建議您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中建立使用者和群組，也可以連線並同步到自己身分識別來源中的一組使用者和群組，以便在所有應用程式 AWS 帳戶 和應用程式中使用。如需 IAM Identity Center 的相關資訊，請參閱 [AWS IAM Identity Center 使用者指南中的什麼是 IAM Identity Center ?](#)。

## IAM 使用者和群組

[IAM 使用者](#)是您內部的身分，具 AWS 帳戶 有單一人員或應用程式的特定許可。建議您盡可能依賴暫時憑證，而不是擁有建立長期憑證 (例如密碼和存取金鑰) 的 IAM 使用者。但是如果特定使用案例需要擁有長期憑證的 IAM 使用者，建議您輪換存取金鑰。如需更多資訊，請參閱 [IAM 使用者指南](#)中的為需要長期憑證的使用案例定期輪換存取金鑰。

[IAM 群組](#)是一種指定 IAM 使用者集合的身分。您無法以群組身分簽署。您可以使用群組來一次為多名使用者指定許可。群組可讓管理大量使用者許可的程序變得更為容易。例如，您可以擁有一個名為 IAMAdmins 的群組，並給予該群組管理 IAM 資源的許可。

使用者與角色不同。使用者只會與單一人員或應用程式建立關聯，但角色的目的是在由任何需要它的人員取得。使用者擁有永久的長期憑證，但角色僅提供暫時憑證。如需進一步了解，請參閱 IAM 使用者指南中的[建立 IAM 使用者 \(而非角色\) 的時機](#)。

## IAM 角色

[IAM 角色](#)是您 AWS 帳戶 內部具有特定許可的身分。它類似 IAM 使用者，但不與特定的人員相關聯。您可以[切換角色，在中暫時擔任 IAM 角色](#)。AWS Management Console 您可以透過呼叫 AWS CLI 或 AWS API 作業或使用自訂 URL 來擔任角色。如需使用角色的方法更多相關資訊，請參閱 IAM 使用者指南中的[使用 IAM 角色](#)。

使用暫時憑證的 IAM 角色在下列情況中非常有用：

- 聯合身分使用者存取 – 若要向聯合身分指派許可，請建立角色，並為角色定義許可。當聯合身分進行身分驗證時，該身分會與角色建立關聯，並獲授予由角色定義的許可。如需有關聯合角色的相關資訊，請參閱 [IAM 使用者指南](#)中的為第三方身分提供者建立角色。如果您使用 IAM Identity Center，

則需要設定許可集。為控制身分驗證後可以存取的內容，IAM Identity Center 將許可集與 IAM 中的角色相關聯。如需有關許可集的資訊，請參閱 AWS IAM Identity Center 使用者指南中的[許可集](#)。

- 暫時 IAM 使用者許可 – IAM 使用者或角色可以擔任 IAM 角色來暫時針對特定任務採用不同的許可。
- 跨帳戶存取權 – 您可以使用 IAM 角色，允許不同帳戶中的某人 (信任的委託人) 存取您帳戶中的資源。角色是授予跨帳戶存取權的主要方式。但是，對於某些策略 AWS 服務，您可以將策略直接附加到資源 (而不是使用角色作為代理)。若要了解跨帳戶存取權角色和資源型政策間的差異，請參閱 IAM 使用者指南中的[IAM 角色與資源類型政策的差異](#)。
- 跨服務訪問 — 有些 AWS 服務 使用其他 AWS 服務功能。例如，當您在服務中進行呼叫時，該服務通常會在 Amazon EC2 中執行應用程式或將物件儲存在 Amazon Simple Storage Service (Amazon S3) 中。服務可能會使用呼叫主體的許可、使用服務角色或使用服務連結角色來執行此作業。
  - 轉寄存取工作階段 (FAS) — 當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱[《轉發存取工作階段》](#)。
  - 服務角色 – 服務角色是服務擔任的 [IAM 角色](#)，可代表您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的[建立角色以委派許可給 AWS 服務](#)。
  - 服務連結角色 — 服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。
- 在 Amazon EC2 上執行的應用程式 — 您可以使用 IAM 角色來管理在 EC2 執行個體上執行的應用程式以及發出 AWS CLI 或 AWS API 請求的臨時登入資料。這是在 EC2 執行個體內儲存存取金鑰的較好方式。若要將 AWS 角色指派給 EC2 執行個體並提供給其所有應用程式，請建立連接至執行個體的執行個體設定檔。執行個體設定檔包含該角色，並且可讓 EC2 執行個體上執行的程式取得暫時憑證。如需更多資訊，請參閱 IAM 使用者指南中的[利用 IAM 角色來授予許可給 Amazon EC2 執行個體上執行的應用程式](#)。

若要了解是否要使用 IAM 角色或 IAM 使用者，請參閱 IAM 使用者指南中的[建立 IAM 角色 \(而非使用者\) 的時機](#)。

## 使用政策管理存取權

您可以透 AWS 過建立原則並將其附加至 AWS 身分識別或資源來控制中的存取。原則是一個物件 AWS，當與身分識別或資源相關聯時，會定義其權限。AWS 當主參與者 (使用者、root 使用者或角色



工作階段) 提出要求時，評估這些原則。政策中的許可決定是否允許或拒絕請求。大多數原則會以 JSON 文件的形式儲存在中。如需 JSON 政策文件結構和內容的更多相關資訊，請參閱 IAM 使用者指南中的 [JSON 政策概觀](#)。

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

預設情況下，使用者和角色沒有許可。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

IAM 政策定義該動作的許可，無論您使用何種方法來執行操作。例如，假設您有一個允許 `iam:GetRole` 動作的政策。具有該原則的使用者可以從 AWS Management Console、AWS CLI、或 AWS API 取得角色資訊。

## 身分型政策

身分型政策是可以附加到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱 IAM 使用者指南中的 [建立 IAM 政策](#)。

身分型政策可進一步分類成內嵌政策或受管政策。內嵌政策會直接內嵌到單一使用者、群組或角色。受管理的策略是獨立策略，您可以將其附加到您的 AWS 帳戶。受管政策包括 AWS 受管政策和客戶管理的策略。若要了解如何在受管政策及內嵌政策間選擇，請參閱 IAM 使用者指南中的 [在受管政策和內嵌政策間選擇](#)。

## 資源型政策

資源型政策是連接到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

資源型政策是位於該服務中的內嵌政策。您無法在以資源為基礎的政策中使用 IAM 的 AWS 受管政策。

## 存取控制清單 (ACL)

存取控制清單 (ACL) 可控制哪些委託人 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。



Amazon S3 和 Amazon VPC 是支援 ACL 的服務範例。AWS WAF若要進一步了解 ACL，請參閱 Amazon Simple Storage Service 開發人員指南中的[存取控制清單 \(ACL\) 概觀](#)。

## 其他政策類型

AWS 支援其他較不常見的原則類型。這些政策類型可設定較常見政策類型授予您的最大許可。

- 許可界限 – 許可範圍是一種進階功能，可供您設定身分型政策能授予 IAM 實體 (IAM 使用者或角色) 的最大許可。您可以為實體設定許可界限。所產生的許可會是實體的身分型政策和其許可界限的交集。會在 Principal 欄位中指定使用者或角色的資源型政策則不會受到許可界限限制。所有這類政策中的明確拒絕都會覆寫該允許。如需許可範圍的更多相關資訊，請參閱 IAM 使用者指南中的[IAM 實體許可範圍](#)。
- 服務控制策略 (SCP) — SCP 是 JSON 策略，用於指定中組織或組織單位 (OU) 的最大權限。AWS Organizations AWS Organizations 是一種用於分組和集中管理您企業擁 AWS 帳戶 有的多個服務。若您啟用組織中的所有功能，您可以將服務控制政策 (SCP) 套用到任何或所有帳戶。SCP 限制成員帳戶中實體的權限，包括每個 AWS 帳戶根使用者帳戶。如需組織和 SCP 的更多相關資訊，請參閱 AWS Organizations 使用者指南中的[SCP 運作方式](#)。
- 工作階段政策 – 工作階段政策是一種進階政策，您可以在透過編寫程式的方式建立角色或聯合使用者的暫時工作階段時，作為參數傳遞。所產生工作階段的許可會是使用者或角色的身分型政策和工作階段政策的交集。許可也可以來自資源型政策。所有這類政策中的明確拒絕都會覆寫該允許。如需更多資訊，請參閱 IAM 使用者指南中的[工作階段政策](#)。

## 多種政策類型

將多種政策類型套用到請求時，其結果形成的許可會更為複雜、更加難以理解。要了解如何在涉及多個政策類型時 AWS 確定是否允許請求，請參閱《IAM 使用者指南》中的[政策評估邏輯](#)。

## AWS IoT 如何與 IAM FleetWise 搭配運作

在您使用 IAM 管理 AWS IoT 存取之前 FleetWise，請先了解哪些 IAM 功能可用於 AWS IoT FleetWise。

可搭配 AWS IoT 使用的 IAM 功能 FleetWise

| IAM 功能                | AWS IoT FleetWise 支援 |
|-----------------------|----------------------|
| <a href="#">身分型政策</a> | 是                    |

| IAM 功能                       | AWS IoT FleetWise 支援 |
|------------------------------|----------------------|
| <a href="#">資源型政策</a>        | 否                    |
| <a href="#">政策動作</a>         | 是                    |
| <a href="#">政策資源</a>         | 是                    |
| <a href="#">政策條件索引鍵</a>      | 是                    |
| <a href="#">ACL</a>          | 否                    |
| <a href="#">ABAC(政策中的標籤)</a> | 部分                   |
| <a href="#">臨時憑證</a>         | 是                    |
| <a href="#">主體許可</a>         | 是                    |
| <a href="#">服務角色</a>         | 否                    |
| <a href="#">服務連結角色</a>       | 否                    |

若要深入瞭解 AWS IoT FleetWise 和其他 AWS 服務如何搭配大多數 IAM 功能運作，請參閱 IAM 使用者指南中的搭配 IAM 使用的[AWS 服務](#)。

## IoT 的身分識別型原則 AWS FleetWise

|         |   |
|---------|---|
| 支援身分型政策 | 是 |
|---------|---|

身分型政策是可以連接到身分 (例如 IAM 使用者、使用者群組或角色) 的 JSON 許可政策文件。這些政策可控制身分在何種條件下能對哪些資源執行哪些動作。若要了解如何建立身分類型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

使用 IAM 身分型政策，您可以指定允許或拒絕的動作和資源，以及在何種條件下允許或拒絕動作。您無法在身分型政策中指定主體，因為這會套用至連接的使用者或角色。如要了解您在 JSON 政策中使用的所有元素，請參閱《IAM 使用者指南》中的[IAM JSON 政策元素參考](#)。

## IoT 的身分識別原則範例 AWS FleetWise

若要檢視 AWS IoT FleetWise 身分型原則的範例，請參閱 [IoT 的身分識別原則範例 AWS FleetWise](#)

## IoT 中 AWS 以資源為基礎的政策 FleetWise

支援以資源基礎的政策 否

資源型政策是附加到資源的 JSON 政策文件。資源型政策的最常見範例是 IAM 角色信任政策和 Amazon S3 儲存貯體政策。在支援資源型政策的服務中，服務管理員可以使用它們來控制對特定資源的存取權限。對於附加政策的資源，政策會定義指定的主體可以對該資源執行的動作以及在何種條件下執行的動作。您必須在資源型政策中 [指定主體](#)。主參與者可以包括帳戶、使用者、角色、同盟使用者或。AWS 服務

若要啟用跨帳戶存取，您可以指定在其他帳戶內的所有帳戶或 IAM 實體，作為資源型政策的主體。新增跨帳戶主體至資源型政策，只是建立信任關係的一半。當主體和資源位於不同時 AWS 帳戶，受信任帳戶中的 IAM 管理員也必須授與主體實體 (使用者或角色) 權限，才能存取資源。其透過將身分型政策連接到實體來授與許可。不過，如果資源型政策會為相同帳戶中的主體授予存取，這時就不需要額外的身分型政策。如需詳細資訊，請參閱《IAM 使用者指南》中的 [IAM 角色與資源型政策有何差異](#)。

## AWS IoT 的政策動作 FleetWise

支援政策動作 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

JSON 政策的 Action 元素描述您可以用來允許或拒絕政策中存取的動作。原則動作通常與關聯的 AWS API 作業具有相同的名稱。有一些例外狀況，例如沒有相符的 API 操作的僅限許可動作。也有一些作業需要政策中的多個動作。這些額外的動作稱為相依動作。

政策會使用動作來授予執行相關聯動作的許可。

若要查看 AWS IoT FleetWise 動作清單，請參閱服務授權參考資料 FleetWise 中 [由 AWS IoT 定義的動作](#)。

AWS IoT 中的原則動作會在動作之前 FleetWise 使用下列前置詞：

```
iotfleetwise
```

如需在單一陳述式中指定多個動作，請用逗號分隔。

```
"Action": [  
  "iotfleetwise:action1",  
  "iotfleetwise:action2"  
]
```

您也可以使用萬用字元 (\*) 來指定多個動作。例如，若要指定開頭是 List 文字的所有動作，請包含以下動作：

```
"Action": "iotfleetwise:List*"
```

若要檢視 AWS IoT FleetWise 身分型原則的範例，請參閱 [IoT 的身分識別原則範例 AWS FleetWise](#)

## AWS IoT 的政策資源 FleetWise

支援政策資源

是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Resource JSON 政策元素可指定要套用動作的物件。陳述式必須包含 Resource 或 NotResource 元素。最佳實務是使用其 [Amazon Resource Name \(ARN\)](#) 來指定資源。您可以針對支援特定資源類型的動作 (稱為資源層級許可) 來這麼做。

對於不支援資源層級許可的動作 (例如列出操作)，請使用萬用字元 (\*) 來表示陳述式適用於所有資源。

```
"Resource": "*"
```

若要查看 AWS IoT FleetWise 資源類型及其 ARN 的清單，請參閱服務授權參考資料 [FleetWise 中由 AWS IoT 定義](#) 的資源。若要瞭解可以使用哪些動作指定每個資源的 ARN，請參閱 [AWS IoT FleetWise 定義的動作](#)。

若要檢視 AWS IoT FleetWise 身分型原則的範例，請參閱 [IoT 的身分識別原則範例 AWS FleetWise](#)

## AWS IoT 的原則條件金鑰 FleetWise

支援服務特定政策條件金鑰 是

管理員可以使用 AWS JSON 政策來指定誰可以存取哪些內容。也就是說，哪個主體在什麼條件下可以對什麼資源執行哪些動作。

Condition 元素 (或 Condition 區塊) 可讓您指定使陳述式生效的條件。Condition 元素是選用項目。您可以建立使用[條件運算子](#)的條件運算式 (例如等於或小於)，來比對政策中的條件和請求中的值。

若您在陳述式中指定多個 Condition 元素，或是在單一 Condition 元素中指定多個索引鍵，AWS 會使用邏輯 AND 操作評估他們。如果您為單一條件索引鍵指定多個值，請使用邏輯 OR 運算來 AWS 評估條件。必須符合所有條件，才會授與陳述式的許可。

您也可以在指定條件時使用預留位置變數。例如，您可以只在使用者使用其 IAM 使用者名稱標記時，將存取資源的許可授予該 IAM 使用者。如需更多資訊，請參閱 IAM 使用者指南中的[IAM 政策元素：變數和標籤](#)。

AWS 支援全域條件金鑰和服務特定條件金鑰。若要查看所有 AWS 全域條件金鑰，請參閱《IAM 使用者指南》中的[AWS 全域條件內容金鑰](#)。

若要查看 AWS IoT FleetWise 條件金鑰清單，請參閱服務授權參考資料 FleetWise 中的[AWS IoT 條件金鑰](#)。若要瞭解您可以使用條件金鑰的動作和資源，請參閱[AWS IoT 定義的動作 FleetWise](#)。

若要檢視 AWS IoT FleetWise 身分型原則的範例，請參閱。[IoT 的身分識別原則範例 AWS FleetWise](#)

## IoT 中 AWS 的存取控制清單 (ACL) FleetWise

支援 ACL 否

存取控制清單 (ACL) 可控制哪些主體 (帳戶成員、使用者或角色) 擁有存取某資源的許可。ACL 類似於資源型政策，但它們不使用 JSON 政策文件格式。

## 基於屬性的訪問控制 ( ABAC ) 與 IoT AWS FleetWise

支援 ABAC (政策中的標籤) 部分

屬性型存取控制 (ABAC) 是一種授權策略，可根據屬性來定義許可。在中 AWS，這些屬性稱為標籤。您可以將標籤附加到 IAM 實體 (使用者或角色) 和許多 AWS 資源。為實體和資源加上標籤是 ABAC 的第一步。您接著要設計 ABAC 政策，允許在主體的標籤與其嘗試存取的資源標籤相符時操作。

ABAC 在成長快速的環境中相當有幫助，並能在政策管理變得繁瑣時提供協助。

若要根據標籤控制存取，請使用 `aws:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 條件金鑰，在政策的 [條件元素](#) 中，提供標籤資訊。

如果服務支援每個資源類型的全部三個條件金鑰，則對該服務而言，值為 Yes。如果服務僅支援某些資源類型的全部三個條件金鑰，則值為 Partial。

如需 ABAC 的詳細資訊，請參閱《IAM 使用者指南》中的 [什麼是 ABAC?](#)。如要查看含有設定 ABAC 步驟的教學課程，請參閱《IAM 使用者指南》中的 [使用屬性型存取控制 \(ABAC\)](#)。

#### Note

AWS IoT FleetWise 僅支持 `iam:PassRole`，這是 `CreateCampaign` API 操作所必需的。

## 搭配 AWS IoT 使用臨時登入資料 FleetWise

支援臨時憑證 是

當您使用臨時憑據登錄時，某些 AWS 服務不起作用。如需其他資訊，包括哪些 AWS 服務與臨時登入資料 [搭配 AWS 服務使用](#)，請參閱 IAM 使用者指南中的 IAM。

如果您使用除了使用者名稱和密碼以外的任何方法登入，則您正在 AWS Management Console 使用臨時認證。例如，當您 AWS 使用公司的單一登入 (SSO) 連結存取時，該程序會自動建立暫時認證。當您以使用者身分登入主控台，然後切換角色時，也會自動建立臨時憑證。如需切換角色的詳細資訊，請參閱《IAM 使用者指南》中的 [切換至角色 \(主控台\)](#)。

您可以使用 AWS CLI 或 AWS API 手動建立臨時登入資料。然後，您可以使用這些臨時登入資料來存取 AWS。AWS 建議您動態產生臨時登入資料，而非使用長期存取金鑰。如需詳細資訊，請參閱 [IAM 中的暫時性安全憑證](#)。

## IoT 的 AWS 跨服務主體權限 FleetWise

支援轉寄存取工作階段 (FAS) 是

當您使用 IAM 使用者或角色在中執行動作時 AWS，您會被視為主體。使用某些服務時，您可能會執行某個動作，進而在不同服務中啟動另一個動作。FAS 會使用主體呼叫的權限 AWS 服務，並結合要求 AWS 服務 向下游服務發出要求。只有當服務收到需要與其 AWS 服務 他資源互動才能完成的請求時，才會發出 FAS 請求。在此情況下，您必須具有執行這兩個動作的許可。如需提出 FAS 請求時的政策詳細資訊，請參閱 [《轉發存取工作階段》](#)。

## AWS IoT 的服務角色 FleetWise

|        |   |
|--------|---|
| 支援服務角色 | 否 |
|--------|---|

服務角色是服務擔任的 [IAM 角色](#)，可代您執行動作。IAM 管理員可以從 IAM 內建立、修改和刪除服務角色。如需更多資訊，請參閱 IAM 使用者指南中的 [建立角色以委派許可給 AWS 服務](#)。

### Warning

變更服務角色的權限可能會中斷 AWS IoT FleetWise 功能。僅在 AWS IoT FleetWise 提供指導時，才編輯服務角色。

## IoT 的 AWS 服務連結角色 FleetWise

|           |   |
|-----------|---|
| 支援服務連結角色。 | 否 |
|-----------|---|

服務連結角色是連結至 AWS 服務服務可以擔任代表您執行動作的角色。服務連結角色會顯示在您的中，AWS 帳戶 且屬於服務所有。IAM 管理員可以檢視，但不能編輯服務連結角色的許可。

如需建立或管理服務連結角色的詳細資訊，請參閱 [可搭配 IAM 運作的 AWS 服務](#)。在表格中尋找服務，其中包含服務連結角色欄中的 Yes。選擇是連結，以檢視該服務的服務連結角色文件。

## 針對 AWS IoT 使用服務連結角色 FleetWise

AWS IoT FleetWise 使用 AWS Identity and Access Management (IAM) [服務連結角色](#)。服務連結角色是直接連結至 AWS IoT FleetWise 的唯一 IAM 角色類型。服務連結角色由 AWS IoT 預先定義，FleetWise 並包含 AWS IoT 傳送指標到 Amazon CloudWatch 所 FleetWise 需的許可。如需詳細資訊，請參閱 [FleetWise 用 Amazon 監控 AWS IoT CloudWatch](#)。



服務連結角色可讓您 FleetWise 更快速地設定 AWS IoT，因為您不需要手動新增必要的許可。AWS IoT 會 FleetWise 定義其服務連結角色的許可，除非另有定義，否則只有 AWS IoT FleetWise 可以擔任其角色。已定義的許可包括信任政策和許可政策。此許可政策無法附加到任何其他 IAM 實體。

您必須先刪除服務連結角色的相關資源，才能將其刪除。這樣可以保護您的 AWS IoT FleetWise 資源，因為您無法意外移除存取資源的許可。

如需其他支援服務連結角色之服務[AWS 務的相關資訊](#)，請參閱[使用 IAM](#)的服務，並在服務連結角色欄中尋找具有 Yes 的服務。若要檢視該服務的服務連結角色文件，請選擇具有連結的 Yes (是)。

## AWS IoT 的服務連結角色許可 FleetWise

AWS IoT FleetWise 使用名為的服務連結角色 `AWSServiceRoleForIoT FleetWise`— 用於 AWS IoT FleetWise 所有 out-of-the-box 許可的 AWS 受管政策。

服務 `AWSServiceRoleForIoT FleetWise` 服務連結角色會信任下列服務擔任該角色：

- `IoTFleetWise`

名為的角色許可政策 `AWSIoT FleetWise ServiceRolePolicy` 允許 AWS IoT FleetWise 在指定的資源上完成以下動作：

- 動作：`cloudwatch:PutMetricData`在資源上：\*

您必須設定許可，IAM 實體 (如使用者、群組或角色) 才可建立、編輯或刪除服務連結角色。如需詳細資訊，請參閱 IAM 使用者指南中的[服務連結角色許可](#)。

## 為 AWS IoT 建立服務連結角色 FleetWise

您不需要手動建立一個服務連結角色。當您在 AWS IoT FleetWise 主控台、或 AWS API 中註冊帳戶時 AWS CLI，AWS IoT FleetWise 會為您建立服務連結角色。如需詳細資訊，請參閱[進行設定](#)。

## 在 AWS IoT FleetWise (主控台) 中建立服務連結角色

您不需要手動建立一個服務連結角色。當您在 AWS IoT FleetWise 主控台、AWS CLI 或 AWS API 中註冊帳戶時，AWS IoT FleetWise 會為您建立服務連結角色。

## 編輯 AWS IoT 的服務連結角色 FleetWise

您無法在 AWS IoT FleetWise 中編輯 `AWSServiceRoleForIoT FleetWise` 服務連結角色。由於各種實體可能會參考您建立的任何服務連結角色，因此您無法變更角色的名稱。然而，您可使用 IAM 來編輯角色描述。如需詳細資訊，請參閱 IAM 使用者指南中的[編輯服務連結角色](#)。



## 清除服務連結角色

在您使用 IAM 刪除服務連結角色之前，您必須先刪除該角色所使用的任何資源。

### Note

如果您嘗試刪除資源時，AWS IoT FleetWise 正在使用該角色，則刪除可能會失敗。若此情況發生，請等待數分鐘後並再次嘗試操作。若要了解如何 service-linked-role 透過主控台、AWS CLI 或 AWS API 刪除，請參閱 IAM 使用者指南中的[使用服務連結角色](#)。

如果您刪除此服務連結角色，然後需要重新建立，則可以在 AWS IoT FleetWise 註冊帳戶。FleetWise 然後，AWS IoT 會再次為您建立服務連結角色。

## IoT 的身分識別原則範例 AWS FleetWise

依預設，使用者和角色沒有建立或修改 AWS IoT FleetWise 資源的權限。他們也無法使用 AWS Management Console、AWS Command Line Interface (AWS CLI) 或 AWS API 來執行工作。若要授予使用者對其所需資源執行動作的許可，IAM 管理員可以建立 IAM 政策。然後，管理員可以將 IAM 政策新增至角色，使用者便能擔任這些角色。

若要了解如何使用這些範例 JSON 政策文件建立 IAM 身分型政策，請參閱《IAM 使用者指南》中的[建立 IAM 政策](#)。

有關 AWS IoT 定義的動作和資源類型的詳細資訊 FleetWise，包括每種資源類型的 ARN 格式，請參閱服務授權參考 FleetWise 中的[AWS IoT 動作、資源和條件金鑰](#)。

### 主題

- [政策最佳實務](#)
- [使用 AWS IoT FleetWise 主控台](#)
- [允許使用者檢視他們自己的許可](#)
- [在 Amazon Timestream 中訪問資源](#)

## 政策最佳實務

以身分識別為基礎的原則會決定某人是否可以建立、存取或刪除您帳戶中的 AWS IoT FleetWise 資源。這些動作可能會讓您的 AWS 帳戶產生費用。當您建立或編輯身分型政策時，請遵循下列準則及建議事項：

- 開始使用 AWS 受管原則並邁向最低權限權限 — 若要開始授與使用者和工作負載的權限，請使用可授與許多常見使用案例權限的 AWS 受管理原則。它們可用在您的 AWS 帳戶。建議您透過定義特定於您使用案例的 AWS 客戶管理政策，進一步降低使用權限。如需更多資訊，請參閱 IAM 使用者指南中的 [AWS 受管政策](#) 或 [任務職能的 AWS 受管政策](#)。
- 套用最低許可許可 – 設定 IAM 政策的許可時，請僅授予執行任務所需的權限。為實現此目的，您可以定義在特定條件下可以對特定資源採取的動作，這也稱為最低權限許可。如需使用 IAM 套用許可的更多相關資訊，請參閱 IAM 使用者指南中的 [IAM 中的政策和許可](#)。
- 使用 IAM 政策中的條件進一步限制存取權 – 您可以將條件新增至政策，以限制動作和資源的存取。例如，您可以撰寫政策條件，指定必須使用 SSL 傳送所有請求。您也可以使用條件來授與對服務動作的存取權 (如透過特定) 使用這些動作 AWS 服務，例如 AWS CloudFormation。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM JSON 政策元素：條件](#)。
- 使用 IAM Access Analyzer 驗證 IAM 政策，確保許可安全且可正常運作 – IAM Access Analyzer 驗證新政策和現有政策，確保這些政策遵從 IAM 政策語言 (JSON) 和 IAM 最佳實務。IAM Access Analyzer 提供 100 多項政策檢查及切實可行的建議，可協助您編寫安全且實用的政策。如需更多資訊，請參閱 IAM 使用者指南中的 [IAM Access Analyzer 政策驗證](#)。
- 需要多因素身份驗證 (MFA) — 如果您的案例需要 IAM 使用者或根使用者 AWS 帳戶，請開啟 MFA 以獲得額外的安全性。若要在呼叫 API 作業時請求 MFA，請將 MFA 條件新增至您的政策。如需更多資訊，請參閱 [IAM 使用者指南](#) 中的設定 MFA 保護的 API 存取。

如需 IAM 中最佳實務的相關資訊，請參閱 IAM 使用者指南中的 [IAM 安全最佳實務](#)。

## 使用 AWS IoT FleetWise 主控台

若要存取 AWS IoT FleetWise 主控台，您必須擁有一組最低權限。這些權限必須允許您列出和檢視 AWS 帳戶。AWS FleetWise 如果您建立比最基本必要許可更嚴格的身分型政策，則對於具有該政策的實體 (使用者或角色) 而言，主控台就無法如預期運作。

您不需要為僅對 AWS CLI 或 AWS API 進行呼叫的使用者允許最低主控台權限。反之，只需允許存取符合他們嘗試執行之 API 操作的動作就可以了。

若要確保使用者和角色仍可使用 AWS IoT FleetWise 主控台，請同時將 AWS IoT FleetWise ConsoleAccess 或 ReadOnly AWS 受管理的原則附加至實體。如需詳細資訊，請參閱《IAM 使用者指南》中的 [新增許可到使用者](#)。

## 允許使用者檢視他們自己的許可

此範例會示範如何建立政策，允許 IAM 使用者檢視附加到他們使用者身分的內嵌及受管政策。此原則包含在主控台上或以程式設計方式使用 AWS CLI 或 AWS API 完成此動作的權限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

## 在 Amazon Timestream 中訪問資源

AWS 雲端 在使用 AWS IoT 之前 FleetWise，您必須註冊 AWS 帳戶、IAM 和 Amazon Timestream 資源，以授予 AWS IoT FleetWise 權限，以代表您將車輛資料傳送給您。要註冊，您需要：

- Amazon Timestream 數據庫。
- 在指定的 Amazon Timestream 資料庫中建立的表格。
- 允許 AWS IoT FleetWise 將資料傳送到 Amazon Timestream 的 IAM 角色。

如需詳細資訊，包括程序和範例原則，請參閱進行[設定](#)。

## 疑難排解 AWS IoT FleetWise 身分與存取

使用下列資訊可協助您診斷並修正使用 AWS IoT FleetWise 和 IAM 時可能會遇到的常見問題。

### 主題

- [我沒有在 AWS IoT 中執行動作的授權 FleetWise](#)
- [我沒有授權執行 iam : PassRole](#)
- [我想讓我以外的人員存 AWS 帳戶 取我的 AWS IoT FleetWise 資源](#)

### 我沒有在 AWS IoT 中執行動作的授權 FleetWise

如果 AWS Management Console 告訴您您沒有執行動作的授權，則您必須聯絡您的管理員以尋求協助。您的管理員是為您提供簽署憑證的人員。

當 mateojackson IAM 使用者嘗試使用主控台來檢視虛構 *myVehicle* 資源的詳細資料，但沒有 `iotfleetwise:GetVehicleStatus` 權限時，就會發生下列範例錯誤。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
iotfleetwise:GetVehicleStatus on resource: myVehicle
```

在此情況下，Mateo 會請求管理員更新他的政策，允許他使用 *myVehicle* 動作存取 `iotfleetwise:GetVehicleStatus` 資源。

### 我沒有授權執行 iam : PassRole

如果您收到未獲授權執行 `iam:PassRole` 動作的錯誤訊息，則必須更新您的原則以允許您將角色傳遞至 AWS IoT FleetWise。

有些 AWS 服務 允許您將現有角色傳遞給該服務，而不是建立新的服務角色或服務連結角色。如需執行此作業，您必須擁有將角色傳遞至該服務的許可。

當名為的 IAM 使用者 `marymajor` 嘗試使用主控台在 AWS IoT 中執行動作時，就會發生下列範例錯誤 FleetWise。但是，動作請求服務具備服務角色授予的許可。Mary 沒有將角色傳遞至該服務的許可。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

在這種情況下，Mary 的政策必須更新，允許她執行 iam:PassRole 動作。

如果您需要協助，請聯絡您的 AWS 系統管理員。您的管理員提供您的簽署憑證。

## 我想讓我以外的人員存取 AWS 帳戶 取我的 AWS IoT FleetWise 資源

您可以建立一個角色，讓其他帳戶中的使用者或您組織外部的人員存取您的資源。您可以指定要允許哪些信任物件取得該角色。針對支援基於資源的政策或存取控制清單 (ACL) 的服務，您可以使用那些政策來授予人員存取您的資源的許可。

如需進一步了解，請參閱以下內容：

- 若要瞭解 AWS IoT 是否 FleetWise 支援這些功能，請參閱 [AWS IoT 如何與 IAM FleetWise 搭配運作](#)。
- 若要了解如何提供對您所擁有資源 AWS 帳戶 的存取權，請參閱 [《IAM 使用者指南》中您擁有的另一 AWS 帳戶 個 IAM 使用者提供存取權限](#)。
- 若要了解如何將資源存取權提供給第三方 AWS 帳戶，請參閱 IAM 使用者指南中的 [提供第三方 AWS 帳戶 擁有的存取權](#)。
- 若要了解如何透過聯合身分提供存取權，請參閱 IAM 使用者指南中的 [將存取權提供給在外部進行身分驗證的使用者 \(聯合身分\)](#)。
- 若要了解使用角色和資源型政策進行跨帳戶存取之間的差異，請參閱 IAM 使用者指南中的 [IAM 角色 與資源型政策的差異](#)。

## AWS IoT 的合規性驗證 FleetWise

### Note

AWS IoT FleetWise 不在任何 AWS 合規性方案的範圍內。

若要瞭解 AWS 服務 是否屬於特定規範遵循方案的範圍內，請參閱 [AWS 服務 遵循規範計劃](#) 方案中的，並選擇您感興趣的合規方案。如需一般資訊，請參閱 [AWS 規範計劃](#) [AWS](#)。

您可以使用下載第三方稽核報告 AWS Artifact。如需詳細資訊，請參閱 [下載中的報告中的](#) AWS Artifact。

您在使用時的合規責任取決 AWS 服務 於資料的敏感性、公司的合規目標以及適用的法律和法規。AWS 提供下列資源以協助遵循法規：

- [安全性與合規性快速入門指南](#) — 這些部署指南討論架構考量，並提供部署以安全性和合規性 AWS 為重點的基準環境的步驟。
- [在 Amazon Web Services 上架構 HIPAA 安全性與合規性](#) — 本白皮書說明公司如何使用建立符合 HIPAA 資格的應 AWS 應用程式。

#### Note

並非所有人 AWS 服務 都符合 HIPAA 資格。如需詳細資訊，請參閱 [HIPAA 資格服務參照](#)。

- [AWS 合規資源AWS](#) — 此工作簿和指南集合可能適用於您的產業和所在地。
- [AWS 客戶合規指南](#) — 透過合規的角度瞭解共同的責任模式。這份指南總結了在多個架構 (包括美國國家標準技術研究所 (NIST)、支付卡產業安全標準委員會 (PCI) 和國際標準化組織 (ISO)) 中，保 AWS 服務 護指引並對應至安全控制的最佳實務。
- [使用AWS Config 開發人員指南中的規則評估資源](#) — 此 AWS Config 服務會評估您的資源組態符合內部實務、產業準則和法規的程度。
- [AWS Security Hub](#)— 這 AWS 服務 提供了內部安全狀態的全面視圖 AWS。Security Hub 使用安全控制，可評估您的 AWS 資源並檢查您的法規遵循是否符合安全業界標準和最佳實務。如需支援的服務和控制清單，請參閱 [Security Hub controls reference](#)。
- [Amazon GuardDuty](#) — 透過監控環境中的 AWS 帳戶可疑和惡意活動，藉此 AWS 服務 偵測您的工作負載、容器和資料的潛在威脅。GuardDuty 可協助您滿足特定合規性架構所要求的入侵偵測需求，例如 PCI DSS 等各種合規性需求。
- [AWS Audit Manager](#)— 這 AWS 服務 有助於您持續稽核您的 AWS 使用情況，以簡化您管理風險的方式，以及遵守法規和業界標準的方式。

## AWS IoT 的韌性 FleetWise

AWS 全球基礎架構是圍繞區 AWS 域和可用區域建立的。區域提供多個分開且隔離的實際可用區域，並以低延遲、高輸送量和高度備援網路連線相互連結。透過可用區域，您可以設計與操作的應用程式和資料庫，在可用區域之間自動容錯移轉而不會發生中斷。可用區域的可用性、容錯能力和擴展能力，均較單一或多個資料中心的傳統基礎設施還高。

如需區域和可用區域的相關 AWS 資訊，請參閱[AWS 全域基礎結構](#)。



**Note**

AWS IoT 處理的資料 FleetWise 會儲存在 Amazon Timestream 資料庫中。時間流支援備份至其他 AWS 可用區域或區域。但是，您可以使用 Timestream SDK 編寫自己的應用程序來查詢數據並將其保存到您選擇的目的地。

如需 Amazon Timestream 的詳細資訊，請參閱 [Amazon Timestream 開發人員指南](#) 中的。

## AWS IoT 中的基礎架構安全 FleetWise

作為一項託管服務，AWS IoT FleetWise 受到 AWS 全球網絡安全的保護。有關 AWS 安全服務以及如何 AWS 保護基礎結構的詳細資訊，請參閱 [AWS 雲端安全](#) 若要使用基礎架構安全性的最佳做法來設計您的 AWS 環境，請參閱 [安全性支柱架構](#) 良好的架構中的基礎結構保護。

您可以使用 AWS 已發佈的 API 呼叫 FleetWise 透過網路存取 AWS IoT。使用者端必須支援下列專案：

- Transport Layer Security (TLS)。我們需要 TLS 1.2 並建議使用 TLS 1.3。
- 具備完美轉送私密(PFS)的密碼套件，例如 DHE (Ephemeral Diffie-Hellman) 或 ECDHE (Elliptic Curve Ephemeral Diffie-Hellman)。現代系統(如 Java 7 和更新版本)大多會支援這些模式。

此外，請求必須使用存取金鑰 ID 和與 IAM 主體相關聯的私密存取金鑰來簽署。或者，您可以透過 [AWS Security Token Service](#) (AWS STS) 來產生暫時安全憑證來簽署請求。

您可以從任何網路位置呼叫這些 API 作業，但 AWS IoT FleetWise 確實支援以資源為基礎的存取原則，其中可能包含以來源 IP 位址為基礎的限制。您也可以使用 AWS IoT FleetWise 政策來控制來自特定 Amazon Virtual Private Cloud 端 (Amazon VPC) 端點或特定 VPC 的存取。實際上，這可以將對特定 AWS IoT FleetWise 資源的網路存取從網路中的特定 VPC 隔離出來 AWS。

### 主題

- [FleetWise 透過介面 VPC AWS 端點連線至 IoT](#)

## FleetWise 透過介面 VPC AWS 端點連線至 IoT

您可以使用 Virtual Private Cloud (VPC [AWS PrivateLink](#) (VPC) 中的介面 VPC 端點 () 直接連線到 AWS IoT，而不是 FleetWise 透過網際網路連線。當您使用介面虛擬私人雲端端點時，VPC 與 AWS

IoT FleetWise 之間的通訊完全在網路中進行。AWS 每個 VPC 端點皆會由一個或多個具私有 IP 地址[彈性網路介面](#) (ENI) 來表示，而該介面位於 VPC 子網路中。

無需網際網路閘道、NAT 裝置、VPN 連線或 AWS Direct Connect 連線，FleetWise 即可將您的 VPC 直接連線至 AWS IoT。VPC 中的執行個體不需要公有 IP 位址即可與 AWS IoT FleetWise API 通訊。

若要 FleetWise 透過 VPC 使用 AWS IoT，您必須從 VPC 內的執行個體進行連線，或使用 AWS Virtual Private Network (VPN) 或將私人網路連線到 VPC。AWS Direct Connect 如需有關 Amazon VPN 的資訊，請參閱《Amazon Virtual Private Cloud 使用者指南》中的[VPN 連接](#)。若要取得有關資訊 AWS Direct Connect，請參閱《[使用指南](#)》中的[AWS Direct Connect](#) [〈建立連接〉](#)。

您可以使用 AWS 主控台或 AWS Command Line Interface (AWS CLI) 命令建立介面 VPC FleetWise 人雲端端點以連接到 AWS IoT。如需詳細資訊，請參閱[建立介面端點](#)。

建立介面 VPC 端點後，如果您為端點啟用私人 DNS 主機名稱，則預設 AWS IoT 端點會解析為您的 VPC FleetWise 端點。AWS IoT FleetWise 的預設服務名稱端點格式如下。

```
iotfleetwise.Region.amazonaws.com
```

如果您未啟用私人 DNS 主機名稱，Amazon VPC 會提供 DNS 端點名稱，您可以使用下列格式。

```
VPCE_ID.iotfleetwise.Region.vpce.amazonaws.com
```

如需詳細資訊，請參閱 Amazon VPC 使用者指南中的介面虛擬私人雲端端點 ([AWS PrivateLink](#))。

AWS IoT FleetWise 支援呼叫 VPC 內的所有 [API 動作](#)。

可以將 VPC 端點政策附接至某個 VPC 端點，以控制 IAM 主體的存取權。您也可以將安全群組與 VPC 端點建立關聯，藉以根據網路流量的來源和目的地 (例如 IP 位址範圍) 來控制輸入和輸出存取。如需詳細資訊，請參閱[使用 VPC 端點控制服務的存取](#)。

## 建立適用於 IoT 的 VPC 私人雲端端點原則 AWS FleetWise

您可以為 AWS IoT 的 Amazon VPC 端點建立政策，以指 FleetWise 定下列項目：

- 可執行或不可執行動作的主體
- 可以執行或不能執行的動作

如需詳細資訊，請參閱《Amazon VPC 使用者指南》中的[使用 VPC 端點控制對服務的存取](#)。



## Example — 用於拒絕來自指定 AWS 帳戶的所有訪問的 VPC 端點策略

下列 VPC 端點策略會拒絕 AWS 帳戶 **1234567890** 12 所有使用該端點的 API 呼叫。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    },
    {
      "Action": "*",
      "Effect": "Deny",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "123456789012"
        ]
      }
    }
  ]
}
```

## Example – 可用來僅允許來自指定 IAM 主體 (使用者) 之 VPC 存取 VPC 端點政策

**## VPC ##### AWS ## 123456789012 #####** 它會拒絕所有其他 IAM 主體存取端點。

```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/lijuan"
        ]
      }
    }
  ]
}
```

## Example — AWS IoT FleetWise 動作的 VPC 私人雲端端點政策

以下是 AWS IoT 端點原則的範例 FleetWise。##### 123456789012 ###  
*FleetWise IAM ### FleetWise #### AWS IoT ### AWS ##*

```
{
  "Statement": [
    {
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/fleetWise"
        ],
      },
      "Resource": "*",
      "Effect": "Allow",
      "Action": [
        "iotfleetwise:ListFleets",
        "iotfleetwise:ListCampaigns",
        "iotfleetwise:CreateVehicle",
      ]
    }
  ]
}
```

## AWS IoT 中的組態和漏洞分析 FleetWise

IoT 環境可能由大量裝置組成，各自具有多樣化的功能、長時間在線上，並且散佈在多個地理位置。這些特點使裝置設定複雜且極易出錯。此外，由於設備通常受到計算能力，內存和存儲功能的限制，因此在設備上使用加密和其他形式的安全性也受到限制。裝置通常會使用具有已知漏洞的軟體。這些因素使 IoT 設備（包括為 AWS IoT 收集數據的車輛）成為黑客的吸引力目標 FleetWise，並使其難以持續保護它們。

配置和 IT 控制是與您（我們的客戶）AWS 之間共同責任。如需詳細資訊，請參閱 AWS [共用的責任模型](#)。

## AWS IoT 的安全性最佳做法 FleetWise

AWS IoT FleetWise 提供許多安全性功能，可在您開發和實作自己的安全性原則時考量。以下最佳實務為一般準則，並不代表完整的安全解決方案。這些最佳實務可能不適用或無法滿足您的環境需求，因此請將其視為實用建議就好，而不要當作是指示。

若要深入瞭解安全性，AWS IoT 請參閱AWS IoT 開發人員指南 [AWS IoT Core中的安全性最佳做法](#)

## 盡可能授予最低的許可

在 IAM 角色中使用最低一組許可，遵循最低權限原則。限制 IAM 政策中Action和Resource屬性的\*萬用字元使用。相反地，在可能的情況下，宣告一組有限的動作和資源。如需最低權限和其他原則最佳實務的詳細資訊，請參閱 [the section called “政策最佳實務”](#)。

## 請勿記錄敏感資訊

您應該防止記錄登入資料和其他個人識別資訊 (PII)。我們建議您實施以下保護措施：

- 請勿在裝置名稱中使用敏感資訊。
- 不要在 AWS IoT FleetWise 資源的名稱和 ID 中使用敏感信息，例如活動名稱，解碼器清單，車輛模型和信號目錄，或車輛和車隊的 ID。

## 用 AWS CloudTrail 於查看 API 呼叫歷史記錄

您可以檢視在您帳戶上發出的 AWS IoT FleetWise API 呼叫歷史記錄，以進行安全性分析和操作疑難排解。若要接收對您帳戶進行之 AWS IoT FleetWise API 呼叫的歷史記錄，只要 CloudTrail 在 AWS Management Console. 如需詳細資訊，請參閱 [the section called “CloudTrail 日誌”](#)。

## 讓裝置的時鐘保持同步

在裝置上保持準確的時間是很重要的。X.509 憑證具到期日期和時間。裝置上的時鐘用來驗證伺服器憑證是否仍然有效。裝置時鐘可能會隨著時間而偏移，或是電池可能會放電。

如需詳細資訊，請參閱AWS IoT Core 開發人員指南 [中的保持裝置時鐘同步](#)最佳做法。

# 監控 AWS IoT FleetWise

監控是維持 AWS IoT FleetWise 和其他AWS解決方案的可靠性、可用性和效能的重要組成部分。AWS提供下列監控工具來觀看 AWS IoT FleetWise、在發生錯誤時回報，並在適當時採取自動動作：

- Amazon 會即時 CloudWatch監控您的AWS資源和執行AWS的應用程式。您可以收集和追蹤指標、建立自訂儀表板，以及設定警示，在指標達到您指定的閾值時通知您或採取動作。例如，您可以 CloudWatch 追蹤 Amazon EC2 執行個體的 CPU 使用率或其他指標，並在需要時自動啟動新執行個體。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。
- Amazon CloudWatch 日誌可用來監控、存放和存取 Amazon EC2 執行個體和其他來源的日誌檔。CloudTrail CloudWatch 記錄檔可以監控記錄檔中的資訊，並在符合特定臨界值時通知您。您也可以將日誌資料存檔在高耐用性的儲存空間。如需詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。
- AWS CloudTrail 會擷取來自或代表 AWS 帳戶發出的 API 呼叫和相關事件。然後，它會將日誌檔傳送到您指定的 Amazon S3 儲存貯體。您可以找出哪些使用者和帳戶呼叫 AWS、發出呼叫的來源 IP 地址，以及呼叫的發生時間。如需詳細資訊，請參閱《AWS CloudTrail 使用者指南》<https://docs.aws.amazon.com/awscloudtrail/latest/userguide/>。

## FleetWise 用 Amazon 監控 AWS IoT CloudWatch

Amazon CloudWatch 指標是監控AWS資源及其效能的一種方式。AWSIoT FleetWise 將指標傳送至 CloudWatch. 您可以使用AWS Management ConsoleAWS CLI、或 API 來列出 AWS IoT FleetWise 傳送至的指標 CloudWatch。如需詳細資訊，請參閱 [Amazon CloudWatch 使用者指南](#)。

### Important

您必須進行設定，以便 AWS IoT FleetWise 可以將指標傳送至 CloudWatch。如需詳細資訊，請參閱[進行設定](#)。

AWS/IoTFleetWise 命名空間包含下列指標。

## 信號指標

| 指標                     | 描述   |
|------------------------|--|
| IllegalMessageFromEdge | <p>從車輛發送並由 AWS IoT 接收的消息與所需的格式 FleetWise 不匹配。</p> <p>單位：計數</p> <p>尺寸: VehicleName</p> <p>有效的統計資訊：總和</p>                              |
| MessageThrottled       | <p>從車輛傳送至 AWS IoT 的訊息 FleetWise 被限制。這是因為您已超過目前區域中此帳戶的<a href="#">服務限制</a>。</p> <p>單位：計數</p> <p>尺寸: VehicleName</p> <p>有效的統計資訊：總和</p> |
| ModelingError          | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise 包含無法根據車輛型號驗證的信號。</p> <p>單位：計數</p> <p>尺寸: ModelManifestName</p> <p>有效的統計資訊：總和</p>                  |
| DecodingError          | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise 包含無法對車輛的解碼器清單進行解碼器的信號。</p> <p>單位：計數</p> <p>尺寸: DecoderName</p> <p>有效的統計資訊：總和</p>                  |

## 行銷活動指標

| 指標               | 描述  |
|------------------|---|
| VehicleNotFound  | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise，其中車輛未知。</p> <p>單位：計數</p> <p>尺寸: VehicleName</p> <p>有效的統計資訊：總和</p>    |
| CampaignInvalid  | <p>從車輛傳送並由 AWS IoT 接收的訊息 FleetWise，其中行銷活動無效。</p> <p>單位：計數</p> <p>尺寸: CampaignName</p> <p>有效的統計資訊：總和</p> |
| CampaignNotFound | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise，其中活動未知。</p> <p>單位：計數</p> <p>尺寸: CampaignName</p> <p>有效的統計資訊：總和</p>   |

## 促銷活動資料目的地

| 指標                   | 描述  |
|----------------------|---|
| TimestreamWriteError | <p>AWSIoT FleetWise 無法將車輛的消息寫入 Amazon Timestream 表。</p> <p>單位：計數</p> <p>尺寸: DatabaseName, TableName</p> |

| 指標            | 描述   |
|---------------|--|
|               | 有效的統計資訊：總和   |
| S3 WriteError | <p>AWSIoT FleetWise 無法將消息從車輛寫入亞馬遜簡單存儲服務 ( Amazon S3 ) 存儲桶。</p> <p>單位：計數</p> <p>尺寸: BucketName</p> <p>有效的統計資訊：總和</p>    |
| S3 ReadError  | <p>AWSIoT FleetWise 無法從亞馬遜簡單存儲服務 ( Amazon S3 ) 存儲桶中的車輛讀取對象密鑰。</p> <p>單位：計數</p> <p>尺寸: BucketName</p> <p>有效的統計資訊：總和</p> |

#### 客戶管理的AWS KMS關鍵指標

| 指標                 | 描述   |
|--------------------|--|
| 公里 KeyAccessDenied | <p>AWSIoT FleetWise 無法將訊息從車輛寫入時間流表或 Amazon S3 儲存貯體，因為AWS KMS 金鑰存取遭拒錯誤。</p> <p>單位：計數</p> <p>維度:公里 KeyId</p> <p>有效的統計資訊：總和</p> |

# FleetWise 使用 Amazon CloudWatch 日誌監控 AWS IoT

Amazon CloudWatch Logs 會監控資源中發生的事件，並在發生任何問題時向您發出警示。如果收到警示，您可以存取記錄檔以取得有關特定事件的資訊。如需詳細資訊，請參閱 [Amazon CloudWatch 日誌使用者指南](#)。

## 在 CloudWatch 主控台中檢視 AWS IoT FleetWise 記錄

### Important

在 CloudWatch 主控台中查看 AWS IoT FleetWise 記錄群組之前，請確定下列條件為真：

- 您已啟用 AWS IoT 中的記錄功能 FleetWise。如需記錄的詳細資訊，請參閱 [設定 AWS IoT FleetWise 記錄](#)。
- AWS IoT 作業已經寫入記錄項目。

在 CloudWatch 主控台中檢視您的 AWS IoT FleetWise 記錄

1. 開啟 [CloudWatch 主控台](#)。
2. 在瀏覽窗格中，選擇 [記錄] > [記錄群組]。
3. 選擇 日誌群組。
4. 選擇 Search log group (搜尋日誌群組)。您將看到為您的帳戶生成的日誌事件的完整列表。
5. 選擇展開圖示以查看個別串流，並尋找記錄層級為的所有記錄檔ERROR。

您也可以在此篩選事件搜尋方塊中輸入查詢。例如，您可以嘗試以下查詢：

```
{ $.logLevel = "ERROR" }
```

如需有關建立篩選器運算式的詳細資訊，請參閱 Amazon CloudWatch Logs 使用者指南中的 [篩選器和模式語法](#)。

### Example 記錄項目

```
{
  "accountId": "123456789012",
  "vehicleName": "test-vehicle",
  "message": "Unrecognized signal ID",
```



```

"eventType": "MODELING_ERROR",
"logLevel": "ERROR",
"timestamp": 1685743214239,
"campaignName": "test-campaign",
"signalCatalogName": "test-catalog",
"signalId": 10242
}

```

## 訊號事件類型

| 事件類型      | 描述   |
|-----------|--|
| 模型錯誤      | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise 包含無法根據車輛型號驗證的信號。</p> <p>屬性：車輛名稱，商品名稱，信號 ID，信號值 signalCatalogName，最小值，最大值， signalValueRange signalValueRange modelManifestName</p> |
| 來自邊緣的非法訊息 | <p>從車輛發送並由 AWS IoT 接收的消息與所需的格式 FleetWise 不匹配。</p> <p>屬性：車輛名稱、行銷活動名稱、 signalCatalogName</p>   |
| 解碼錯誤      | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise 包含無法對車輛的解碼器清單進行解碼器的信號。</p> <p>屬性：促銷活動名稱、 signalCatalogName decoderManifestName、(選用) 信號名稱、(選用) S3uri</p>                           |

## 廣告活動類型

| 事件類型   | 描述   |
|--------|--|
| 沒有找到車輛 | <p>從車輛發送並由 AWS IoT 接收的消息 FleetWise，其中車輛未知。</p> |

| 事件類型  | 描述  |
|-------|---|
|       | 屬性：車輛名稱、行銷活動名稱  |
| 未找到活動 | 從車輛發送並由 AWS IoT 接收的消息 FleetWise，其中活動未知。<br><br>屬性：車輛名稱（可選），廣告系列名稱 |
| 活動無效  | 從車輛發送並由 AWS IoT 接收的消息 FleetWise，其中活動無效。<br><br>屬性：車輛名稱（可選），廣告系列名稱 |

### 廣告活動資料目的地事件

| 事件類型       | 描述  |
|------------|---|
| 時間流 _ 寫入錯誤 | AWSIoT FleetWise 無法將車輛的消息寫入 Amazon Timestream 表。<br><br>屬性：車輛名稱、活動名稱、timestreamDatabaseName timestreamTableName |
| S3_ 寫入錯誤   | AWSIoT FleetWise 無法將消息從車輛寫入亞馬遜簡單存儲服務（Amazon S3）存儲桶。<br><br>屬性：廣告活動名稱、目的地名稱                                      |
| S3_ 讀取錯誤   | AWSIoT FleetWise 無法從亞馬遜簡單存儲服務（Amazon S3）存儲桶中的車輛讀取對象密鑰。<br><br>屬性：廣告活動名稱、目的地名稱                                   |

## 客戶管理的AWS KMS金鑰事件類型

| 事件類型           | 描述  |
|----------------|---|
| KMS 密鑰 _ 訪問被拒絕 | AWSIoT FleetWise 無法將訊息從車輛寫入時間流表或 Amazon S3 儲存貯體，因為AWS KMS 金鑰存取遭拒錯誤。 |

### Attributes

所有 CloudWatch 記錄項目都包含下列屬性：

accountId

您的 AWS 帳戶 ID。

eventType

產生日誌的事件類型。事件類型的值會根據產生該日誌項目的事件而定。每個日誌項目說明都包含該 eventType 日誌項目的值。

logLevel

正在使用的記錄層級。如需詳細資訊，請參閱AWS IoT Core開發人員指南中的[記錄層級](#)。

message

包含有關記錄檔的特定詳細資料。

timestamp

AWSIoT FleetWise 處理記錄檔時的紀元毫秒時間戳記。

### 選擇性屬性

CloudWatch 記錄項目可選擇性地包含這些屬性，具體取決於eventType：

decoderManifestName

包含信號的解碼器清單的名稱。

目的地名稱

車輛資料的目的地名稱。例如，Amazon S3 存儲桶名稱。

## 促銷活動名稱

行銷活動的名稱。

## signalCatalogName

包含信號的信號目錄的名稱。

## 訊號識別碼

錯誤訊號的識別碼。

## 信號

錯誤信號 ID 的列表。

## 信號名稱

訊號的名稱。

## signalTimestampEpoch

錯誤訊號的時間戳記。

## 信號值

誤差信號的值。

## signalValueRange

誤差信號的最大範圍。

## signalValueRange

誤差信號的最小範圍。

## S3URI

來自車輛消息的 Amazon 離子文件的 Amazon S3 唯一標識符。

## timestreamDatabaseName

Timestream 資料庫的名稱。

## timestreamTableName

Timestream 資料表的名稱。

## 車輛名稱

車輛的名稱。

## 設定 AWS IoT FleetWise 記錄

您可以將 AWS IoT 記錄 FleetWise 資料傳送至 CloudWatch 記錄群組。CloudWatch 如果 AWS IoT FleetWise 無法處理來自車輛的訊息，記錄可提供可見性。例如，這可能是因為配置錯誤或其他客戶端錯誤而發生。您會收到任何錯誤的通知，以便識別和緩解問題。

您必須先建立記錄群組 CloudWatch，才能將記錄檔傳送至。CloudWatch 使用與 AWS IoT 搭配使用的相同帳戶和相同區域設定記錄群組 FleetWise。啟用 AWS IoT 中的記錄時 FleetWise，請提供記錄群組名稱。啟用記錄之後，AWS IoT 會將記錄 FleetWise 傳遞至 CloudWatch 串流中的記錄群組。

您可以在 CloudWatch 主控台中檢視從 AWS IoT 傳送 FleetWise 的記錄資料。如需有關設定記錄群組和檢視 CloudWatch 記錄資料的詳細資訊，請參閱[使用記錄群組](#)。

### 將記錄發佈至的權限 CloudWatch

設定記錄群組的 CloudWatch 記錄需要本節所述的權限設定。如需管理許可的相關資訊，請參閱[IAM 使用者指南中的 AWS 資源存取管理](#)。

透過這些權限，您可以變更記錄組態、設定記錄傳遞 CloudWatch，以及擷取記錄群組的相關資訊。

```
{
  "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"
}
]
```

當允許對所有AWS資源執行動作時，會在策略中以"Resource"設定為指示"\*"。這表示每個動作支援的所有AWS資源都允許執行這些動作。

## 在 AWS IoT FleetWise (主控台) 中設定記錄

本節說明如何使用 AWS IoT 主 FleetWise 控制台設定記錄。

使用 AWS IoT 主 FleetWise 控制台設定記錄

1. 開啟 [AWS IoT FleetWise 主控台](#)。
2. 在左側窗格中，選擇 Settings (設定)。
3. 在「設定值」頁面的「記錄日誌」段落中，選擇編輯。
4. 在記CloudWatch 錄區段中，輸入記錄群組。
5. 若要儲存變更，請選擇「提交」。

啟用記錄後，您可以在[CloudWatch 主控台](#)中檢視記錄資料。

## 在 AWS IoT FleetWise (CLI) 中設定預設記錄

本節說明如何使用 CLI 設 FleetWise 定 AWS IoT 的記錄。

您也可以使用 API 中對應於此處顯示的 CLI 命令的方法，在 AWS API 中執行此程序。您可以使用 [GetLoggingOptions](#) API 操作來獲取當前配置，並使用 [PutLoggingOptions](#) API 操作來修改配置。

若要使用 CLI 設定 AWS IoT 的記錄 FleetWise

1. 若要取得帳戶的記錄選項，請使用get-logging-options指令。

```
aws iotfleetwise get-logging-options
```

- 若要啟用記錄，請使用put-logging-options指令。

```
aws iotfleetwise put-logging-options --cloud-watch-log-delivery  
logType=ERROR,logGroupName=MyLogGroup
```

其中：

logType

要傳送資料至記錄檔的 CloudWatch 記錄檔類型。若要停用記錄，請將值變更為OFF。

logGroupName

作業將資料傳送到的 CloudWatch 記錄群組。在啟用 AWS IoT 的記錄之前，請確定已建立記錄群組名稱 FleetWise。

啟用記錄之後，請參閱[使用 AWS CLI 搜尋記錄項目](#)。

## 記錄AWS物聯網FleetWise使用 API 呼叫AWS CloudTrail

AWS物聯網FleetWise已整合AWS CloudTrail，提供使用者、角色或使用者所採取之動作記錄的服務AWS服務於AWS物聯網FleetWise。CloudTrail擷取的所有 API 呼叫AWS物聯網FleetWise作為事件。捕獲的呼叫包括來自AWS物聯網FleetWise控制台和代碼調用AWS物聯網FleetWiseAPI 作業。如果您建立追蹤，您可以啟用持續傳遞CloudTrail亞馬遜 S3 存儲桶的事件，包括事件AWS物聯網FleetWise。如果您不設定追蹤記錄，仍然可以透過 CloudTrail 主控台 Event history (事件歷史記錄) 檢視最新的事件。使用所收集的資訊CloudTrail，您可以決定提出的要求AWS物聯網FleetWise，提出請求的 IP 地址，提出請求的人，提出請求的時間以及其他詳細信息。

若要進一步了解 CloudTrail，請參閱 [AWS CloudTrail 使用者指南](#)。

## AWS物聯網FleetWise中的資訊CloudTrail

當您建立帳戶時，系統會在您的 AWS 帳戶中啟用 CloudTrail。當活動發生在AWS物聯網FleetWise，該活動記錄在CloudTrail與其他一起事件AWS服務事件歷史。您可以檢視、搜尋和下載 AWS 帳戶的最新事件。如需詳細資訊，請參閱[使用 CloudTrail 事件歷程記錄檢視事件](#)。

在您的事件的持續記錄AWS帳戶，包括事件AWS物聯網FleetWise，建立系統線。一個線索啟用 CloudTrail將日誌文件交付到亞馬遜 S3 存儲桶。根據預設，當您在主控台建立追蹤記錄時，追蹤記錄會套用到所有 AWS 區域。該追蹤會記錄來自 AWS 分割區中所有區域的事件，並將日誌檔案交付到您指定的 Amazon S3 儲存貯體。此外，您可以設定其他 AWS 服務，以進一步分析和處理 CloudTrail 日誌中所收集的事件資料。如需詳細資訊，請參閱下列內容：

- [建立追蹤的概觀](#)
- [CloudTrail 支援的服務和整合](#)
- [設定 CloudTrail 的 Amazon SNS 通知](#)
- [從多個區域接收 CloudTrail 日誌檔案](#)
- [從多個帳戶接收 CloudTrail 日誌檔案](#)

所有AWS物聯網FleetWise動作記錄CloudTrail並記錄在[AWS物聯網FleetWiseAPI 參考資料](#)。例如，對 CreateCampaign、AssociateVehicleFleet 及 GetModelManifest 動作發出的呼叫會在 CloudTrail 日誌檔案中產生項目。

每一筆事件或日誌項目都會包含產生請求者的資訊。身分資訊可協助您判斷下列事項：

- 該請求是否使用根或 IAM 使用者憑證提出。
- 提出該請求時，是否使用了特定角色或聯合身分使用者的暫時安全憑證。
- 該請求是否由另一項 AWS 服務提出。

如需詳細資訊，請參閱 [CloudTrail userIdentity 元素](#)。

## 了解AWS物聯網FleetWise記錄檔項目

追蹤是一種組態，能讓事件以日誌檔案的形式交付到您指定的 Amazon S3 儲存貯體。CloudTrail 日誌檔案包含一個或多個日誌項目。一個事件為任何來源提出的單一請求，並包含請求動作、請求的日期和時間、請求參數等資訊。CloudTrail 日誌檔案並非依公有 API 呼叫追蹤記錄的堆疊排序，因此不會以任何特定順序出現。

以下範例顯示的 CloudTrail 日誌項目會示範 *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 開發人員指南的文件歷史記錄

下表說明 AWS IoT 的文件發行版本 FleetWise。

| 變更                               | 描述   | 日期               |
|----------------------------------|--|------------------|
| <a href="#">視覺系統資料預覽</a>         | 您可以使用來自 AWS IoT FleetWise 的視覺系統資料預覽，從車輛視覺系統收集和整理資料，包括攝影機、雷達和蓋板。它會在雲端中自動同步結構化和非結構化視覺系統資料、中繼資料 (事件 ID、促銷活動、車輛) 和標準感測器 (遙測資料)。        | 2023 年 11 月 26 日 |
| <a href="#">AWS KMS 客戶管理的金鑰</a>  | AWS IoT FleetWise 現在支援 AWS KMS 客戶管理的金鑰。您可以使用 KMS 金鑰加密與儲存在 AWS 雲端中的 AWS IoT FleetWise 資源 (信號目錄、車輛模型、解碼器清單、車輛和資料收集宣傳活動配置) 相關的伺服器端資料。 | 2023 年 10 月 16 日 |
| <a href="#">Amazon S3 中的物件儲存</a> | AWS IoT FleetWise 現在支援使用亞馬遜簡單儲存服務 (Amazon S3) 存放資料。除了 Amazon Timestream 之外，您還可以存放在 Amazon S3 中營銷活動期間收集的資料。                         | 2023 年 6 月 1 日   |
| <a href="#">一般可用性</a>            | 這是 AWS IoT 的公開發行版本 FleetWise。  | 2022 年 9 月 27 日  |
| <a href="#">初始版本</a>             | 這是 AWS IoT FleetWise 開發人員指南的預覽版本。  | 2021 年 11 月 30 日 |

本文為英文版的機器翻譯版本，如內容有任何歧義或不一致之處，概以英文版為準。